Coursera Capston - Applied Data Science

Gary S. Netherton - June 30, 2020

This notebook will be used for the capstone project for the Applied Data Science coursework.

In [4]:
```
import pandas as pd
import numpy as np
```

In [5]:
```
print("Hello Capstone Project Course!")
```

```
Hello Capstone Project Course!
```

Purpose:

> I have the opportunity to relocate. My choice, however, will revolve around the selection of restaurants and the performing arts. I have lived in places such as Chicago that have a plethora of food options. I have also lived places where my options were limited to Subway, McDonald's, and a couple of teriyaki joints. As I want this to be my last move - I am nearing retirement - I want to find a comfortable, compatible city to match my wants and needs.
>
> Additionally, I am a museum and play/performance junky. I really enjoy walking through museums and attending live performances.. I am currently reviewing job offers from two firms - one is located in Temecula, CA and one is located in Portland, OR.

Target Audience(s):

> 1. Me
> 2. Families with children
> 3. Families with cultural backgrounds
>
> *The Foursquare API will enable me to research both dining establishments as well as medical facilities. I will review both the number of both as well as the ratings provided by users. Due to recent changes in the Foursquare API for security purposes, I will not dig into specific users and their comments.*

In [6]:

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library

print('Libraries imported.')
import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library

# Let me know when the libraries are loaded.

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... failed with initial frozen solve. Retrying
with flexible solve.
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: C:\Users\gnetherton\Anaconda3

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-4.1.0               |             py_1         614 KB  conda-forge
    certifi-2019.11.28         |           py37_0         148 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         835 KB

The following NEW packages will be INSTALLED:

  altair             conda-forge/noarch::altair-4.1.0-py_1
  vincent            conda-forge/noarch::vincent-0.4.4-py_1

The following packages will be SUPERSEDED by a higher-priority channel:

  certifi                                         pkgs/main --> conda-forge

The following packages will be DOWNGRADED:

  folium                                        0.11.0-py_0 --> 0.5.0-py_0



Downloading and Extracting Packages

certifi-2019.11.28   | 148 KB    |            |   0% 
certifi-2019.11.28   | 148 KB    | #          |  11% 
certifi-2019.11.28   | 148 KB    | ########## | 100% 

vincent-0.4.4        | 28 KB     |            |   0% 
vincent-0.4.4        | 28 KB     | ########## | 100% 

altair-4.1.0         | 614 KB    |            |   0% 
altair-4.1.0         | 614 KB    | ########## | 100% 

folium-0.5.0         | 45 KB     |            |   0% 
folium-0.5.0         | 45 KB     | ########## | 100% 
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done
```

```
Libraries imported.
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

# All requested packages already installed.

Libraries imported.
```

Data Collection

In [7]:

```
# Use pandas to import data from both California and Oregon
df_All = pd.read_csv('zip_code_database_CA.csv')

df_All['latitude']=pd.to_numeric(df_All['latitude'])
df_All['longitude']=pd.to_numeric(df_All['longitude'])
df_All['irs_estimated_population_2015']=pd.to_numeric(df_All['irs_estimated_population_2015'])

#df_All.head()
df_CA = df_All.loc[df_All['area_codes']=='951',:]
print(df_CA.shape)
#df_OR = df_All.loc[(df_All['area_codes']=='503' & df_All['area_codes']=='971'),:]
df_OR = df_All.loc[(df_All['area_codes']=='503') | (df_All['area_codes']=='971'), :]
print(df_OR.shape)
```

```
(48, 15)
(118, 15)
```

In [8]:

```
df_CA.head()
```

Out[8]:

	zip	type	decommissioned	primary_city	acceptable_cities	unacceptable_cities
38625	91752	STANDARD	0	Mira Loma	Eastvale	NaN
38863	92220	STANDARD	0	Banning	NaN	NaN
39012	92502	PO BOX	0	Riverside	NaN	NaN
39019	92509	STANDARD	0	Riverside	Jurupa Valley	Rubidoux
39020	92513	PO BOX	0	Riverside	NaN	NaN

In [9]:

```
df_OR.head()
```

Out[9]:

	zip	type	decommissioned	primary_city	acceptable_cities	unacceptable_cities
41134	97002	STANDARD	0	Aurora	NaN	Butteville
41136	97004	STANDARD	0	Beavercreek	NaN	NaN
41141	97009	STANDARD	0	Boring	Damascus	NaN
41142	97010	PO BOX	0	Bridal Veil	NaN	NaN
41144	97013	STANDARD	0	Canby	NaN	Barlow

In [10]:

```
# Oregon Latitude 44.1156, Longitude -120.5148
# California Latitude 37.2461, Longitude -119.6104
```

In [11]:

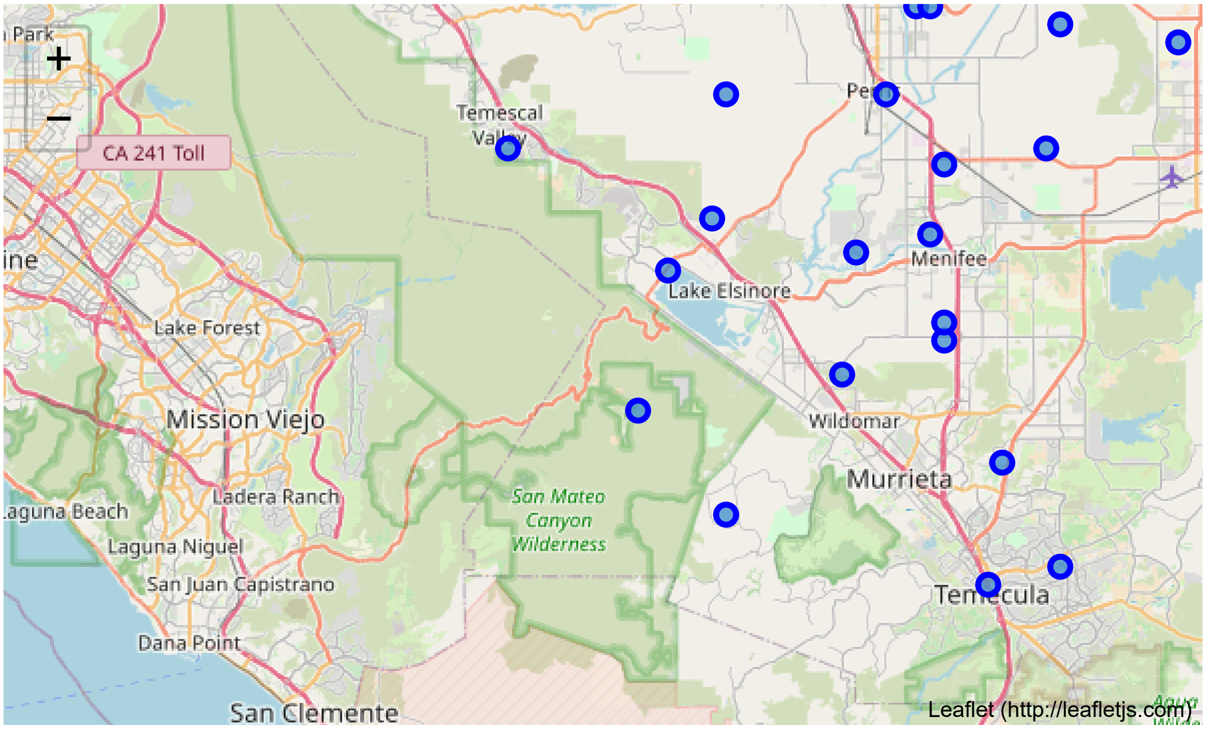
```
# Setup the Foursquare API information

CLIENT_ID = 'BRBDGCKR00L1JCYN5RD3ZDJ2BNOJ5YOVAN3F2KZNVVUVB3VW' # your Foursqua
re ID
CLIENT_SECRET = '5N1FQNDEEVF0LSK4A1SFM41P4KFSGECE1T23QNK20K1SD4PR' # your Four
square Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: BRBDGCKR00L1JCYN5RD3ZDJ2BNOJ5YOVAN3F2KZNVVUVB3VW
CLIENT_SECRET:5N1FQNDEEVF0LSK4A1SFM41P4KFSGECE1T23QNK20K1SD4PR
```

In [12]:

```
# create map of Temecula using latitude and longitude values

map_temecula = folium.Map(location=[33.4934, -117.14879], zoom_start=10)

# add markers to map
for lat, lng, primCity, popul in zip(df_CA['latitude'], df_CA['longitude'], df_CA['primary_city'], df_CA['irs_estimated_population_2015']):
    label = '{}, {}'.format(primCity, popul)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_temecula)

map_temecula
```

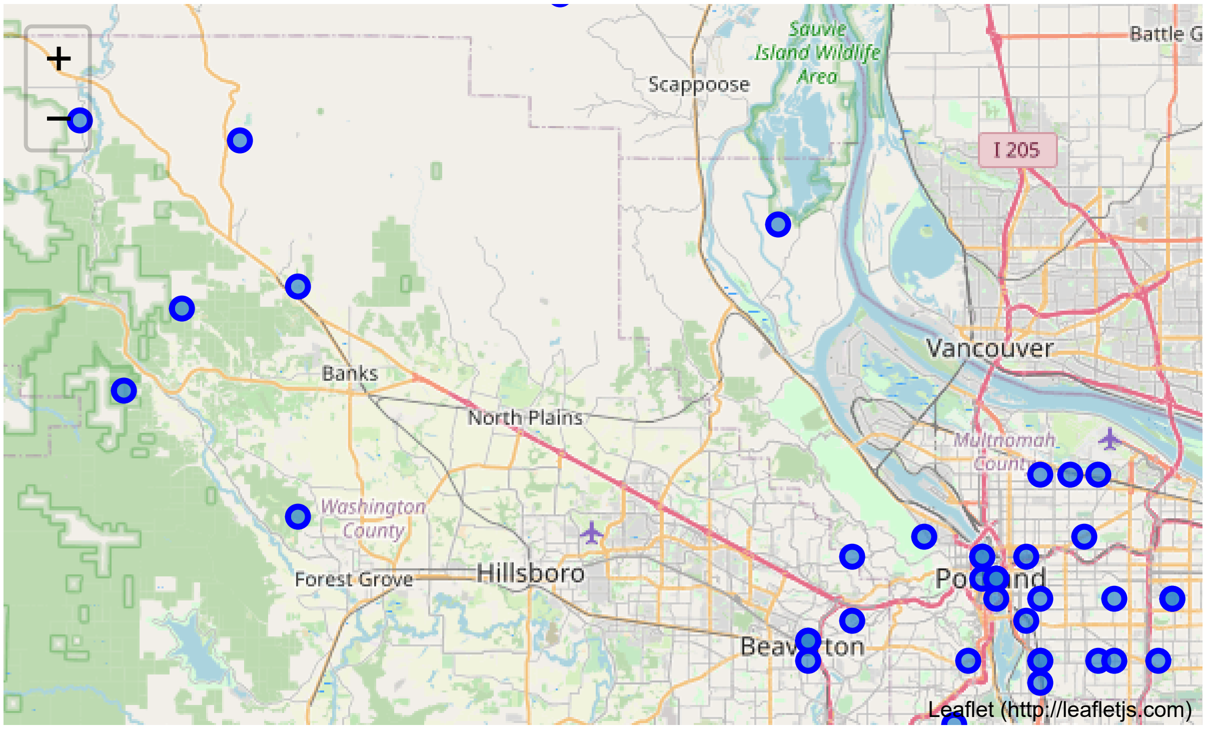
Out[12]:



In [13]:

```
# create map of Portland using latitude and longitude values

map_portland = folium.Map(location=[45.51179, -122.67563], zoom_start=10)

# add markers to map
for lat, lng, primCity, popul in zip(df_OR['latitude'], df_OR['longitude'], df
_OR['primary_city'], df_OR['irs_estimated_population_2015']):
    label = '{}, {}'.format(primCity, popul)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_portland)

map_portland
```

Out[13]:



In [14]:

```
LIMIT = 100 # limit of number of venues returned by Foursquare API
radius = 500 # define radius
```

In [15]:

```
# Create function to collect venue information from all neighborhoods in Temec
ula & Portland
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&clie
nt_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item
in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [16]:

```python
# Collect the information from Foursquare for Temecula neighborhoods
temecula_venues = getNearbyVenues(df_CA['primary_city'],
                                   df_CA['latitude'],
                                   df_CA['longitude']
                                  )
```

```
Mira Loma
Banning
Riverside
Riverside
Riverside
Riverside
Riverside
Riverside
Riverside
March Air Reserve Base
Riverside
Riverside
Riverside
Lake Elsinore
Lake Elsinore
Lake Elsinore
Aguanga
Anza
Hemet
Homeland
Idyllwild
Moreno Valley
Moreno Valley
Moreno Valley
Moreno Valley
Moreno Valley
Moreno Valley
Moreno Valley
Murrieta
Murrieta
Murrieta
Nuevo
Perris
Perris
Perris
San Jacinto
San Jacinto
Menifee
Sun City
Sun City
Sun City
Temecula
Temecula
Wildomar
Perris
Corona
Corona
Corona
```

In [17]:

```
# Collect the information from Foursquare for Portland neighborhoods
portland_venues = getNearbyVenues(df_OR['primary_city'],
                                   df_OR['latitude'],
                                   df_OR['longitude']
                                  )
```

```
Aurora
Beavercreek
Boring
Bridal Veil
Canby
Clackamas
Colton
Columbia City
Corbett
Donald
Eagle Creek
Estacada
Fairview
Gervais
Government Camp
Lake Oswego
Lake Oswego
Marylhurst
Molalla
Mulino
Warren
Deer Island
Sandy
Scappoose
Troutdale
Beaverton
Beaverton
Beaverton
Beaverton
Happy Valley
Damascus
Arch Cape
Banks
Beaver
Buxton
Cannon Beach
Carlton
Dundee
Forest Grove
Gales Creek
Garibaldi
Hammond
Hebo
Hillsboro
Manning
Lafayette
Manzanita
Oceanside
Pacific City
Sherwood
Netarts
Timber
Tolovana Park
Warrenton
Wheeler
Neskowin
Portland
```

```
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Portland
Salem
Salem
Keizer
Salem
Salem
Salem
Salem
Salem
Salem
Salem
Salem
Dallas
Detroit
Grand Ronde
Independence
Lyons
Rickreall
```

```
Saint Benedict
Scio
Stayton
Mehama
```

In [18]:
```
portland_venues.head()
```

Out[18]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Aurora	45.23	-122.80	Rome Transportation LLC	45.231671	-122.801974	Building
1	Aurora	45.23	-122.80	XFINITY Store by Comcast	45.228450	-122.803878	Business Service
2	Aurora	45.23	-122.80	Carpet Cleaning Wilsonville	45.229599	-122.805369	Carpet Store
3	Aurora	45.23	-122.80	Sunrise Custom Painting	45.225655	-122.800477	Paintball Field
4	Bridal Veil	45.58	-122.11	Cousin's Restaurant & Saloon	45.579640	-122.111006	American Restaurant

In [19]:
```
temecula_venues.head()
```

Out[19]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Mira Loma	33.99	-117.53	Party Productions	33.987651	-117.534519	Food
1	Banning	33.95	-116.83	Malki Museum	33.949280	-116.825279	Museum
2	Riverside	33.98	-117.37	The Old Spaghetti Factory	33.980494	-117.367675	Italian Restaurant
3	Riverside	33.98	-117.37	Riverside Municipal Auditorium	33.982062	-117.370789	Performing Arts Venue
4	Riverside	33.98	-117.37	Romano's Riverside	33.981286	-117.372714	Italian Restaurant

In [20]:

```
# Number of venues returned for each neighborhood
temecula_venues.groupby('Neighborhood').count()
```

Out[20]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Banning	1	1	1	1	1	1
Corona	8	8	8	8	8	8
Hemet	3	3	3	3	3	3
Homeland	1	1	1	1	1	1
Idyllwild	1	1	1	1	1	1
Lake Elsinore	14	14	14	14	14	14
March Air Reserve Base	2	2	2	2	2	2
Menifee	2	2	2	2	2	2
Mira Loma	1	1	1	1	1	1
Moreno Valley	36	36	36	36	36	36
Murrieta	14	14	14	14	14	14
Perris	24	24	24	24	24	24
Riverside	173	173	173	173	173	173
San Jacinto	3	3	3	3	3	3
Sun City	4	4	4	4	4	4
Temecula	52	52	52	52	52	52

In [21]:

```
# Number of venues returned for each neighborhood
portland_venues.groupby('Neighborhood').count()
```

Out[21]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Arch Cape	1	1	1	1	1	1
Aurora	4	4	4	4	4	4
Beaverton	207	207	207	207	207	207
Bridal Veil	1	1	1	1	1	1
Canby	1	1	1	1	1	1
Cannon Beach	50	50	50	50	50	50
Clackamas	4	4	4	4	4	4
Columbia City	3	3	3	3	3	3
Damascus	9	9	9	9	9	9
Fairview	3	3	3	3	3	3
Forest Grove	1	1	1	1	1	1
Grand Ronde	1	1	1	1	1	1
Happy Valley	2	2	2	2	2	2
Hillsboro	1	1	1	1	1	1
Keizer	2	2	2	2	2	2
Lafayette	4	4	4	4	4	4
Lake Oswego	29	29	29	29	29	29
Manzanita	1	1	1	1	1	1
Marylhurst	18	18	18	18	18	18
Mehama	2	2	2	2	2	2
Mulino	1	1	1	1	1	1
Netarts	3	3	3	3	3	3
Oceanside	1	1	1	1	1	1
Portland	974	974	974	974	974	974
Rickreall	1	1	1	1	1	1
Salem	50	50	50	50	50	50
Sherwood	4	4	4	4	4	4
Timber	1	1	1	1	1	1
Troutdale	2	2	2	2	2	2
Wheeler	9	9	9	9	9	9

In [22]:

```
print('There are {} uniques categories for Temecula.'.format(len(temecula_venues['Venue Category'].unique())))
print('There are {} uniques categories for Portland.'.format(len(portland_venues['Venue Category'].unique())))
```

```
There are 121 uniques categories for Temecula.
There are 230 uniques categories for Portland.
```

In [23]:

```
# one hot encoding - Temecula
temecula_onehot = pd.get_dummies(temecula_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
temecula_onehot['Neighborhood'] = temecula_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [temecula_onehot.columns[-1]] + list(temecula_onehot.columns[:-1])
temecula_onehot = temecula_onehot[fixed_columns]
print(temecula_onehot.shape)
temecula_onehot.head()
```

```
(339, 122)
```

Out[23]:

	Neighborhood	ATM	Accessories Store	Advertising Agency	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store
0	Mira Loma	0	0	0	0	0	0	0	0
1	Banning	0	0	0	0	0	0	0	0
2	Riverside	0	0	0	0	0	0	0	0
3	Riverside	0	0	0	0	0	0	0	0
4	Riverside	0	0	0	0	0	0	0	0

In [24]:

```
# one hot encoding - Portland
portland_onehot = pd.get_dummies(portland_venues[['Venue Category']], prefix=
"", prefix_sep="")

# add neighborhood column back to dataframe
portland_onehot['Neighborhood'] = portland_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [portland_onehot.columns[-1]] + list(portland_onehot.columns[:
-1])
portland_onehot = portland_onehot[fixed_columns]
print(portland_onehot.shape)
portland_onehot.head()
```

```
(1390, 231)
```

Out[24]:

	Neighborhood	ATM	Airport	Airport Terminal	American Restaurant	Antique Shop	Art Gallery	Arts & Crafts Store	Asian Restaurant	Athl & Sp
0	Aurora	0	0	0	0	0	0	0	0	
1	Aurora	0	0	0	0	0	0	0	0	
2	Aurora	0	0	0	0	0	0	0	0	
3	Aurora	0	0	0	0	0	0	0	0	
4	Bridal Veil	0	0	0	1	0	0	0	0	

In [25]:

```
temecula_grouped = temecula_onehot.groupby('Neighborhood').mean().reset_index
()
print('Temecula:')
print(temecula_grouped.shape)
portland_grouped = portland_onehot.groupby('Neighborhood').mean().reset_index
()
print("Portland:")
print(portland_grouped.shape)
```

```
Temecula:
(16, 122)
Portland:
(30, 231)
```

In [26]:

```
num_top_venues = 5

for hood in temecula_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = temecula_grouped[temecula_grouped['Neighborhood'] == hood].T.reset_
index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).hea
d(num_top_venues))
    print('\n')
```

```
----Banning----
                   venue  freq
0                 Museum   1.0
1                    ATM   0.0
2              Hotel Bar   0.0
3            Pizza Place   0.0
4  Performing Arts Venue   0.0


----Corona----
               venue  freq
0  Convenience Store  0.50
1              Diner  0.25
2               Park  0.25
3                Pub  0.00
4        Pizza Place  0.00


----Hemet----
            venue  freq
0  Sandwich Place  0.33
1   Grocery Store  0.33
2    Liquor Store  0.33
3             ATM  0.00
4  Massage Studio  0.00


----Homeland----
                   venue  freq
0           Home Service   1.0
1                    ATM   0.0
2                 Market   0.0
3  Performing Arts Venue   0.0
4                   Park   0.0


----Idyllwild----
                   venue  freq
0                    Spa   1.0
1                    ATM   0.0
2                   Pool   0.0
3  Performing Arts Venue   0.0
4                   Park   0.0


----Lake Elsinore----
                       venue  freq
0             Sandwich Place  0.14
1       Fast Food Restaurant  0.14
2  Cajun / Creole Restaurant  0.07
3                        Bar  0.07
4        Fried Chicken Joint  0.07


----March Air Reserve Base----
                  venue  freq
0           Coffee Shop   0.5
```

```
1  Fast Food Restaurant   0.5
2                   ATM   0.0
3           Men's Store   0.0
4           Pizza Place   0.0


----Menifee----
                   venue  freq
0          Garden Center   0.5
1      Mobile Phone Shop   0.5
2                   Pool   0.0
3  Performing Arts Venue   0.0
4                   Park   0.0


----Mira Loma----
                   venue  freq
0                   Food   1.0
1                    ATM   0.0
2                   Pool   0.0
3  Performing Arts Venue   0.0
4                   Park   0.0


----Moreno Valley----
                  venue  freq
0    Mexican Restaurant  0.08
1          Dessert Shop  0.06
2                Bakery  0.06
3         Deli / Bodega  0.06
4  Fast Food Restaurant  0.06


----Murrieta----
                venue  freq
0  Mexican Restaurant  0.14
1         Coffee Shop  0.07
2  Frozen Yogurt Shop  0.07
3      Baseball Field  0.07
4       Moving Target  0.07


----Perris----
                  venue  freq
0                  Park  0.12
1    Mexican Restaurant  0.08
2  Fast Food Restaurant  0.08
3           Pizza Place  0.08
4                 Hotel  0.04


----Riverside----
                  venue  freq
0        Clothing Store  0.06
1  Fast Food Restaurant  0.05
2   American Restaurant  0.04
3    Mexican Restaurant  0.04
```

```
4        Cosmetics Shop  0.03


----San Jacinto----
            venue  freq
0            Park  0.33
1            Food  0.33
2    Liquor Store  0.33
3             ATM  0.00
4  Massage Studio  0.00


----Sun City----
             venue  freq
0  Automotive Shop  0.50
1   Rental Service  0.25
2  Laundry Service  0.25
3              ATM  0.00
4   Massage Studio  0.00


----Temecula----
                 venue  freq
0                Hotel  0.10
1  American Restaurant  0.10
2   Mexican Restaurant  0.10
3   Italian Restaurant  0.06
4       Breakfast Spot  0.06
```

In [27]:

```
for hood in portland_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp1 = portland_grouped[portland_grouped['Neighborhood'] == hood].T.reset_index()
    temp1.columns = ['venue','freq']
    temp1 = temp1.iloc[1:]
    temp1['freq'] = temp1['freq'].astype(float)
    temp1 = temp1.round({'freq': 2})
    print(temp1.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Arch Cape----
         venue  freq
0       Tunnel   1.0
1          ATM   0.0
2  Pastry Shop   0.0
3  Music Store   0.0
4  Music Venue   0.0


----Aurora----
              venue  freq
0          Building  0.25
1  Business Service  0.25
2   Paintball Field  0.25
3      Carpet Store  0.25
4               ATM  0.00


----Beaverton----
                venue  freq
0   Korean Restaurant  0.06
1         Coffee Shop  0.05
2  Mexican Restaurant  0.04
3      Sandwich Place  0.04
4       Grocery Store  0.04


----Bridal Veil----
                   venue  freq
0    American Restaurant   1.0
1                    ATM   0.0
2  Performing Arts Venue   0.0
3            Music Venue   0.0
4             Nail Salon   0.0


----Canby----
                     venue  freq
0              Cheese Shop   1.0
1    Performing Arts Venue   0.0
2              Music Venue   0.0
3               Nail Salon   0.0
4  New American Restaurant   0.0


----Cannon Beach----
                 venue  freq
0  American Restaurant  0.06
1           Restaurant  0.06
2        Women's Store  0.04
3          Candy Store  0.04
4          Art Gallery  0.04


----Clackamas----
                     venue  freq
0                  Brewery  0.25
```

```
1           Clothing Store  0.25
2                      Bar  0.25
3         Basketball Court  0.25
4  New American Restaurant  0.00


----Columbia City----
            venue  freq
0            Park  0.33
1  History Museum  0.33
2           Beach  0.33
3             ATM  0.00
4     Music Store  0.00


----Damascus----
            venue  freq
0   Grocery Store  0.22
1     Coffee Shop  0.11
2             Pub  0.11
3  Sandwich Place  0.11
4  Ice Cream Shop  0.11


----Fairview----
               venue  freq
0      Moving Target  0.33
1  Recreation Center  0.33
2   Business Service  0.33
3        Pastry Shop  0.00
4        Music Venue  0.00


----Forest Grove----
           venue  freq
0           Farm   1.0
1            ATM   0.0
2  Moving Target   0.0
3    Music Venue   0.0
4     Nail Salon   0.0


----Grand Ronde----
         venue  freq
0        River   1.0
1          ATM   0.0
2  Pastry Shop   0.0
3  Music Store   0.0
4  Music Venue   0.0


----Happy Valley----
              venue  freq
0  Insurance Office   0.5
1      Home Service   0.5
2       Pastry Shop   0.0
3       Music Store   0.0
```

```
4       Music Venue   0.0


----Hillsboro----
           venue  freq
0           Farm   1.0
1            ATM   0.0
2  Moving Target   0.0
3    Music Venue   0.0
4     Nail Salon   0.0


----Keizer----
         venue  freq
0         Park   0.5
1          Pub   0.5
2          ATM   0.0
3  Music Store   0.0
4  Music Venue   0.0


----Lafayette----
             venue  freq
0     Antique Shop  0.25
1      Gas Station  0.25
2      Coffee Shop  0.25
3  Bed & Breakfast  0.25
4              ATM  0.00


----Lake Oswego----
           venue  freq
0  Grocery Store  0.07
1    Pizza Place  0.07
2           Bank  0.07
3    Coffee Shop  0.07
4   Burger Joint  0.03


----Manzanita----
         venue  freq
0   Campground   1.0
1          ATM   0.0
2  Pastry Shop   0.0
3  Music Venue   0.0
4   Nail Salon   0.0


----Marylhurst----
                venue  freq
0  Chinese Restaurant  0.11
1  Italian Restaurant  0.06
2              Bakery  0.06
3        Optical Shop  0.06
4  Mexican Restaurant  0.06
```

```
----Mehama----
                   venue  freq
0          Deli / Bodega   0.5
1    American Restaurant   0.5
2  Performing Arts Venue   0.0
3            Music Venue   0.0
4             Nail Salon   0.0


----Mulino----
              venue  freq
0  Business Service   1.0
1               ATM   0.0
2       Pastry Shop   0.0
3       Music Venue   0.0
4        Nail Salon   0.0


----Netarts----
         venue  freq
0      RV Park  0.67
1          Bar  0.33
2          ATM  0.00
3  Pastry Shop  0.00
4  Music Venue  0.00


----Oceanside----
             venue  freq
0  Vacation Rental   1.0
1              ATM   0.0
2      Pastry Shop   0.0
3      Music Store   0.0
4      Music Venue   0.0


----Portland----
         venue  freq
0  Coffee Shop  0.06
1   Food Truck  0.06
2        Hotel  0.04
3         Park  0.04
4   Restaurant  0.03


----Rickreall----
         venue  freq
0     Vineyard   1.0
1          ATM   0.0
2  Pastry Shop   0.0
3  Music Store   0.0
4  Music Venue   0.0


----Salem----
                 venue  freq
0          Pizza Place  0.06
```

```
1  American Restaurant  0.04
2        Grocery Store  0.04
3                 Park  0.04
4          Coffee Shop  0.04


----Sherwood----
            venue  freq
0            Farm  0.25
1  Cosmetics Shop  0.25
2      Bike Trail  0.25
3      Playground  0.25
4             ATM  0.00


----Timber----
           venue  freq
0         Forest   1.0
1            ATM   0.0
2  Moving Target   0.0
3    Music Venue   0.0
4     Nail Salon   0.0


----Troutdale----
         venue  freq
0    Disc Golf   0.5
1         Farm   0.5
2          ATM   0.0
3  Pastry Shop   0.0
4  Music Venue   0.0


----Wheeler----
              venue  freq
0            Bakery  0.33
1      Antique Shop  0.11
2  Business Service  0.11
3        Restaurant  0.11
4              Café  0.11
```

In [28]:

```
# Insert the most common venues data into a pandas dataframe
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]
```

In [29]:

```
# The following code creates a new dataframe that will display the top 10 venu
es for each neighborhood
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
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
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = temecula_grouped['Neighborhood']

for ind in np.arange(temecula_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(teme
cula_grouped.iloc[ind, :], num_top_venues)

print('For Temecula: ')
neighborhoods_venues_sorted.head(10)
```

```
For Temecula:
```

Out[29]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	
0	Banning	Museum	Women's Store	Convenience Store	Deli / Bodega	Department Store	Dessert Shop	
1	Corona	Convenience Store	Park	Diner	Women's Store	Deli / Bodega	Department Store	
2	Hemet	Liquor Store	Grocery Store	Sandwich Place	Fried Chicken Joint	Cosmetics Shop	Deli / Bodega	D
3	Homeland	Home Service	Hot Dog Joint	Cosmetics Shop	Deli / Bodega	Department Store	Dessert Shop	
4	Idyllwild	Spa	Women's Store	Fried Chicken Joint	Cosmetics Shop	Deli / Bodega	Department Store	
5	Lake Elsinore	Sandwich Place	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Convenience Store	Bar	F
6	March Air Reserve Base	Coffee Shop	Fast Food Restaurant	Women's Store	Convenience Store	Deli / Bodega	Department Store	
7	Menifee	Mobile Phone Shop	Garden Center	Women's Store	Fried Chicken Joint	Deli / Bodega	Department Store	
8	Mira Loma	Food	Women's Store	Hot Dog Joint	Cosmetics Shop	Deli / Bodega	Department Store	
9	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bakery	A

In [30]:

```
# The following code creates a new dataframe that will display the top 10 venues for each neighborhood
#num_top_venues = 10

indicators1 = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns1 = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns1.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns1.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted1 = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted1['Neighborhood'] = portland_grouped['Neighborhood']

for ind in np.arange(portland_grouped.shape[0]):
    neighborhoods_venues_sorted1.iloc[ind, 1:] = return_most_common_venues(portland_grouped.iloc[ind, :], num_top_venues)

print('For Portland: ')
neighborhoods_venues_sorted1.head(10)
```

```
For Portland:
```

Out[30]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue
0	Arch Cape	Tunnel	Yoga Studio	Donut Shop	Food Truck	Food Court	Food	Fondue Restaurant
1	Aurora	Carpet Store	Building	Business Service	Paintball Field	Yoga Studio	Food Court	Food
2	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery	Wings Joint
3	Bridal Veil	American Restaurant	Yoga Studio	Eastern European Restaurant	Forest	Food Truck	Food Court	Food
4	Canby	Cheese Shop	Yoga Studio	Eastern European Restaurant	Forest	Food Truck	Food Court	Food
5	Cannon Beach	American Restaurant	Restaurant	Toy / Game Store	Art Gallery	Bakery	Café	Furniture / Home Store
6	Clackamas	Basketball Court	Clothing Store	Bar	Brewery	Yoga Studio	Fountain	Forest
7	Columbia City	Park	History Museum	Beach	Eastern European Restaurant	Food Truck	Food Court	Food
8	Damascus	Grocery Store	Home Service	Business Service	Sandwich Place	Coffee Shop	Ice Cream Shop	Pub
9	Fairview	Moving Target	Recreation Center	Business Service	Yoga Studio	Eastern European Restaurant	Food Court	Food

In [31]:

```python
# set number of clusters
kclusters = 5

temecula_grouped_clustering = temecula_grouped.drop('Neighborhood', 1)

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

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

Out[31]:

```
array([2, 1, 1, 3, 4, 1, 1, 1, 0, 1])
```

In [32]:
```
# add clustering labels for new dataframe to include the cluster and its top 1
0 venues for Temecula
# currently commented out since I have run this cell so many times
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
# neighborhoods_venues_sorted.head()
temecula_merged = df_CA[['primary_city', 'latitude', 'longitude']]
temecula_merged.rename(columns = {'primary_city':'Neighborhood'}, inplace = Tr
ue)
temecula_merged.head()


# merge temecula_grouped with df_CA to add latitude/longitude for each neighbo
rhood
neighborhoods_venues_sorted = neighborhoods_venues_sorted.join(temecula_merged
.set_index('Neighborhood'), on='Neighborhood')
neighborhoods_venues_sorted.head()
```

```
C:\Users\gnetherton\Anaconda3\lib\site-packages\pandas\core\frame.py:4133: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

Out[32]:

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
0	2	Banning	Museum	Women's Store	Convenience Store	Deli / Bodega	Department Store	Dessert Shop
1	1	Corona	Convenience Store	Park	Diner	Women's Store	Deli / Bodega	Department Store
1	1	Corona	Convenience Store	Park	Diner	Women's Store	Deli / Bodega	Department Store
1	1	Corona	Convenience Store	Park	Diner	Women's Store	Deli / Bodega	Department Store
2	1	Hemet	Liquor Store	Grocery Store	Sandwich Place	Fried Chicken Joint	Cosmetics Shop	Deli / Bodega

In [33]:
```
portland_grouped_clustering = portland_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans2 = KMeans(n_clusters=kclusters, random_state=0).fit(portland_grouped_cl
ustering)

# check cluster labels generated for each row in the dataframe
kmeans2.labels_[0:10]
```

Out[33]:
```
array([4, 0, 0, 2, 0, 0, 0, 0, 0, 0])
```

In [34]:
```
# add clustering labels for new dataframe to include the cluster and its top 10 venues for Portland
# currently commented out since I have run this cell so many times
neighborhoods_venues_sorted1.insert(0, 'Cluster Labels', kmeans2.labels_)
# neighborhoods_venues_sorted1.head()
portland_merged = df_OR[['primary_city', 'latitude', 'longitude']]
portland_merged.rename(columns = {'primary_city':'Neighborhood'}, inplace = True)
#portland_merged.head()


# merge portland_merged with df_CA to add latitude/longitude for each neighborhood
neighborhoods_venues_sorted1 = neighborhoods_venues_sorted1.join(portland_merged.set_index('Neighborhood'), on='Neighborhood')
neighborhoods_venues_sorted1.head()
```

```
C:\Users\gnetherton\Anaconda3\lib\site-packages\pandas\core\frame.py:4133: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

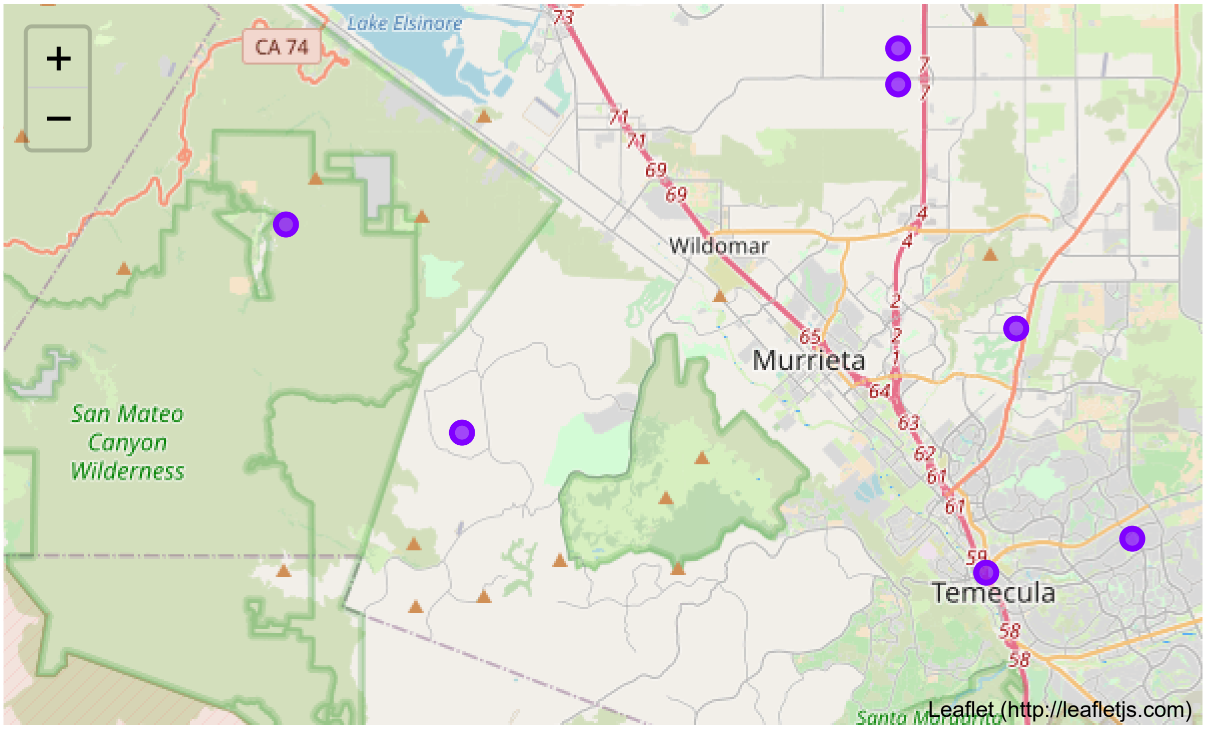
Out[34]:

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Com V
0	4	Arch Cape	Tunnel	Yoga Studio	Donut Shop	Food Truck	Food Court	Food	Fo Resta
1	0	Aurora	Carpet Store	Building	Business Service	Paintball Field	Yoga Studio	Food Court	
2	0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery	W
2	0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery	W
2	0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery	W

In [35]:

```
map_clustersCA = folium.Map(location=[33.4934, -117.14879], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]



# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(neighborhoods_venues_sorted['latitude'], nei
ghborhoods_venues_sorted['longitude'], \
                                  neighborhoods_venues_sorted['Neighborhood'],
neighborhoods_venues_sorted['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=Tru
e)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clustersCA)

map_clustersCA
```

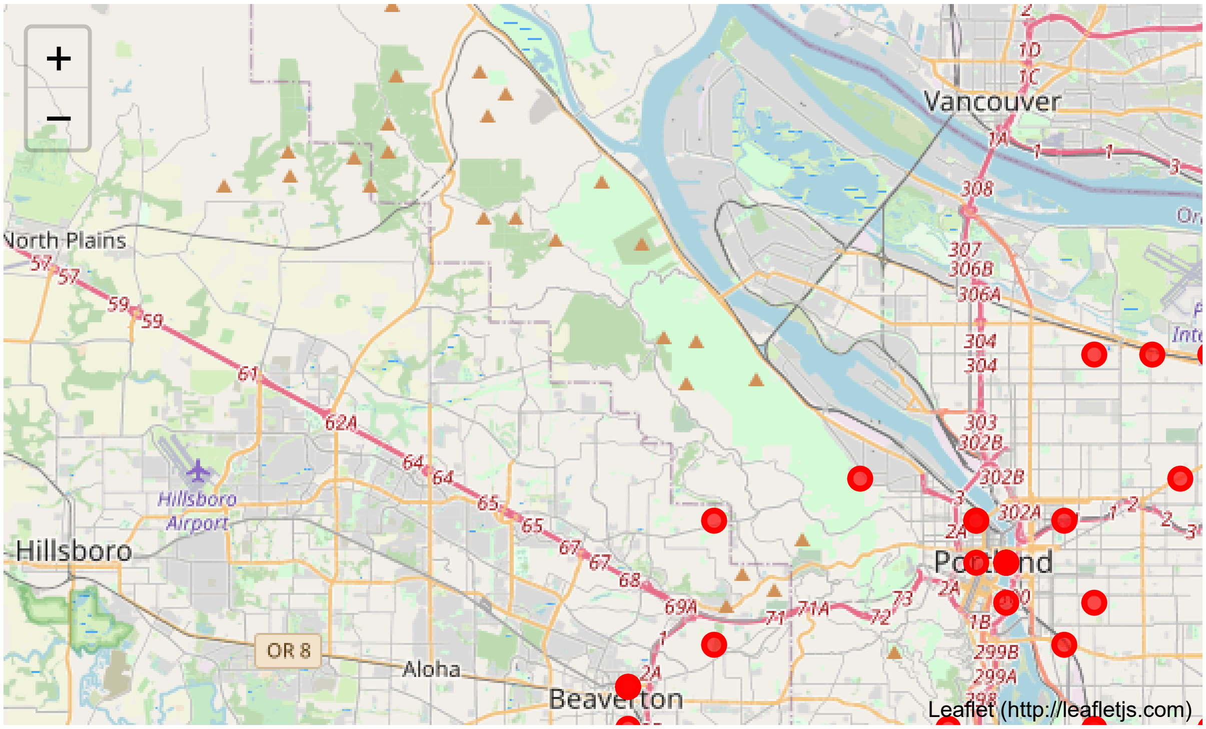
Out[35]:



In [36]:

```
map_clustersOR = folium.Map(location=[45.51179, -122.67563], zoom_start=11)

# set color scheme for the clusters
# x = np.arange(kclusters)
# ys = [i + x + (i*x)**2 for i in range(kclusters)]
# colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
# rainbow = [colors.rgb2hex(i) for i in colors_array]

# temecula_merged.rename(columns = {'primary_city':'Neighborhood'}, inplace = True)

# add markers to the map
markers_colors2 = []
for lat, lon, poi, cluster in zip(neighborhoods_venues_sorted1['latitude'], neighborhoods_venues_sorted1['longitude'], \
                                  neighborhoods_venues_sorted1['Neighborhood'], neighborhoods_venues_sorted1['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clustersOR)

map_clustersOR
```

Out[36]:



In [88]:
```
filter1 = neighborhoods_venues_sorted1['1st Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted1['1st Most Common Venue'].str.cont
ains('Joint')
filter2 = neighborhoods_venues_sorted1['2nd Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted1['2nd Most Common Venue'].str.cont
ains('Joint')
filter3 = neighborhoods_venues_sorted1['1st Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted1['1st Most Common Venue'].str.cont
ains('Joint')
docs_OR = neighborhoods_venues_sorted1.where(filter1 | filter2 | filter3)
docs_OR.dropna()
```

Out[88]:

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
2	0.0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery
2	0.0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery
2	0.0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery
2	0.0	Beaverton	Korean Restaurant	Coffee Shop	Sandwich Place	Mexican Restaurant	Grocery Store	Bakery
3	2.0	Bridal Veil	American Restaurant	Yoga Studio	Eastern European Restaurant	Forest	Food Truck	Food Court
5	0.0	Cannon Beach	American Restaurant	Restaurant	Toy / Game Store	Art Gallery	Bakery	Café
18	0.0	Marylhurst	Chinese Restaurant	Ramen Restaurant	Martial Arts Dojo	Optical Shop	Mexican Restaurant	Coffee Shop
19	2.0	Mehama	Deli / Bodega	American Restaurant	Yoga Studio	Electronics Store	Forest	Food Truck

Orgeon's table suggests that I would prefer to live in Beaverton, OR rather than Portland proper. Southern CA is looking a bit more appealing.

In [63]:

```
filter1a = neighborhoods_venues_sorted['3rd Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted['3rd Most Common Venue'].str.conta
ins('Joint')
filter2a = neighborhoods_venues_sorted['2nd Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted['2nd Most Common Venue'].str.conta
ins('Joint')
filter3a = neighborhoods_venues_sorted['1st Most Common Venue'].str.contains(
'Restaurant') | neighborhoods_venues_sorted['1st Most Common Venue'].str.conta
ins('Joint')
docs_CA = neighborhoods_venues_sorted.where(filter1a | filter2a | filter3a)
docs_CA.dropna()
```

Out[63]:

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th M Comm Ver
3	3.0	Homeland	Home Service	Hot Dog Joint	Cosmetics Shop	Deli / Bodega	Department Store	Dess Sh
4	4.0	Idyllwild	Spa	Women's Store	Fried Chicken Joint	Cosmetics Shop	Deli / Bodega	Departm St
5	1.0	Lake Elsinore	Sandwich Place	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Convenience Store	I
5	1.0	Lake Elsinore	Sandwich Place	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Convenience Store	I
5	1.0	Lake Elsinore	Sandwich Place	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Convenience Store	I
6	1.0	March Air Reserve Base	Coffee Shop	Fast Food Restaurant	Women's Store	Convenience Store	Deli / Bodega	Departm St
8	0.0	Mira Loma	Food	Women's Store	Hot Dog Joint	Cosmetics Shop	Deli / Bodega	Departm St
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
9	1.0	Moreno Valley	Mexican Restaurant	Fried Chicken Joint	Ice Cream Shop	Filipino Restaurant	Fast Food Restaurant	Bak
10	1.0	Murrieta	Mexican Restaurant	Music Venue	Dessert Shop	Office	Sandwich Place	Basel Fi
10	1.0	Murrieta	Mexican Restaurant	Music Venue	Dessert Shop	Office	Sandwich Place	Basel Fi
10	1.0	Murrieta	Mexican Restaurant	Music Venue	Dessert Shop	Office	Sandwich Place	Basel Fi
11	1.0	Perris	Park	Pizza Place	Mexican Restaurant	Fast Food Restaurant	Hotel	La

	Cluster Labels	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th M Comm Ver
11	1.0	Perris	Park	Pizza Place	Mexican Restaurant	Fast Food Restaurant	Hotel	La
11	1.0	Perris	Park	Pizza Place	Mexican Restaurant	Fast Food Restaurant	Hotel	La
11	1.0	Perris	Park	Pizza Place	Mexican Restaurant	Fast Food Restaurant	Hotel	La
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
12	1.0	Riverside	Clothing Store	Fast Food Restaurant	American Restaurant	Mexican Restaurant	Cosmetics Shop	Cof Sh
15	1.0	Temecula	Hotel	Mexican Restaurant	American Restaurant	Italian Restaurant	Breakfast Spot	Su Restaur
15	1.0	Temecula	Hotel	Mexican Restaurant	American Restaurant	Italian Restaurant	Breakfast Spot	Su Restaur

The two tables above suggest that for a food-lover, Temecula is defintely worth digging deeper.

Let's evaluate it from a cultural perspective.

In [96]:
```
df_CA_arts = temecula_venues[temecula_venues['Venue Category'].str.contains('M
useum')]
df_CA_arts1 = temecula_venues[temecula_venues['Venue Category'].str.contains(
'Perform')]
df_CA_arts.append(df_CA_arts1)
```

Out[96]:

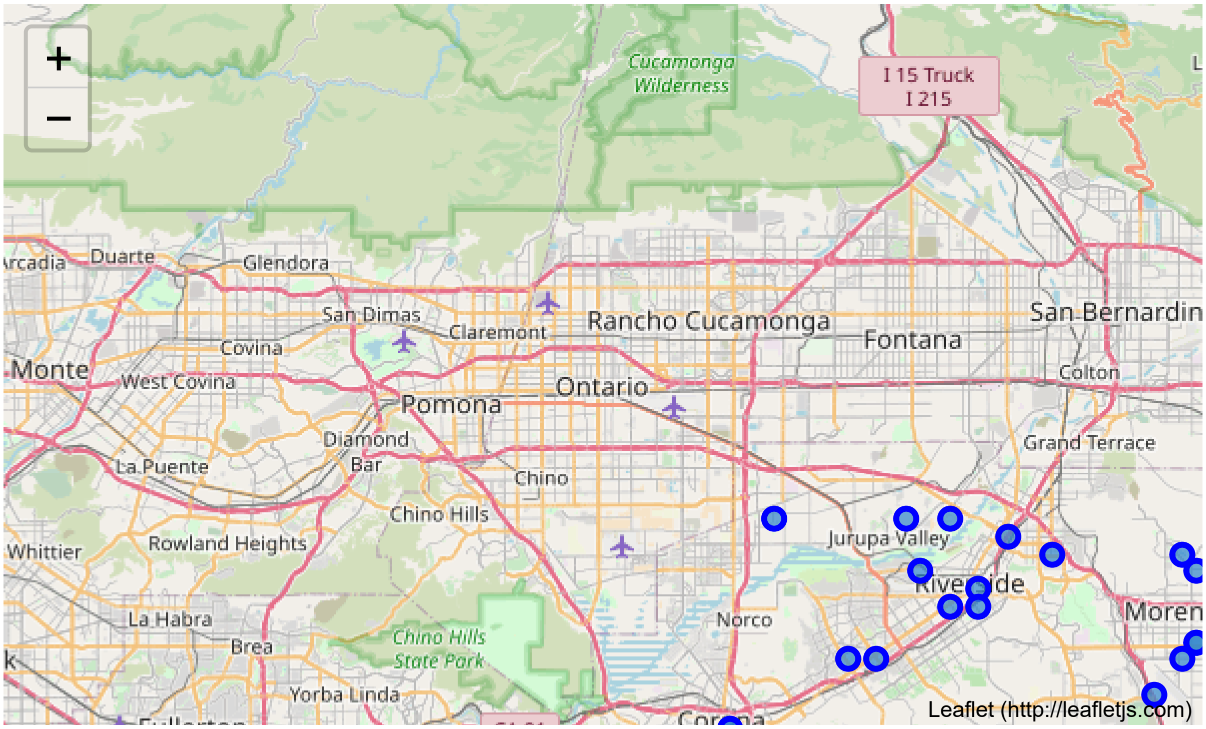
	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venu Categor
1	Banning	33.95	-116.83	Malki Museum	33.949280	-116.825279	Museun
7	Riverside	33.98	-117.37	Riverside Art Museum (RAM)	33.981682	-117.370581	Ar Museun
17	Riverside	33.98	-117.37	UCR California Museum Of Photography	33.981360	-117.374530	Ar Museun
28	Riverside	33.98	-117.37	Riverside Metropolitan Museum	33.982215	-117.372374	Museun
320	Temecula	33.50	-117.15	Temecula Valley Museum	33.497441	-117.150993	Histor Museun
3	Riverside	33.98	-117.37	Riverside Municipal Auditorium	33.982062	-117.370789	Performin Arts Venu
13	Riverside	33.98	-117.37	Riverside Life And Arts Center	33.981170	-117.371643	Performin Arts Venu
33	Riverside	33.98	-117.37	Culver Center of the Arts	33.981375	-117.374569	Performin Arts Venu

Much as the Portland data suggested I live in Beaverton, the Temecula data suggests I should live in Riverside as the food and cultural scene are more prominent in that area.

In [97]:

```
# create map of Temecula using latitude and longitude values

map_riverside = folium.Map(location=[33.946484, -117.38317], zoom_start=10)

# add markers to map
for lat, lng, primCity, popul in zip(df_CA['latitude'], df_CA['longitude'], df
_CA['primary_city'], df_CA['irs_estimated_population_2015']):
    label = '{}, {}'.format(primCity, popul)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_riverside)

map_riverside
```

Out[97]:



My new home! Riverside, CA here I come!

In [ ]: