

# 1. Introduction

In the past few years, Austin, Texas has been widely recognized by many media as the best city to live in the United States (<https://www.cnn.com/2019/04/15/us-news-world-report-best-places-to-live-in-the-us-in-2019.html>). With an ever increasing cluster of high-tech companies, this growing Austin metropolitan area is thus nicknamed silicon hills. The name is analogous to Silicon Valley, but refers to the hilly terrain on the west side of Austin. High tech industries in the area include enterprise software, semiconductors, corporate R&D, biotechnology, the video game industry, and a variety of startup companies.



Seattle and San Francisco are the two existing gravity centers of the tech industry. Among the five largest tech companies in the United States, three (Apple, Alphabet, Facebook) headquartered in San Francisco area and two (Microsoft and Amazon) based their roots in Seattle area. Therefore, one would ask the question how different is the emerging Austin from the two existing tech centers on the west coast? How can Austin develop its city to better catch up with the two leading champions?



This project compares distribution of venues in Austin to that in Seattle and San Francisco. The comparison aims to explore how Austin can develop itself with respect to venues to catch up with Seattle and San Francisco.

## 2. Data Requirement

This project contains the following data:

1) Zip code of the three cities with associated latitude and longitude

The Zip code will be used as surrogate of neighborhood of each city. The data is available

from: <https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/export/>

2) Venues in each zip code area in the three cities:

Venues will be obtained from Foursquare APIs. By using this API, we will get all the venues in each zip code.

### 2.1 Location Data Preparation

We first import all the needed modules. In this study, we use zip codes as the surrogate for neighborhoods. We find zipcodes of all the cities in the United States along with latitude and longitude from the following website.

```
#https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/export/

df = pd.read_csv('us-zip-code-latitude-and-longitude.csv')
df = df.drop(['Timezone', 'Daylight savings time flag'], axis=1)
```

In the United States, there are many cities in different states with the same name. For example, there is Austin in Texas, Minnesota, Arkansas, and Indiana. Therefore, the dataframe is filtered out by both city name and state. Finally, a dataframe containing all the zip codes in Austin, Seattle, and San Francisco is obtained.

```
city=['Austin', 'Seattle', 'San Francisco']
df_1=df.loc[(df['City']=='Austin')&(df['State'] == 'TX')]
df_2=df.loc[(df['City']=='Seattle')&(df['State'] == 'WA')]
df_3=df.loc[(df['City']=='San Francisco')&(df['State'] == 'CA')]
frame=[df_1,df_2,df_3]
df_three=pd.concat(frame)
df_three.reset_index(inplace=True)
df_three.rename(columns={'Zip':'Zip Code'}, inplace=True)
df_three = df_three.drop('index', axis=1)
df_three.head()
```

	Zip Code	City	State	Latitude	Longitude
0	78701	Austin	TX	30.271270	-97.741030
1	78705	Austin	TX	30.292424	-97.738560
2	78727	Austin	TX	30.425652	-97.714190
3	78762	Austin	TX	30.326374	-97.771258
4	78763	Austin	TX	30.335398	-97.559807

## 2.2 Exploratory Data Preparation

Foursquare Credentials and Version were defined. The Foursquare API will get the top 100 venues in each zip code area within a radius of 5000 meters.

```
CLIENT_ID = 'VZV2XVTGSGOPBFSNTDGZFKNAAOKSSDNORH5S5G4QB244PAVQ' # your Foursquare
CLIENT_SECRET = 'SXX2KXI0BBIXVH0ZSXEAT1X0H5AT5PPSFQPPXAZXDKC0ZNJKO' # your Foursqu
VERSION = '20180604'

limit=100
radius=5000
```

The Foursquare API is then used to obtain all the venues for each zip code based on latitude and longitude in Austin, Seattle, and San Francisco.

```
# run function getNearbyVenues on each zip code
three_venues = getNearbyVenues(names=df_three['Zip Code'],
                                latitudes=df_three['Latitude'],
                                longitudes=df_three['Longitude']
                                )

three_venues.head()
```

	Zip Code	Zip Code Latitude	Zip Code Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	78701	30.27127	-97.74103	Chi'lantro BBQ	30.270600	-97.741928	Food Truck
1	78701	30.27127	-97.74103	Caffé Medici	30.270119	-97.742154	Coffee Shop
2	78701	30.27127	-97.74103	Capitol Visitors Center	30.272625	-97.739300	Capitol Building
3	78701	30.27127	-97.74103	Paramount Theatre	30.269457	-97.742077	Movie Theater
4	78701	30.27127	-97.74103	The Townsend	30.269611	-97.742448	Lounge

The venue category at each zip code of each city will be analyzed to explore the similarity between Austin and Seattle or San Francisco.

### 3. Analysis and Discussion

To explore the distribution of venues in each zip code, we analyze each zip code with pandas one hot encoding for the venue categories.

```
# one hot encoding
three_onehot = pd.get_dummies(three_venues[['Venue Category']], prefix="", prefix

# add neighborhood column back to dataframe
three_onehot['Zip Code'] = three_venues['Zip Code']

# move neighborhood column to the first column
fixed_columns = [three_onehot.columns[-1]] + list(three_onehot.columns[:-1])
three_onehot = three_onehot[fixed_columns]

three_onehot.head()
```

```
three_grouped = three_onehot.groupby('Zip Code').mean().reset_index()
```

A dataframe was then created for each zip code along with the top 20 most common venues categories in each zip code.

```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

```
num_top_venues = 20

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Zip Code']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
zipcodes_venues_sorted = pd.DataFrame(columns=columns)
zipcodes_venues_sorted['Zip Code'] = three_grouped['Zip Code']

for ind in np.arange(three_grouped.shape[0]):
    zipcodes_venues_sorted.iloc[ind, 1:] = return_most_common_venues(three_groupe

zipcodes_venues_sorted.head()
```

### 3.1 Clustering zip codes

All the zip codes were then clustered into 5 categories based on the 20 most common venue categories in each zip code. In this way, we can evaluate the similarity among zip codes within a city. Among the three cities, Austin, Seattle, and San Francisco, we can evaluate whether the distribution of venue categories is similar or not.

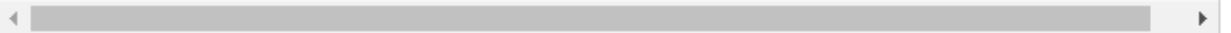
There are many methods to perform clustering, such as k-means, affinity propagation and mean-shift, to name a few. In this study, we used KMeans from sklearn.cluster. We set the number of clusters 5 and called KMeans.

```
# set number of clusters
kclusters = 5

three_grouped_clustering = three_grouped.drop('Zip Code', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(three_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```



```
array([2, 2, 0, 0, 0, 0, 0, 2, 2, 0])
```

```
# add clustering labels
zipcodes_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

We then assign latitude and longitude to each zip code associated with the 20 most common venue categories of each zip code.

```
three_merged = df_three

# merge three_grouped with df_three to add Latitude/Longitude for each zip code
three_merged = three_merged.join(zipcodes_venues_sorted.set_index('Zip Code'), on='Zip Code')

three_merged.head() # check the last columns!
```

	Zip Code	City	State	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
0	78701	Austin	TX	30.271270	-97.741030	0.0	Cocktail Bar	Hotel	Sandwich Place	
1	78705	Austin	TX	30.292424	-97.738560	0.0	Burger Joint	Food Truck	Boutique	Conv
2	78727	Austin	TX	30.425652	-97.714190	0.0	Mediterranean Restaurant	Intersection	Cosmetics Shop	Fishir
3	78762	Austin	TX	30.326374	-97.771258	2.0	Lake	Yoga Studio	Eye Doctor	
4	78763	Austin	TX	30.335398	-97.559807	0.0	Discount Store	Bar	Video Store	Flea

After cleaning the dataframe, we obtained the number of zip codes falls under each cluster. As we can see, only 1 zip code falls under Cluster 5.

```
# drop any NaN value
three_merged.dropna(how='any',inplace=True)
three_merged.reset_index(inplace=True)
three_merged = three_merged.drop('index', axis=1)
```

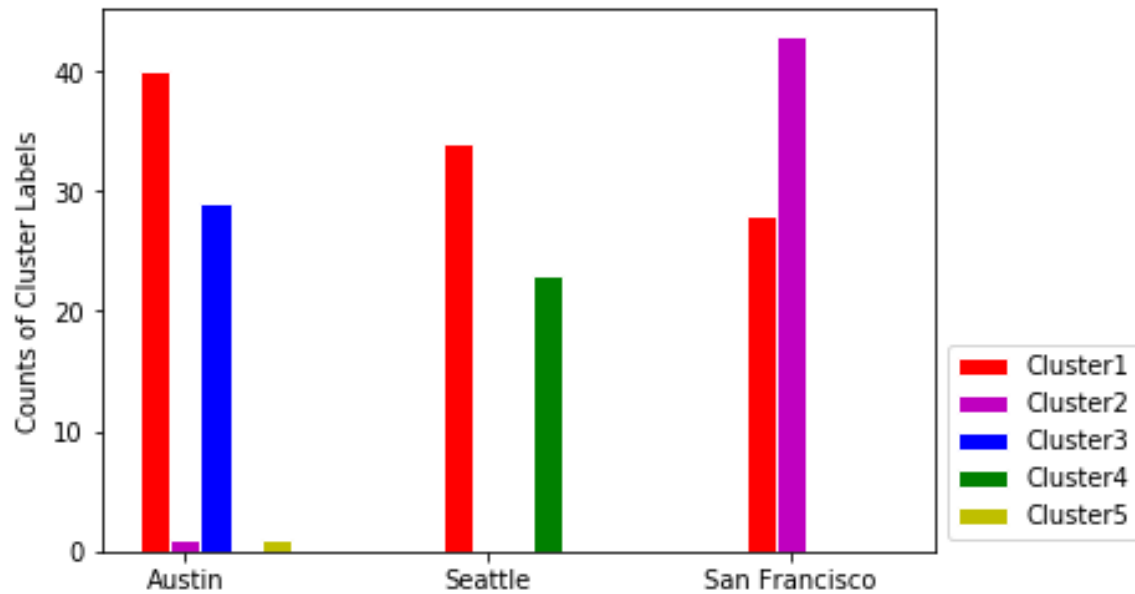
```
three_merged['Cluster Labels'].value_counts()
```

```
0.0    102
1.0     44
2.0     29
3.0     23
4.0      1
Name: Cluster Labels, dtype: int64
```



### 3.2 Analyze clusters

**Among cities.** The distribution of clusters in each city is then analyzed. After obtaining the number of zip code under each cluster for each city, we use matplotlib to visualize the difference among cities in bar plot.



The bar plot shows that Cluster 1 is most common in both Austin and Seattle, while Cluster 2 is most common in San Francisco. All three cities have lots of zip codes fall under Cluster 1, but other than Cluster 1, venue clusters of the remaining zip codes differ among cities. In Austin, the other major cluster is Cluster 3 while Cluster 2 and 5 are very minimal. In Seattle, the other major cluster is Cluster 4, while no Cluster 2, 3 and 5 are detected in Seattle. In San Francisco, the other major cluster is Cluster 2, while no Cluster 3, 4 and 5 are detected. Therefore, these three cities are very different from each other with respect to clusters except that cluster 1 is a major cluster in all three cities.

**Composition of clusters.** To visualize the detailed composition of each cluster, word cloud is used to help the major venue categories in each cluster



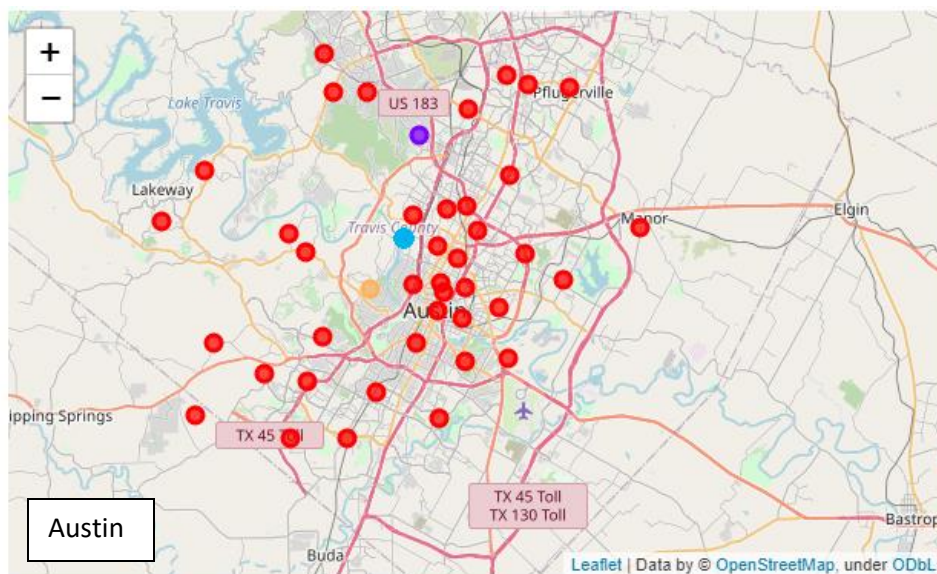


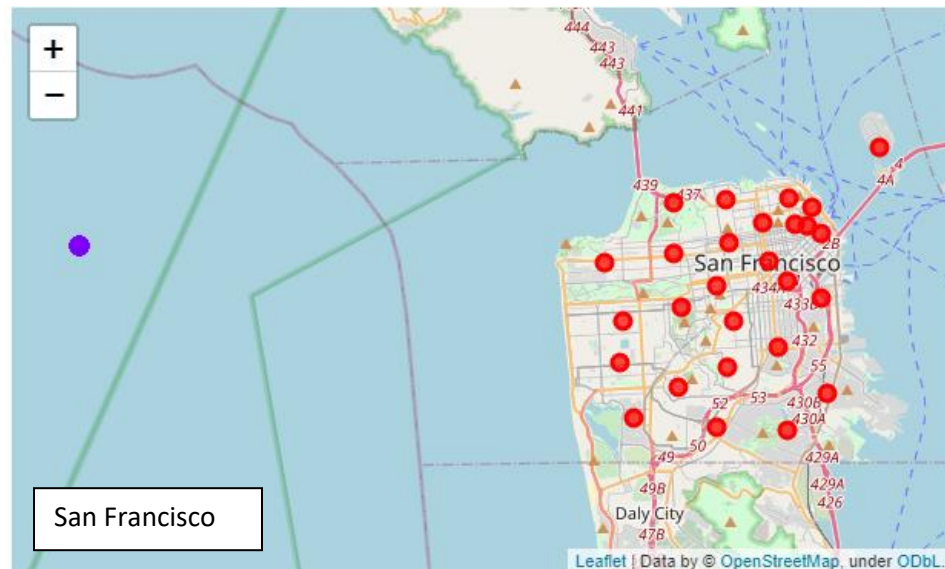
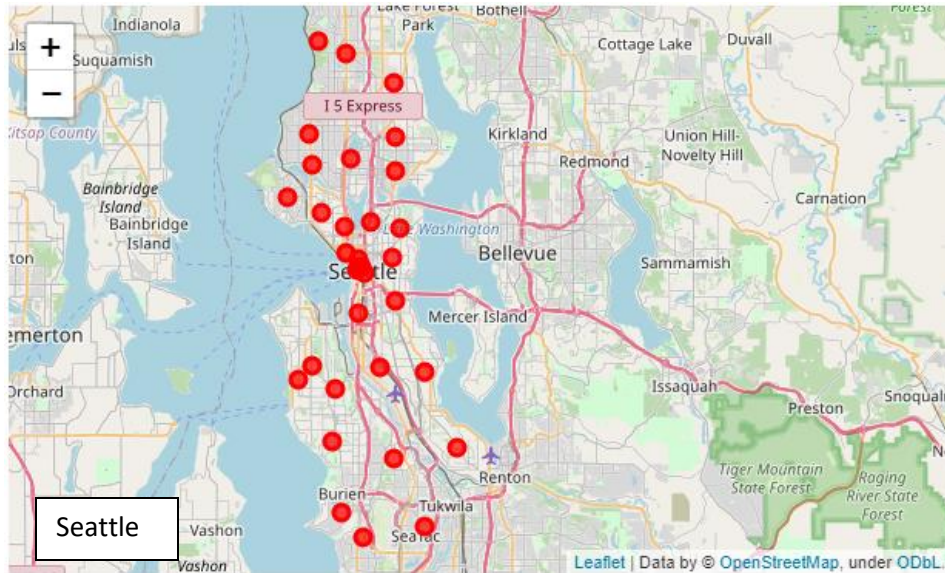
do not have lake in their city's limit. No wonder a lake-focused cluster would not appear in Seattle and San Francisco. Cluster 4 majorly consists of yoga studio and lawyer and doctor offices. Actually Cluster 3 and Cluster 4 are very similar as both include yoga studio and eye doctor as one of the major venues. The difference of Cluster 4 in Seattle and Cluster 3 in Austin probably thanks to the fact that Austin has lakes and Seattle has more lawyer office in its city limits.

Compare Austin to Seattle and San Francisco reveals that while Austin matches Seattle and San Francisco in food and restaurant. Austin lags behind San Francisco in fabric stores, fishing store, and farmers market. To improve its attraction to tech companies, Austin can learn from San Francisco to develop more liberal living style of farmers market and fabric shops. Austin is relatively similar to Seattle in term of venue categories; nevertheless, Austin can still learn from Seattle to develop more professional service such as lawyer office.

### 3.3 Visualize the clusters on map

The maps were visualized by folium map. With respect to distribution of clusters in the city, all three cities are similar. Cluster 1 restaurant spreads across all three cities and the other clusters are more sporadically distributed.





## 4. Conclusion

In this study, the distribution of venue categories in Austin is compared to Seattle and San Francisco. Findings and recommendations are summarized below:

- kmeans and word cloud are power tool to cluster and visualize venue composition of each zip code.
- Austin is more similar to Seattle than to San Francisco.
- All three cities have many zip codes whose primary venues are restaurants.
- San Francisco stands out in term of farmers market and fabric shops probably thanks to its more liberal living culture.
- Austin can improve its attraction to tech companies by develop more farmers market like San Francisco and develop more lawyer office like Seattle.