Motivation

This project is aimed at analyzing Chinese restaurants' current situations in relation with local Chinese-American populations. It subtracts top 20 Chinese-American populated cities from a tsv file and uses selected values to fetch Yelp restaurants related data via Yelp API. This project tries to figure out if there is a correlation between Chinese-American population in a city and this city's Chinese restaurants total counts. Also, this project tries to measure the overall general quality of Chinese restaurant within a city based on customer reviews on Yelp.

Data Sources

There are two types of data this project uses to analyze the problem. One is a tsv file containing information about cities' Chinese-American population, and the other one is data related to cities' restaurants information subtracted using Yelp API service.

The city population data set is organized from "Race Reporting for the Asian Population by Selected Categories: 2010" retrieved from U.S. Census Bureau on http://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml. It is saved in tsv format. Important variables include city names, state names, Chinese-American population, and ranks of all twenty cities sorted in descending order.

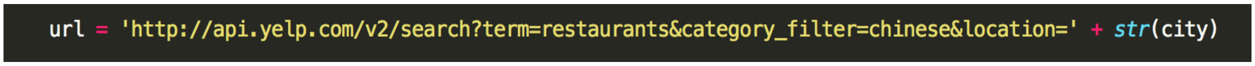
The Yelp API resources are located at the Yelp website, in reference with its documentation at https://www.yelp.com/developers/documentation/v2/search_api, the main url used to retrieve data with Yelp API is shown as below.

```
url = 'http://api.yelp.com/v2/search?term=restaurants&category_filter=chinese&location=' + str(city)
```

The returned data format for Yelp API is json strings. This project uses two different python programs to retrieve data from Yelp API for different purposes. The first program is getting general Chinese restaurant information about each city, the most important variables are city names and total Chinese restaurant counts in these cities. The second program is more focused on each city's specific business information regarding Chinese restaurants, such as review counts, ratings, etc. The Yelp data is extracted at the very moment when running the programs.

Data Manipulation Methods

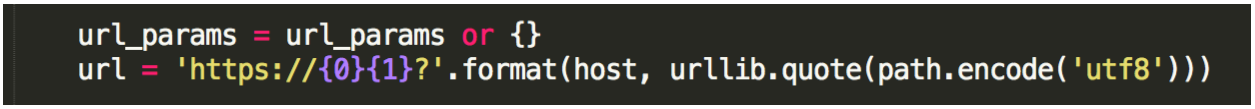
For the tsv file, the data is very clean and the format is quite simple. It doesn't need much manipulation for the data itself, but the values are very important since they will be used for retrieving Yelp data in the yelp1.py program. To process the data in the tsv file, yelp1.py loads tsv file into list format, then subtract and append related values into a new list named cityList. This new list is consisted by city names and Chinese-American population in given cities, sorted in descending order. This new list will be used in later parts of yelp1.py program to retrieval each city's total Chinese restaurant counts via Yelp API. When retrieving each city's total Chinese restaurant counts, the yelp1.py program uses url as below to achieve the goal. This url already uses several conditions to narrow down the data retrieval scope, such as retrieving those with term as restaurants and category_filter as Chinese. These conditions can be accessed in the Yelp API documentations. The only undefined condition is the city name, which uses items in the previous cityList.



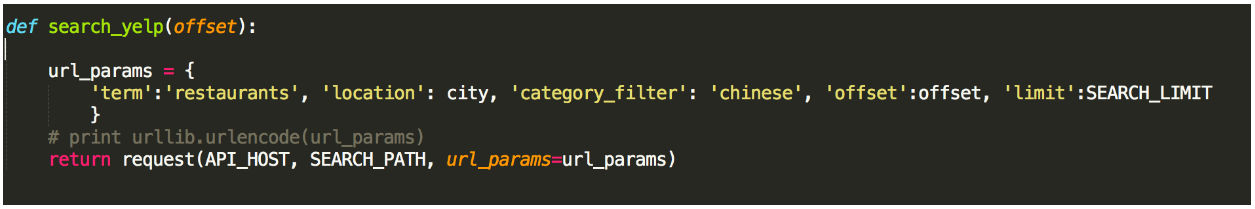
```
url = 'http://api.yelp.com/v2/search?term=restaurants&category_filter=chinese&location=' + str(city)
```

After retrieving each city's total Chinese restaurant counts, the yelp1.py program joins data from tsv file and data retrieved from Yelp API to set up a new list with city names and each city's total Chinese restaurant counts, sorted by total Chinese restaurant counts in descending order. And then the program writes the newly joined list into a csv file.

For yelp2.py, the purpose is to retrieve business detailed data for a specific city. This one is a bit tricky since Yelp API only allows to retrieve 20 entries of businesses data one time, so the program has to handle that issue by adding an offset to the retrieving URL. The code for retrieving more than 20 entries limitation is shown as below.



```
url_params = url_params or {}
url = 'https://{0}{1}?'.format(host, urllib.quote(path.encode('utf8')))
```



```
def search_yelp(offset):

    url_params = {
        'term':'restaurants', 'location': city, 'category_filter': 'chinese', 'offset':offset, 'limit':SEARCH_LIMIT
        }
    # print urllib.urlencode(url_params)
    return request(API_HOST, SEARCH_PATH, url_params=url_params)
```

There are several challenges for retrieving detailed business data for a specific city. The first one is the Yelp API limitations. By default, Yelp API only allows 20 entries of detailed business data per retrieval, thus it is quite difficult to write code to offset the limitation. Another issue is that for this limitation is capped at 1000 entries even with the offset code. Thus the data for several big cities with more than 1000 Chinese restaurants are incomplete, which results slightly inaccuracy when calculating the average rating for all Chinese restaurants in these cities.

In yelp2.py program, I also created an SQL database to store detailed business information for Chinese restaurants in San Francisco just to play around the data.

Analysis and Visualization

To explore the data with further analysis, there is a semi-correlation between Chinese-American population versus Chinese restaurant counts in these cities.

Chart 1: Top 20 Cities with Most Chinese-American Populations

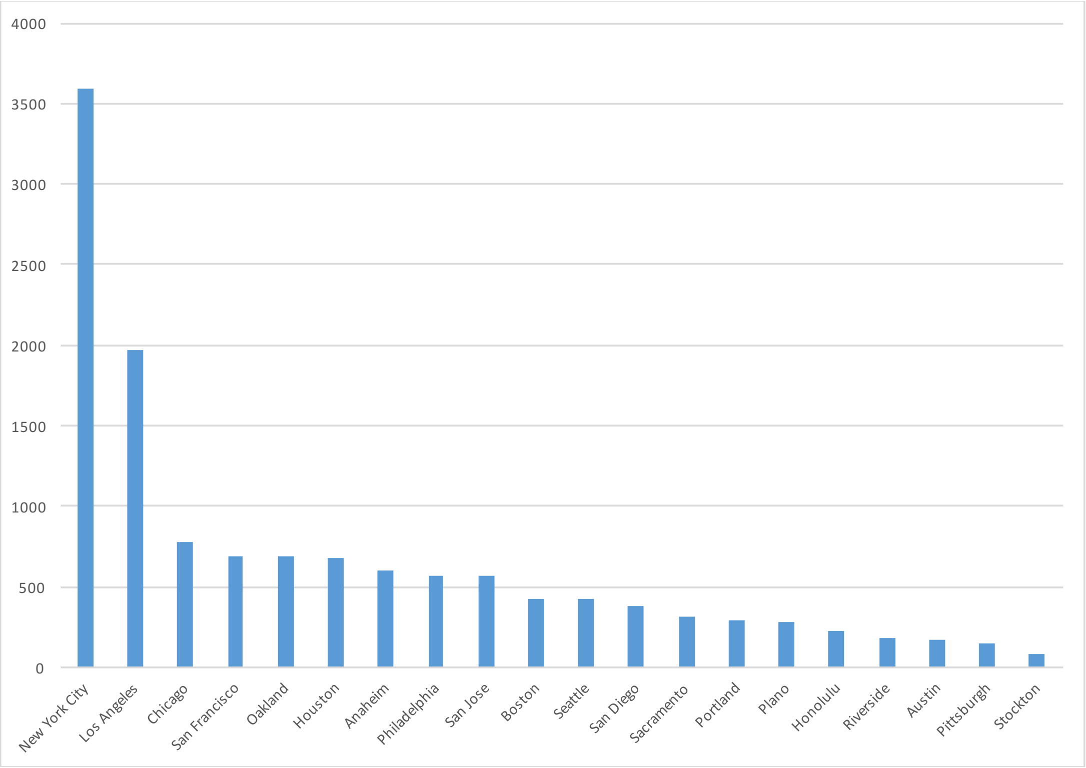


Chart 2: Population vs. Restaurant Counts

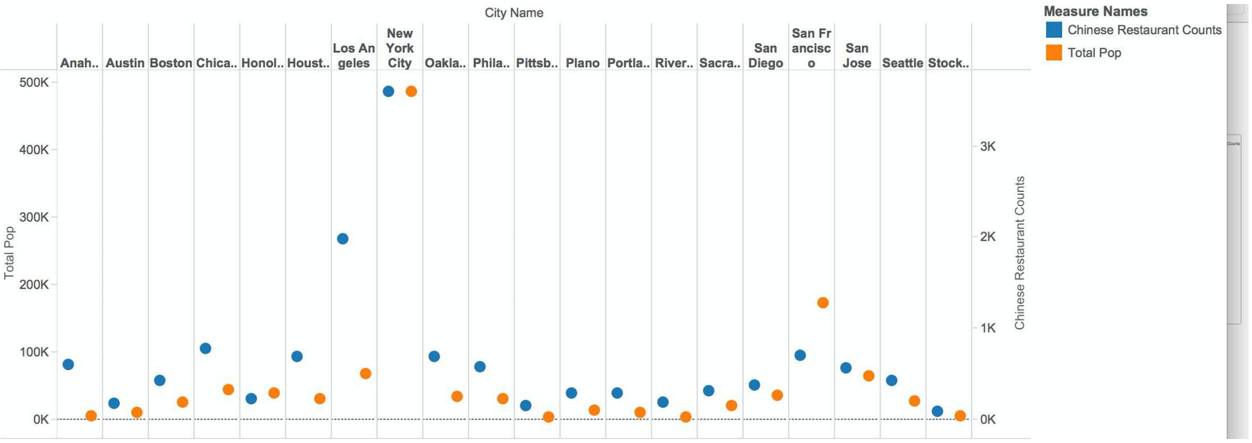


Chart 3: Chinese Restaurant Counts in 20 Cities

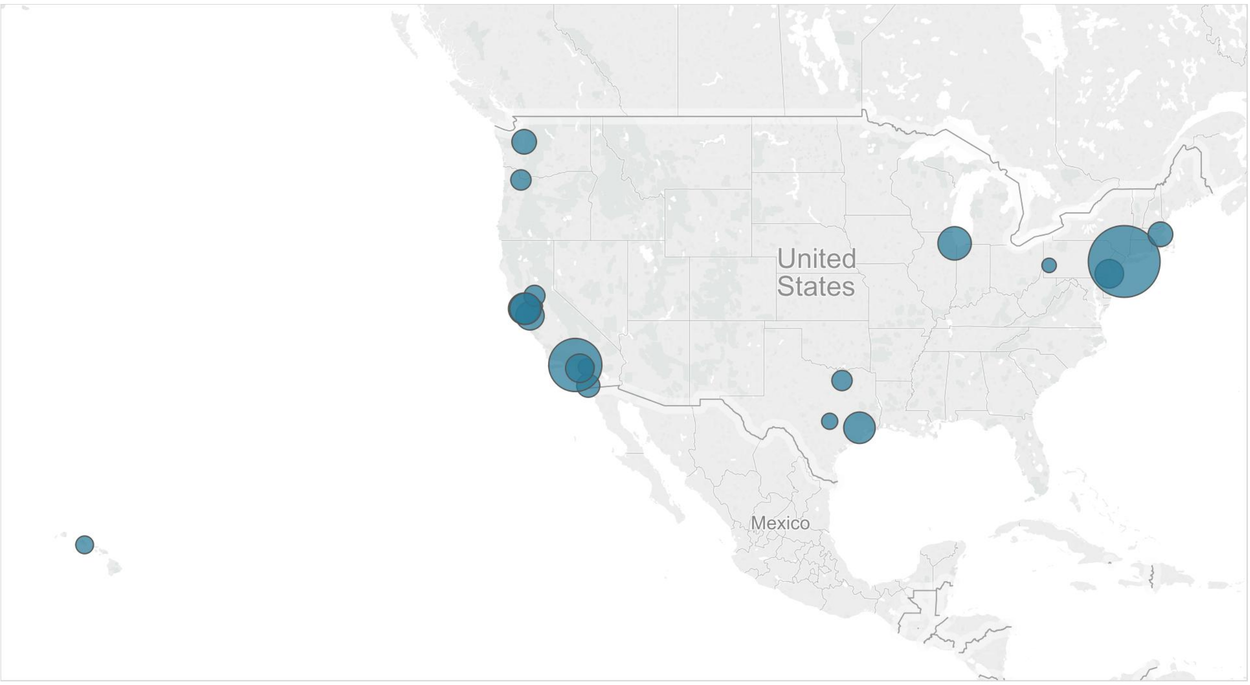
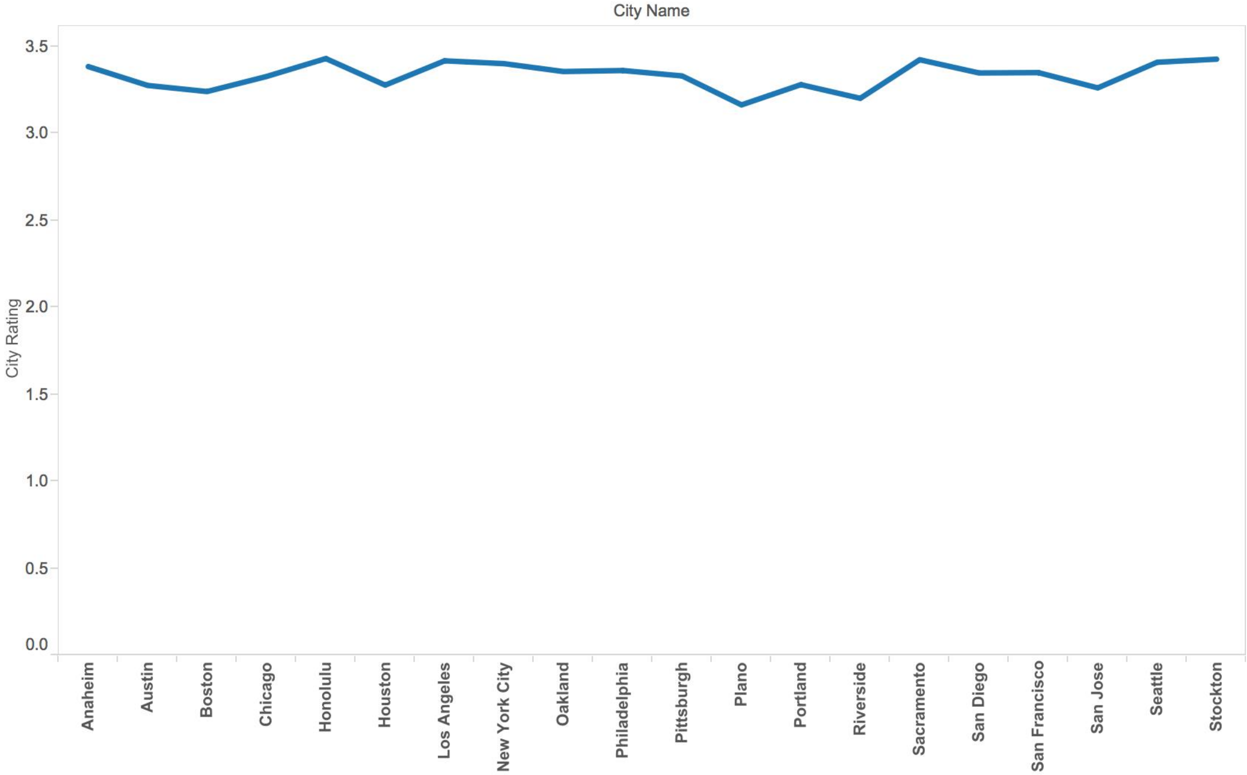


Chart 4: City Average Chinese Restaurant Ratings


City Name
City Rating
Anaheim
Austin
Boston
Chicago
Honolulu
Houston
Los Angeles
New York City
Oakland
Philadelphia
Pittsburgh
Plano
Portland
Riverside
Sacramento
San Diego
San Francisco
San Jose
Seattle
Stockton