

Finding a Zipcodes in Seattle with ice cream shops

Yay-hyung Cho

July 5th, 2020

1. Introduction

1.1. Business problem and target population

Seattle is the Washington State's largest city, it's home to many tech industries, with Microsoft and Amazon headquartered in its metropolitan area. Also, it is where I call home. It is a proper city for avid coffee drinkers since you can easily find a place with a good cup of joe, but that is not the case for ice cream lovers. If you want to get a fantastic cone of ice cream near where you live, where would be an excellent place to live in the greater Seattle area? Any ice cream aficionado who wants to easily walk to a good ice cream shop would be my target population. With summer coming up in the Seattle area, any Seattleites who are searching for their frozen treat would be my population of interest as well.

2. Data acquisition and cleaning

2.1. Data source

I used zip code and geocoordinate data from the website called "[simple maps](https://simplemaps.com/data/us-zips)" (<https://simplemaps.com/data/us-zips>) and focused on the greater Seattle area.

```
[64]: import pandas as pd
import numpy as np
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
import folium # map rendering library
import requests
from sklearn.cluster import KMeans # import k-means from clustering stage
```

read US zipcodes & geocoordinates csv file

```
[35]: df_us_zipcodes = pd.read_csv("https://raw.githubusercontent.com/cholovesdata/Coursera_Capstone/master/us_zipcode_geocoordinates.csv")
# select relevant columns
df_us_zipcodes = df_us_zipcodes[['zip', 'lat', 'lng', 'city', 'state_name']]
df_us_zipcodes
```

```
[35]:
```

	zip	lat	lng	city	state_name
0	601	18.18004	-66.75218	Adjuntas	Puerto Rico
1	602	18.36073	-67.17517	Aguada	Puerto Rico
2	603	18.45439	-67.12202	Aguadilla	Puerto Rico
3	606	18.16724	-66.93828	Maricao	Puerto Rico
4	610	18.29032	-67.12243	Anasco	Puerto Rico
...
33094	99923	55.97796	-130.03671	Hyder	Alaska
33095	99925	55.55767	-132.97627	Klawock	Alaska
33096	99926	55.12617	-131.48928	Metlakatla	Alaska
33097	99927	56.25100	-133.37571	Point Baker	Alaska
33098	99929	56.36950	-131.93648	Wrangell	Alaska

33099 rows × 5 columns

2.2. Data cleaning

After downloading a CSV file from the simple maps, I selected the zip codes in the Washington state only first, and then for Seattle city. Afterward, I used geopy library to get the latitude and longitude values of Seattle, Washington. After creating a map of Seattle, I superimposed neighborhoods on top.

select zipcodes in Washington State

```
[ 44 ]: df_wa_zipcodes = df_us_zipcodes[df_us_zipcodes["State"] == "Washington"]
df_wa_zipcodes
```

```
[ 44 ]:
```

	PostalCode	Latitude	Longitude	City	State	
	32265	98001	47.30998	-122.26521	Auburn	Washington
	32266	98002	47.30836	-122.21639	Auburn	Washington
	32267	98003	47.30513	-122.31508	Federal Way	Washington
	32268	98004	47.61884	-122.20595	Bellevue	Washington
	32269	98005	47.61478	-122.16862	Bellevue	Washington

	32856	99363	46.06652	-118.88846	Walla Walla	Washington
	32857	99371	46.80678	-118.31679	Wahluke	Washington
	32858	99401	46.08744	-117.25143	Anatone	Washington
	32859	99402	46.19394	-117.14740	Asotin	Washington
	32860	99403	46.37243	-117.25274	Clarkston	Washington

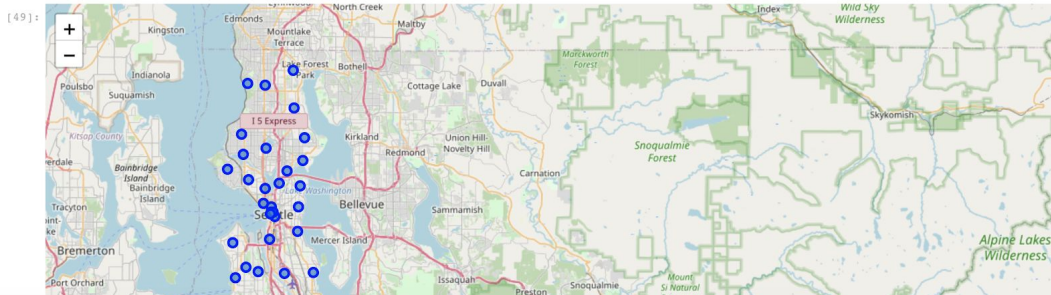
596 rows x 5 columns

Create a map of Seattle, Washington with neighborhoods superimposed on top.

```
[ 49 ]: # create map of Seattle using latitude and longitude values
map_seattle = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
# add markers to map
for latitude, longitude, postal_code in zip(df_seattle_zipcodes['Latitude'], df_seattle_zipcodes['Longitude'], df_seattle_zipcodes['PostalCode']):
    label = '{}'.format(postal_code)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [latitude, longitude],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_seattle)

map_seattle
```



2.3. Link with Foursquare

I set up a Foursquare ID and also a function to get nearby venues given latitude and longitude pairs. I also retrieved nearby venues in Seattle with postal codes with 500 radius.

retrieve nearby venues in seattle with postal code

```
[54]: venues_seattle = getNearbyVenues(names=df_seattle_zipcodes['PostalCode'], latitudes=df_seattle_zipcodes['Latitude'], longitudes=df_seattle_zipcodes['Longitude'], radius=500)
venues_seattle
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	98101	47.61129	-122.33454	Din Tai Fung Dumpling House	47.612671	-122.335073	Dumpling Restaurant
1	98101	47.61129	-122.33454	ACT Theatre	47.610763	-122.332905	Theater
2	98101	47.61129	-122.33454	Monorail Espresso	47.610828	-122.335048	Coffee Shop
3	98101	47.61129	-122.33454	Timbuk2	47.612561	-122.334223	Accessories Store
4	98101	47.61129	-122.33454	Grand Hyatt Seattle	47.612583	-122.333618	Hotel
...
532	98199	47.65142	-122.40270	Walkin' The Streets Of Magnolia	47.651458	-122.405470	Athletics & Sports
533	98199	47.65142	-122.40270	Rudy's Place	47.653865	-122.400653	Breakfast Spot
534	98199	47.65142	-122.40270	Kaspar's Special events and catering	47.654216	-122.401052	Restaurant
535	98199	47.65142	-122.40270	Espresso Bar	47.654860	-122.401153	Coffee Shop
536	98199	47.65142	-122.40270	Discovery Park Emerson Entrance	47.650766	-122.409143	Park

537 rows x 7 columns

As a result, I was able to group Seattle venues by zip codes.

group seattle venues by zipcodes

```
[56]: venues_seattle.groupby("Neighborhood").count()
```

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
98101	30	30	30	30	30	30
98102	30	30	30	30	30	30
98103	8	8	8	8	8	8
98104	30	30	30	30	30	30
98105	15	15	15	15	15	15
98106	5	5	5	5	5	5
98107	30	30	30	30	30	30
98108	3	3	3	3	3	3
98109	25	25	25	25	25	25
98112	4	4	4	4	4	4
98115	1	1	1	1	1	1
98117	22	22	22	22	22	22
98118	2	2	2	2	2	2
98119	13	13	13	13	13	13
98121	30	30	30	30	30	30
98122	26	26	26	26	26	26
98125	22	22	22	22	22	22
98126	8	8	8	8	8	8
98133	27	27	27	27	27	27
98134	30	30	30	30	30	30
98136	13	13	13	13	13	13
98144	14	14	14	14	14	14
98146	12	12	12	12	12	12
98148	4	4	4	4	4	4

I also tried to find out how many unique categories can be curated from all the returned venues

find out how many unique categories can be curated from all the returned venues

```
[57]: print('There are {} uniques categories.'.format(len(venues_seattle['Venue Category'].unique())))
```

There are 171 uniques categories.

Categories of venues in Seattle

```
[58]: print(venues_seattle['Venue Category'].unique())
```

```
['Dumpling Restaurant' 'Theater' 'Coffee Shop' 'Accessories Store' 'Hotel'
 "Women's Store" 'Clothing Store' 'French Restaurant'
 'New American Restaurant' 'Performing Arts Venue' 'Pizza Place'
 'Discount Store' 'Cosmetics Shop' 'American Restaurant' 'Yoga Studio'
 'Dessert Shop' 'Cocktail Bar' 'Creperie' 'Bridal Shop' 'Wine Bar'
 'Concert Hall' 'Seafood Restaurant' 'Gym' 'Trail' 'Garden'
 'Bubble Tea Shop' 'Italian Restaurant' 'Bakery' 'Dog Run' 'Restaurant'
 'Boat or Ferry' 'Sandwich Place' 'Thai Restaurant' 'Massage Studio' 'Spa'
 'Bus Stop' 'Martial Arts Dojo' 'Food & Drink Shop' 'Korean Restaurant'
 'Cemetery' 'Furniture / Home Store' 'Golf Course' 'Caribbean Restaurant'
 'Tennis Court' 'Lake' 'Harbor / Marina' 'Park' 'Japanese Restaurant'
 'Mini Golf' 'Café' 'Breakfast Spot' 'Building' 'Social Club'
 'Sushi Restaurant' 'BBQ Joint' 'Boutique' 'Video Game Store' 'Hotel Bar'
 'Nightclub' 'Poke Place' 'Deli / Bodega' 'Chinese Restaurant' 'Brewery'
 'Steakhouse' 'Food Truck' 'Burger Joint' 'Gift Shop' 'Pharmacy'
 'Juice Bar' 'Bistro' 'Bus Station' 'Bar' 'Baseball Field' 'Bus Line'
 'Donut Shop' 'Vietnamese Restaurant' 'Pet Store' 'Post Office'
 'Vegetarian / Vegan Restaurant' 'Mexican Restaurant' 'Noodle House'
 'Mediterranean Restaurant' 'Toy / Game Store' 'Marijuana Dispensary'
 'Gaming Cafe' 'Supermarket' 'Rock Club' 'Farmers Market'
 'Miscellaneous Shop' 'Hot Dog Joint' 'Museum' 'Airport Terminal'
 'Herbs & Spices Store' 'Playground' 'Airport' 'ATM' 'IT Services'
 'Convenience Store' 'Soccer Field' 'Ski Area' 'Beer Bar' 'Fish Market'
 'Mobile Phone Shop' 'Rental Car Location' 'Grocery Store'
 'Shipping Store' 'Smoke Shop' 'Outdoor Supply Store'
 'General College & University' 'Scenic Lookout' 'Dive Bar' 'Bike Shop'
 'Ice Cream Shop' 'Kids Store' 'Electronics Store'
 'Middle Eastern Restaurant' 'Movie Theater' 'Speakeasy' 'Gastropub'
 'Turkish Restaurant' 'Ethiopian Restaurant' 'Cupcake Shop'
 'Asian Restaurant' 'Jewelry Store' 'Latin American Restaurant'
 'Taco Place' 'Greek Restaurant' 'Gymnastics Gym' 'Bank'
 'Sporting Goods Shop' 'Storage Facility' 'Gym / Fitness Center'
 'Hobby Shop' 'Fabric Shop' 'Casino' 'Fast Food Restaurant' 'Motel'
 'Board Shop' 'Big Box Store' 'Castle' 'Cuban Restaurant'
```

2.3. Exploratory data analysis (Descriptive statistics)

Afterward, I analyzed each zip code in the Seattle area and calculated the mean for each category of venue.

Analyze each zipcode in Seattle city area

```
[59]: # one hot encoding
seattle_onehot = pd.get_dummies(venues_seattle[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
seattle_onehot['Neighborhood'] = venues_seattle['Neighborhood']

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

seattle_onehot.head()
```

```
[59]:
```

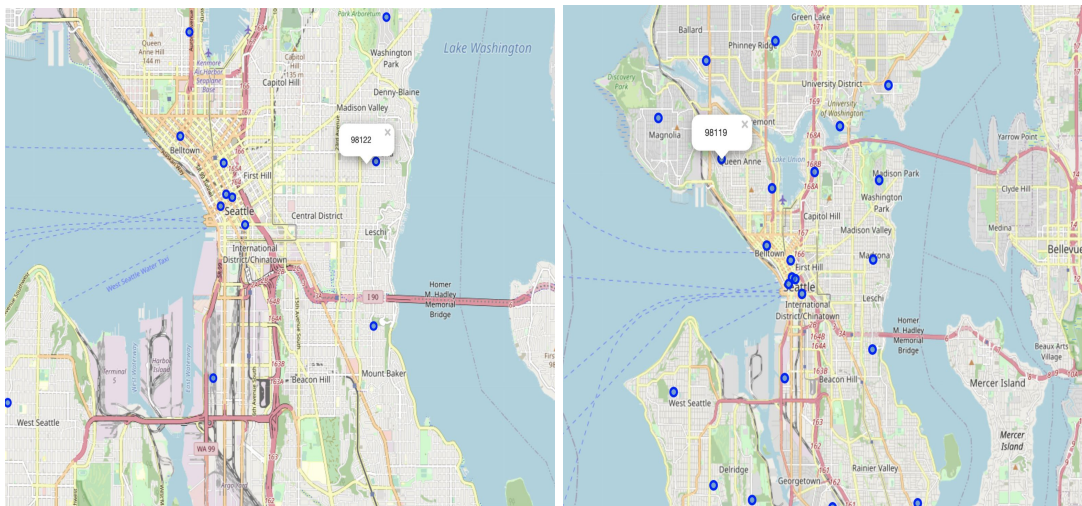
	Neighborhood	ATM	Accessories Store	Airport	Airport Lounge	Airport Terminal	American Restaurant	Animal Shelter	Art Museum	Arts & Entertainment	...	Tour Provider	Toy / Game Store	Trail	Turkish Restaurant	Vegetarian / Vegan Restaurant	Video Game Store	Vietnamese Restaurant	Wine Bar	Women's Store	Yoga Studio
0	98101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	98101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	98101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	98101	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	98101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0


```
[ 60]: seattle_grouped = seattle_onehot.groupby('Neighborhood').mean().reset_index()
seattle_grouped
```

```
[ 60]:
```

	Neighborhood	ATM	Accessories Store	Airport	Airport Lounge	Airport Terminal	American Restaurant	Animal Shelter	Art Museum	Arts & Entertainment	...	Tour Provider	Toy / Game Store	Trail	Turkish Restaurant	Vegetarian / Vegan Restaurant	Video Game Store	Vietnamese Restaurant	Wine Bar
0	98101	0.000000	0.033333	0.00	0.000000	0.000000	0.066667	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333
1	98102	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.066667	0.000000	0.000000	0.000000	0.000000	0.000000
2	98103	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	98104	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000	0.000000
4	98105	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	98106	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.200000	0.000000	0.000000	0.000000	0.000000	0.000000
6	98107	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.033333	0.000000	0.000000	0.033333	0.000000	0.033333	0.000000
7	98108	0.000000	0.000000	0.00	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	98109	0.040000	0.000000	0.04	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.080000	0.000000	0.000000	0.000000	0.000000	0.000000
9	98112	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	98115	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	98117	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
12	98118	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	98119	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	98121	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000
15	98122	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.038462	0.000000	0.000000	0.000000	0.038462
16	98125	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.045455	0.000000
17	98126	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
18	98133	0.037037	0.000000	0.00	0.000000	0.000000	0.037037	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.074074	0.000000
19	98134	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.033333	0.000000	0.033333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000
20	98136	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.078923	0.000000	0.000000	0.000000	0.000000	0.000000
21	98144	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
22	98145	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Lastly, I was able to identify two areas (i.e., WA98112, WA98119) where they had ice cream shop.



Show zipcode neighborhood with Ice Cream Shop

```
[63]: seattle_ice_cream = seattle_grouped[["Neighborhood", "Ice Cream Shop"]]  
seattle_ice_cream
```

```
[63]:
```

	Neighborhood	Ice Cream Shop
0	98101	0.000000
1	98102	0.000000
2	98103	0.000000
3	98104	0.000000
4	98105	0.000000
5	98106	0.000000
6	98107	0.000000
7	98108	0.000000
8	98109	0.000000
9	98112	0.000000
10	98115	0.000000
11	98117	0.000000
12	98118	0.000000
13	98119	0.076923
14	98121	0.000000
15	98122	0.038462
16	98125	0.000000
17	98126	0.000000
18	98133	0.000000
19	98134	0.000000
20	98136	0.000000
21	98144	0.000000
22	98146	0.000000
23	98148	0.000000
24	98154	0.000000

3. Results and future direction

Compare to coffee shops (19 neighborhoods), there were only two zip codes where they had registered ice cream shops from Foursquare. Madison Park (WA 98112) and Queen Anne (WA 98119) area would be a good place to live for ice cream aficionado in the Seattle area.

As for the future direction, it would yield more accurate data if I increase the radius value from 500 to 2000. Also, it would be interesting to know what other venues are related to the place where they have an ice cream shop. For example, I can speculate that it would be more profitable if the ice cream place is located where there are more restaurants or schools.