Introduction

Philadelphia is a city of vibrant neighborhoods with each area owning a distinctive personality. For families with small children however, neighborhoods with plenty of daycares and playgrounds nearby are especially attractive. That is also a sign for kids-friendly neighborhoods.

Now we are going to explore all the neighborhoods and try to find the best ones with the most number of daycares and playgrounds. Young parents with small children will most likely be interested and be the target audience of this report.

Data

The data I was able to find is from this website: https://public.opendatasoft.com (https://public.opendatasoft.com) From there a .CSV file was downloaded which contains the list of zip codes in Philadelphia, PA as well as the Latitude and Longitude for each zip code. Considering many realestate rent / sale websites like zillow.com can comfortably search by zip codes, let's use this data as it is and next we can go straight to Foursquare data to analyze and rank by zipcodes.

Import the .CSV file and print out head()

In [1]:
```
import pandas as pd
csv = pd.read_csv('philadelphia-zip-code-latitude-and-longitude.csv', delimite
r=';')
print(csv.shape)
csv.head()
```

```
(84, 8)
```

Out[1]:

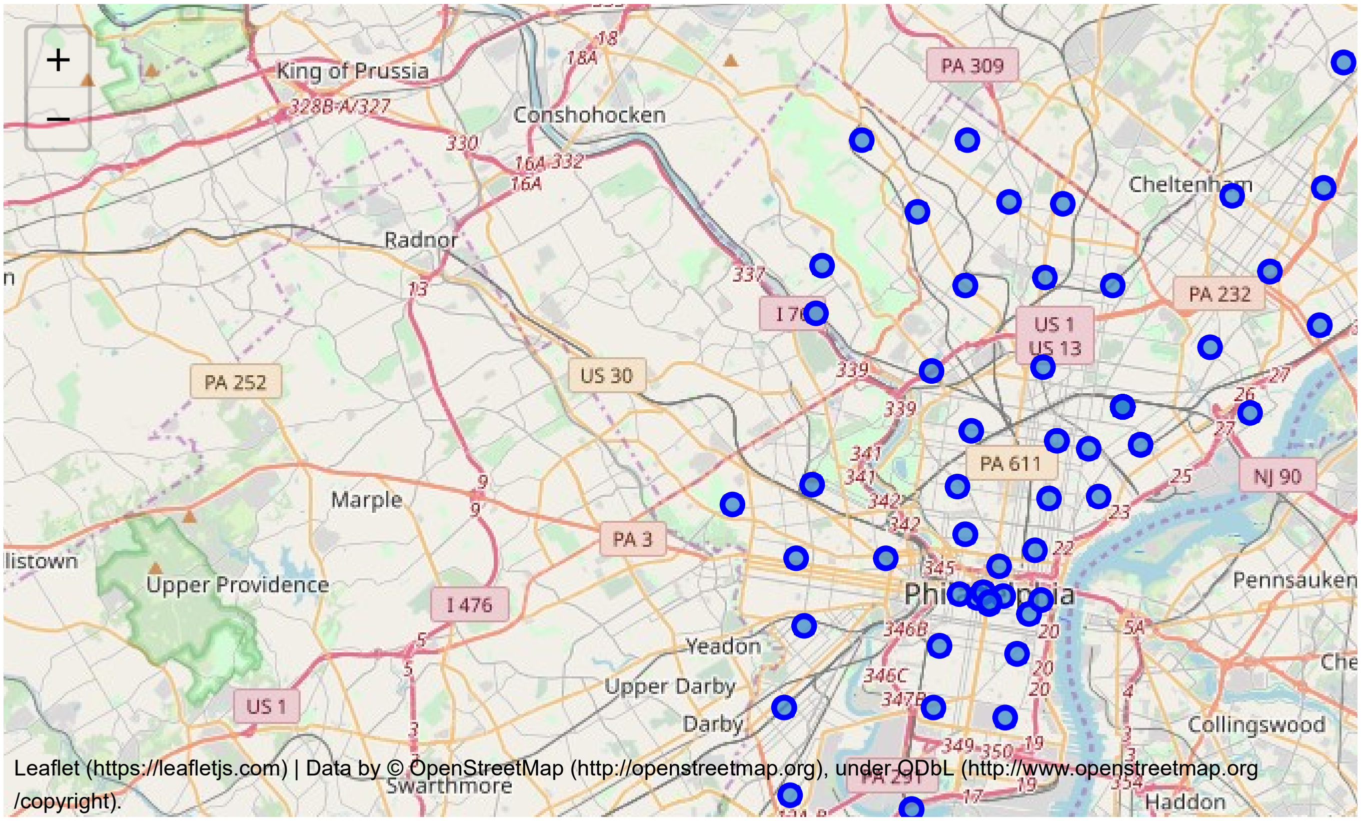
	Zip	City	State	Latitude	Longitude	Timezone	Daylight savings time flag	geopoint
0	19173	Philadelphia	PA	40.001811	-75.117870	-5	1	40.001811,-75.11787
1	19134	Philadelphia	PA	39.991712	-75.111160	-5	1	39.991712,-75.11116
2	19115	Philadelphia	PA	40.092610	-75.041180	-5	1	40.09261,-75.04118
3	19192	Philadelphia	PA	39.951112	-75.167622	-5	1	39.951112,-75.167622
4	19155	Philadelphia	PA	40.001811	-75.117870	-5	1	40.001811,-75.11787

1. Let's Map out the Neighbourhoods of Philadelphia

In [2]:

```
import folium

# Philadelphia lat/long is 39.9526, -75.1652
map_phl = folium.Map(location=[39.9526, -75.1652], zoom_start=11)

# add markers to map
for lat, lng in zip(csv['Latitude'], csv['Longitude']):
    label = '{}, {}'.format(lat, lng)
    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_phl)

map_phl
```

Out[2]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org /copyright).

Get Foursquare data

In [5]:
```
CLIENT_ID = 'J1JNSELXZ11ESAK5NVYDKRGJWHUVHX0QW3VHEOM53L03L1LS' # your Foursquare ID
CLIENT_SECRET = 'WEENNVISZHZRKUL2HJJWJIIXW0WAUEUG3MDYUOVBWJCRGDTS' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

def getNearbyVenues(zipc, latitudes, longitudes, LIMIT=100, radius=500):

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_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([(
            zipc,
            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 = ['Zip',
                  'Latitude',
                  'Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [6]:
```
import requests
phl_venues = getNearbyVenues(zipc=csv['Zip'],
                                   latitudes=csv['Latitude'],
                                   longitudes=csv['Longitude']
                                  )
```

In [7]:

```
print(phl_venues.shape)
phl_venues.head()
```

```
(1831, 7)
```

Out[7]:

	Zip	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	19173	40.001811	-75.11787	Septa Train to Trenton	40.003938	-75.117227	Light Rail Station
1	19173	40.001811	-75.11787	Classic Pizza	40.001532	-75.114627	Pizza Place
2	19173	40.001811	-75.11787	Kitchen Express	39.999921	-75.115444	Food
3	19173	40.001811	-75.11787	Hoagies Plus	40.001568	-75.114342	Deli / Bodega
4	19173	40.001811	-75.11787	Amtrak 2168	40.004719	-75.119246	Train

In [ ]:

```
#pd.options.display.max_rows = 300
#phl_venues['Venue Category'].value_counts()
```

2. Analyze Each Neighbourhood in Philadelphia

In [8]:

```
# one hot encoding
phl_onehot = pd.get_dummies(phl_venues[['Venue Category']], prefix="", prefix_se
p="")

# add neighborhood column back to dataframe
phl_onehot['Zip'] = phl_venues['Zip']

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

print(phl_onehot.shape)
print(phl_onehot.columns)
phl_onehot.head()
```

```
(1831, 243)
Index(['Zip', 'Accessories Store', 'Adult Boutique', 'African Restaurant',
       'Airport', 'Airport Service', 'Airport Terminal', 'American Restaurant',
       'Art Gallery', 'Art Museum',
       ...
       'Vegetarian / Vegan Restaurant', 'Video Game Store', 'Video Store',
       'Vietnamese Restaurant', 'Whisky Bar', 'Wine Bar', 'Wine Shop',
       'Women's Store', 'Yoga Studio', 'Zoo Exhibit'],
      dtype='object', length=243)
```

Out[8]:

	Zip	Accessories Store	Adult Boutique	African Restaurant	Airport	Airport Service	Airport Terminal	American Restaurant	Art Gallery	Art Museum	...	Ve Re
0	19173	0	0	0	0	0	0	0	0	0	...	
1	19173	0	0	0	0	0	0	0	0	0	...	
2	19173	0	0	0	0	0	0	0	0	0	...	
3	19173	0	0	0	0	0	0	0	0	0	...	
4	19173	0	0	0	0	0	0	0	0	0	...	

5 rows × 243 columns

In [9]:

```
phl_onehot.loc[phl_onehot['Playground']==1,  ['Playground']] = 2
```

In [10]:

```
phl_grouped = phl_onehot.groupby('Zip').mean().reset_index()
print(phl_grouped.shape)
phl_grouped.head()
```

```
(84, 243)
```

Out[10]:

	Zip	Accessories Store	Adult Boutique	African Restaurant	Airport	Airport Service	Airport Terminal	American Restaurant	Art Gallery	Art Museum	...	Ve Re
0	19019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1	19092	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2	19093	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
3	19099	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
4	19101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

5 rows × 243 columns

In [11]:

```
import numpy as np
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 = 10

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

# create columns according to number of top venues
columns = ['Zip']
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['Zip'] = phl_grouped['Zip']

for ind in np.arange(phl_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(phl_group
ed.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

Out[11]:

	Zip	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	8th Most Common Venue	9th Most Common Venue	10th Cor V
0	19019	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Light Rail Station	Deli / Bodega	Donut Shop	Drugstore	E Eur Resta
1	19092	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Light Rail Station	Deli / Bodega	Donut Shop	Drugstore	E Eur Resta
2	19093	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Light Rail Station	Deli / Bodega	Donut Shop	Drugstore	E Eur Resta
3	19099	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Light Rail Station	Deli / Bodega	Donut Shop	Drugstore	E Eur Resta
4	19101	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Light Rail Station	Deli / Bodega	Donut Shop	Drugstore	E Eur Resta

3. Cluster Neighborhoods in Philadelphia to 5 clusters and append other fields

In [12]:

```python
from sklearn.cluster import KMeans
# set number of clusters
kclusters = 5
phl_grouped_clustering = phl_grouped.drop('Zip', 1)

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

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

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

phl_merged = csv

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighb
orhood
phl_merged = phl_merged.merge(neighborhoods_venues_sorted, how='inner',\
                              left_on='Zip',right_on='Zip')

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

```
[3 3 3 3 3 1 1 1 3 1]
```

Out[12]:

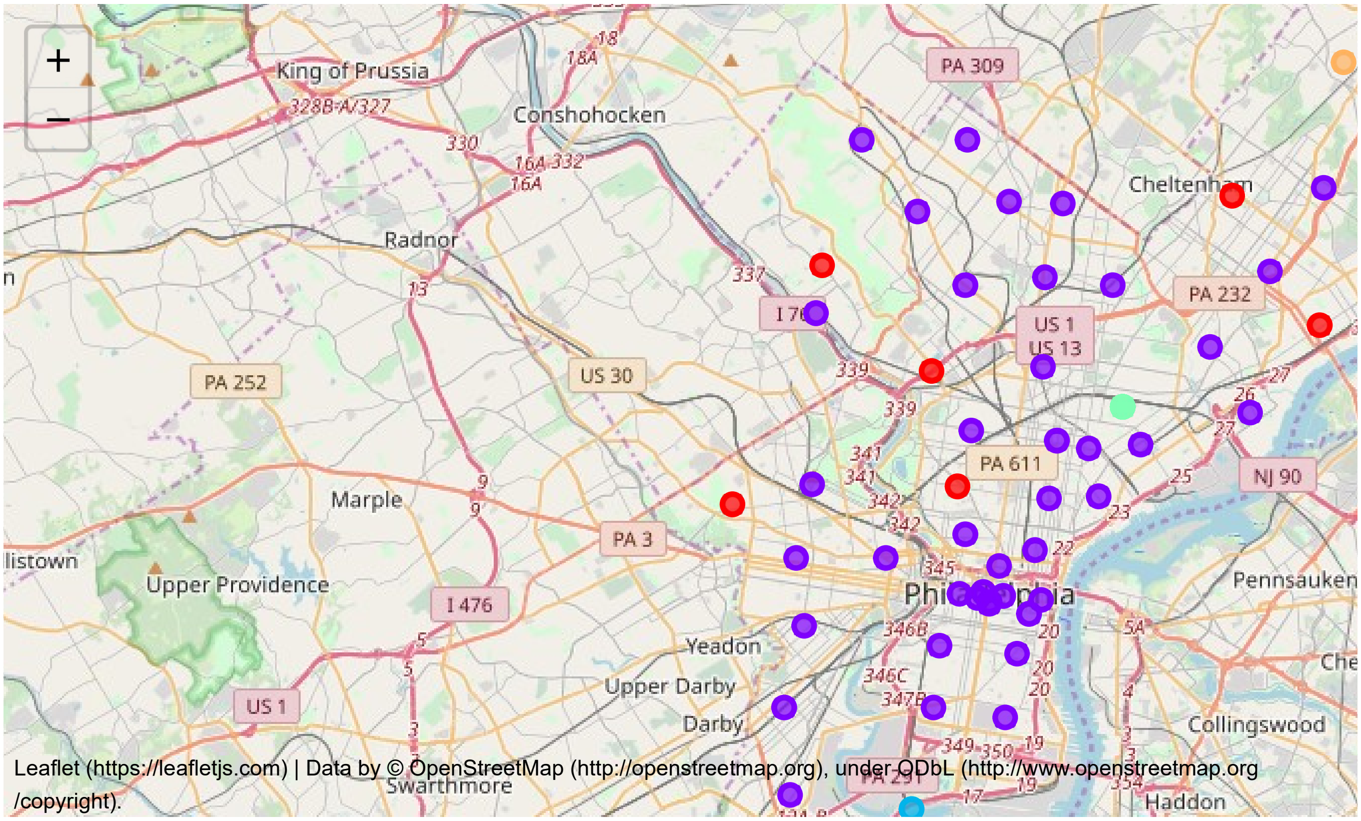
	Zip	City	State	Latitude	Longitude	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	M Comm Ve
0	19173	Philadelphia	PA	40.001811	-75.117870	-5	1	40.001811,-75.11787	3	Spanish Restaurant	T
1	19134	Philadelphia	PA	39.991712	-75.111160	-5	1	39.991712,-75.11116	1	Discount Store	D S
2	19115	Philadelphia	PA	40.092610	-75.041180	-5	1	40.09261,-75.04118	4	Pool	Ba
3	19192	Philadelphia	PA	39.951112	-75.167622	-5	1	39.951112,-75.167622	1	Salad Place	Y St
4	19155	Philadelphia	PA	40.001811	-75.117870	-5	1	40.001811,-75.11787	3	Spanish Restaurant	T

let's map out the clusters visually

In [13]:

```
import matplotlib.cm as cm
import matplotlib.colors as colors

# create map
map_clusters = folium.Map(location=[39.9526, -75.1652], 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(phl_merged['Latitude'],
                                  phl_merged['Longitude'],
                                  phl_merged['Zip'],
                                  phl_merged['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_clusters)

map_clusters
```

Out[13]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

4. Let's check each cluster and draw our recommendation for the young parents with small children

Cluster #0

In [14]:
```
phl_merged.loc[phl_merged['Cluster Labels'] == 0, \
                          phl_merged.columns[[0] + list(range(5, phl_merged.shape[1]))]]
```

Out[14]:

	Zip	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
22	19111	-5	1	40.057661,-75.08018	0	Playground	Baseball Field	Deli / Bodega	Bar	Sandwich Place
32	19128	-5	1	40.038944,-75.22104	0	Playground	Diner	Track	Coffee Shop	Tennis Stadium
41	19151	-5	1	39.975929,-75.25256	0	Cosmetics Shop	Playground	Pharmacy	Park	Deli / Bodega
65	19121	-5	1	39.981062,-75.1745	0	Deli / Bodega	Art Gallery	Playground	Food	Intersection
70	19135	-5	1	40.023611,-75.04966	0	Playground	Convenience Store	Intersection	Pizza Place	Pharmacy
83	19129	-5	1	40.011562,-75.1839	0	Playground	Gym / Fitness Center	Pub	Café	Massage Studio

Cluster #1

In [15]:

```
phl_merged.loc[phl_merged['Cluster Labels'] == 1, \
                     phl_merged.columns[[0] + list(range(5, phl_merged.shape[1]))]]
```

Out[15]:

	Zip	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5tl Co
1	19134	-5	1	39.991712,-75.11116	1	Discount Store	Donut Shop	Rental Service	Pizza Place	Mot
3	19192	-5	1	39.951112,-75.167622	1	Salad Place	Yoga Studio	Cosmetics Shop	New American Restaurant	An Res
7	19136	-5	1	40.041111,-75.02644	1	Discount Store	Zoo Exhibit	Pharmacy	Speakeasy	Sa
8	19109	-5	1	39.949612,-75.163722	1	Pizza Place	Vegetarian / Vegan Restaurant	Italian Restaurant	American Restaurant	C Res
9	19175	-5	1	39.990562,-75.12957	1	Cosmetics Shop	Food	Vietnamese Restaurant	Intersection	Res
11	19107	-5	1	39.952112,-75.15853	1	Bakery	Sandwich Place	Chinese Restaurant	Snack Place	
12	19147	-5	1	39.936562,-75.15409	1	Mexican Restaurant	Italian Restaurant	Pizza Place	Coffee Shop	Vietn Res
14	19154	-5	1	40.09146,-74.97719	1	Convenience Store	Pizza Place	Pharmacy	Deli / Bodega	Do
16	19127	-5	1	40.026626,-75.22311	1	Bar	New American Restaurant	Chinese Restaurant	Food Truck	
18	19124	-5	1	40.017362,-75.08769	1	Bus Station	Gym	Pawn Shop	Clothing Store	Donu
19	19132	-5	1	39.995412,-75.16977	1	Grocery Store	Food	Burger Joint	Fast Food Restaurant	
20	19148	-5	1	39.919812,-75.15803	1	Pizza Place	Mexican Restaurant	Grocery Store	Liquor Store	S Res
23	19119	-5	1	40.053511,-75.18858	1	Playground	Pharmacy	Pizza Place	Train Station	Ren Lo
24	19102	-5	1	39.952962,-75.16558	1	Coffee Shop	Hotel	American Restaurant	Seafood Restaurant	Cos
28	19152	-5	1	40.059611,-75.04837	1	Pizza Place	Pharmacy	Bar	Playground	
29	19108	-5	1	39.959662,-75.1605	1	Pizza Place	Art Gallery	Gym / Fitness Center	Pub	C Res
30	19143	-5	1	39.944162,-75.22718	1	Bakery	American Restaurant	Discount Store	Southern / Soul Food Restaurant	C Res
31	19144	-5	1	40.034111,-75.17203	1	Coffee Shop	American Restaurant	Southern / Soul Food Restaurant	Mobile Phone Shop	C
33	19103	-5	1	39.952162,-75.17406	1	Coffee Shop	American Restaurant	Deli / Bodega	Sushi Restaurant	
34	19141	-5	1	40.035778,-75.1447	1	Intersection	Convenience Store	Donut Shop	Chinese Restaurant	Food
35	19122	-5	1	39.977662,-75.14336	1	Men's Store	American Restaurant	Beer Garden	Restaurant	Athl
37	19114	-5	1	40.064257,-75.00155	1	Golf Course	Sporting Goods Shop	Gym / Fitness Center	Donut Shop	Di

Cluster #2

In [16]:
```
phl_merged.loc[phl_merged['Cluster Labels'] == 2, \
                              phl_merged.columns[[0] + list(range(5, phl_merged.shape[1]))]]
```

Out[16]:

	Zip	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th M Com Ve
25	19112	-5	1	39.895677,-75.19044	2	Food Truck	Zoo Exhibit	Fish Market	Field	Fast Food Restaurant	Far Ma

Cluster #3

In [17]:

```
phl_merged.loc[phl_merged['Cluster Labels'] == 3, \
                     phl_merged.columns[[0] + list(range(5, phl_merged.shape[1]))]]
```

Out[17]:

	Zip	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Com V
0	19173	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
4	19155	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
5	19183	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
6	19185	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
10	19177	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
13	19171	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
15	19196	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
17	19191	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
21	19182	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
26	19184	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
27	19187	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
36	19194	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
43	19181	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
46	19019	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
47	19188	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
48	19178	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
49	19160	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S
51	19193	-5	1	40.001811,-75.11787	3	Spanish Restaurant	Train	Pizza Place	Recreation Center	Food	Ligh S

Cluster #4

```
In [18]: phl_merged.loc[phl_merged['Cluster Labels'] == 4, \
                            phl_merged.columns[[0] + list(range(5, phl_merged.shape[1]))]]
```

Out[18]:

	Zip	Timezone	Daylight savings time flag	geopoint	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Co
2	19115	-5	1	40.09261,-75.04118	4	Pool	Bakery	Italian Restaurant	Construction & Landscaping	Field	Fas Rest

Methodology description and Discussions

After plotting the Foursquare data, we noticed that there are only limited number of playgrounds available and no "daycare" business type in Philadelphia. For our current problem, we would like the Playgrouds to have more weights than other business types so our ranking and clustering could pick those smaller number up. To achieve that, we manually increased the weight of Playgrounds in the data from 1 to 2.

The results from the clustering analysis later have shown the effect. We tried different number of clusters as well and a total of 5 clusters are eventually chosen as the results make more sense to us.

Conclusion (Recommendations)

Our recommendation would be the Cluster 0 zip codes.

In cluster 0, there are a bunch of places (zip codes) with playgrounds as the most common venue available that we can recomend to our users - Yound parents with small children.

Also to make the recommendation more general, the list of zip codes can be visually identifield on the map. Cluster 0 places are marked in Red circles. The map has shown that if you prefer to rent / buy real estate that have playgrouds nearby, the best choices are the Northwest and Northeast of Philadelphia (as marked in RED on the map). If you are a native to Philly , please feel free to let me know your thoughts on this recommendation!

```
In [ ]:
```