

```
In [204]: #load necessary libraries
import numpy as np
import pandas as pd
from geopy.geocoders import Nominatim
import requests
import matplotlib
from sklearn.cluster import KMeans
import folium
from bs4 import BeautifulSoup
import json
from pandas.io.json import json_normalize
import os
import sys
print('Donzos')
```

Donzos

```
In [208]: #create a var to store root_dir to load or write files
Root_dir = os.getcwd()

print(Root_dir)
```

/Users/jeremymcdonald/Desktop/OSP/Untitled Folder

```
In [14]: #load addresses for identified locations
fp= 'address_geo.csv'
data=pd.read_csv(fp)

dc_n='Neighborhood_Labels.csv'
dc_data=pd.read_csv(dc_n)
print ('locked and loaded')
```

locked and loaded

```
In [15]: dc_data.head()
```

Out[15]:

	Longitude	Latitude	OBJECTID	GIS_ID	NAME	WEB_URL	LABEL
0	-76.980348	38.855658	1	nhood_050	Fort Stanton	http://NeighborhoodAction.dc.gov	Fort
1	-76.997950	38.841077	2	nhood_031	Congress Heights	http://NeighborhoodAction.dc.gov	C
2	-76.995636	38.830237	3	nhood_123	Washington Highlands	http://NeighborhoodAction.dc.gov	Wa: H
3	-77.009271	38.826952	4	nhood_008	Bellevue	http://NeighborhoodAction.dc.gov	
4	-76.967660	38.853688	5	nhood_073	Knox Hill/Buena Vista	http://NeighborhoodAction.dc.gov	Hi

In [16]:

```
#setting DC Lat and Long-- got from Google Search
dcLat=38.5342
dcLong=-77.0211

print('center of dc')
```

center of dc

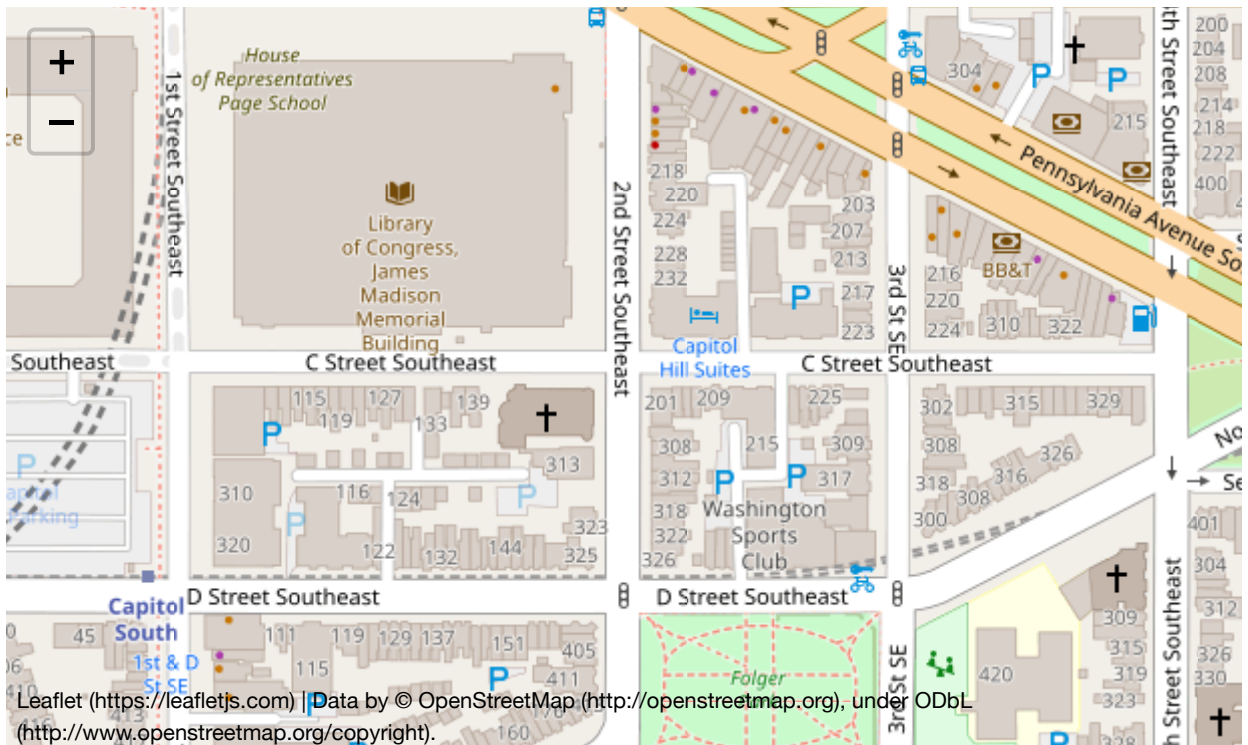
In [19]:

```
#creating the map
map_dc = folium.Map(location=[dcLat, dcLong], zoom_start=10)

#mapping neighborhoods
for lat, lng, label in zip(dc_data['Latitude'], dc_data['Longitude'], dc_data['Neighborhood']):
    popup = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=popup,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_dc)

map_dc
```

Out[19]:



```
In [21]: CLIENT_ID = 'HJOS2S0NEUORGCKMFNVKGB5IUX4TXNAI15POUFGTVIV1PFQ' # your FourS
CLIENT_SECRET = 'H5G2NCHJ5OKXS5OJ3J4XT1DSS0WMDR0YKHFSJPXAVCFT205' # your F
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT SECRET:' + CLIENT_SECRET)
```

Your credentials:

```
CLIENT_ID: HJOS2S0NEUORGCKMFNVKGCB5IUX4TXNAI15POUFGTVIV1PFQ
CLIENT_SECRET: H5G2NCHJ5OKKXS5OJ3J4XT1DSS0WMDR0YKHFSJPXAVCFT205
```

```
In [23]: dc.data.loc[0, 'NAME']
```

```
Out[23]: 'Fort Stanton'
```

```
In [27]: dclat = dc_data.loc[0, 'Latitude']
dclong = dc_data.loc[0, 'Longitude']

neighborhood_name = dc_data.loc[0, 'NAME']

print('Lat and Longs values of {} are {}, {}'.format(neighborhood_name, dclat, dclong))
```

Lat and Longs values of Fort Stanton are 38.85565773, -76.98034771.

```
In [28]: radius=500
Limit=100
print('Donzos')
```

Donzos

```
In [30]: url='https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v=20190601'
          CLIENT_ID,
          CLIENT_SECRET,
          VERSION,
          dclat,
          dclong,
          radius,
          Limit)
          url
```

```
Out[30]: 'https://api.foursquare.com/v2/venues/explore?&client_id=HJOS2S0NEUORGCKM
FNVKGB5IUX4TXNAI15POUFGTVIV1PFQ&client_secret=H5G2NCHJ5OKKXS5OJ3J4XT1DSS
0WMDR0YKHFSJXPXAVCFT205&v=20180604&ll=38.85565773,-76.98034771&radius=500&
limit=100'
```

In [48]:

```
results = requests.get(url).json()
results
{'lat': 38.853278,
 'lng': -76.983289,
 'labeledLatLngs': [{'label': 'display',
 'lat': 38.853278,
 'lng': -76.983289}],
 'distance': 367,
 'postalCode': '20020',
 'cc': 'US',
 'neighborhood': 'Barry Farm',
 'city': 'Washington',
 'state': 'D.C.',
 'country': 'United States',
 'formattedAddress': ['Stanton Road SE at Suitland Parkway SE (Suitland Parkway SE at Stanton Road SE)',
 'Washington, D.C. 20020',
 'United States']],
 'categories': [{'id': '52f2ab2ebcbc57f1066b8b4c',
 'name': 'Intersection',
 'pluralName': 'Intersections',
 'shortName': 'Intersection'}
```

In [84]:

```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

```
In [85]: venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',
nearby_venues = nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type,

# clean columns
nearby_venues.columns = [col.split(".")[1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[85]:

	name	categories	lat	lng
0	Anacostia Community Museum	Museum	38.856728	-76.976899
1	Fort Stanton Recreation Center	Recreation Center	38.857118	-76.978816
2	Fort Stanton Park	Park	38.857541	-76.978266
3	Stanton Road SE & Suitland Parkway SE	Intersection	38.853278	-76.983289

```
In [86]: print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0])

4 venues were returned by Foursquare.
```

```
In [87]: nearby_venues.columns
```

```
Out[87]: Index(['name', 'categories', 'lat', 'lng'], dtype='object')
```

In [89]:

```
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={}&client_secret={}&version={}&lat={}&lng={}&radius={}&limit={}'
        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 [90]:

```
dc_venues = getNearbyVenues(names=dc_data['NAME'],  
                             latitudes=dc_data['Latitude'],  
                             longitudes=dc_data['Longitude']  
                             )
```

Monumental Core
Georgetown
Burleith/Hillandale
Southwest Employment Area
Southwest/Waterfront
Buzzard Point
Fort McNair
Chevy Chase
Friendship Heights
North Cleveland Park
Tenleytown
American University Park
Spring Valley
Cathedral Heights
McLean Gardens
Cleveland Park
Woodley Park
Glover Park
Wesley Heights
Foxhall Crescents

```
In [91]: print(dc_venues.shape)
dc_venues.head(20)
```

```
(2756, 7)
```

```
Out[91]:
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Fort Stanton	38.855658	-76.980348	Anacostia Community Museum	38.856728	-76.976899	Museum
1	Fort Stanton	38.855658	-76.980348	Fort Stanton Recreation Center	38.857118	-76.978816	Recreation Center
2	Fort Stanton	38.855658	-76.980348	Fort Stanton Park	38.857541	-76.978266	Park
3	Fort Stanton	38.855658	-76.980348	Stanton Road SE & Suitland Parkway SE	38.853278	-76.983289	Intersection
4	Congress Heights	38.841077	-76.997950	MLK Deli	38.843560	-76.999808	Deli / Bodega
5	Congress Heights	38.841077	-76.997950	Malcolm X and MLK	38.843144	-76.998744	Road
6	Congress Heights	38.841077	-76.997950	Alabama and MLK Ave SE	38.843196	-76.998774	Intersection
7	Congress Heights	38.841077	-76.997950	Popeyes Louisiana Kitchen	38.844926	-76.997012	Fried Chicken Joint
8	Congress Heights	38.841077	-76.997950	Rita's Italian Ice & Frozen Custard	38.843554	-76.999646	Ice Cream Shop
9	Congress Heights	38.841077	-76.997950	Alberta's Kitchen Domain	38.839236	-76.994707	American Restaurant
10	Congress Heights	38.841077	-76.997950	Capital City Creamery	38.844296	-76.998362	Ice Cream Shop
11	Congress Heights	38.841077	-76.997950	Simon elementary Baseball field	38.837850	-76.997396	Baseball Field
12	Congress Heights	38.841077	-76.997950	Holiday Liquors	38.839218	-76.993997	Liquor Store
13	Congress Heights	38.841077	-76.997950	Holiday Market	38.839218	-76.993997	Convenience Store
14	Congress Heights	38.841077	-76.997950	Southeast Tennis Center	38.838545	-76.994595	Tennis Court
15	Congress Heights	38.841077	-76.997950	Mart Liquor	38.845341	-76.997656	Liquor Store
16	Congress Heights	38.841077	-76.997950	Mart Liquors Mjc	38.845341	-76.997656	Liquor Store

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
17	Washington Highlands	38.830237	-76.995636	Chesapeake Big Market	38.829696	-76.998270	Grocery Store
18	Washington Highlands	38.830237	-76.995636	Ferebee Hope Recreation Center	38.832526	-76.995676	Basketball Court
19	Washington Highlands	38.830237	-76.995636	Lucky	38.828160	-76.992383	Food

```
In [92]: dc_venues.groupby('Neighborhood').count()
```

Out[92]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
16th Street Heights	15	15	15	15	15	15
Adams Morgan	75	75	75	75	75	75
American University Park	1	1	1	1	1	1
Arboretum	15	15	15	15	15	15
Barnaby Woods	3	3	3	3	3	3
...
West End	55	55	55	55	55	55
Woodland	6	6	6	6	6	6
Woodland-Normanstone	4	4	4	4	4	4
Woodley Park	18	18	18	18	18	18
Woodridge	4	4	4	4	4	4

128 rows × 6 columns

```
In [93]: print('There are {} unique categories.'.format(len(dc_venues['Venue Category'])))
```

There are 310 unique categories.

```
In [94]: #onehot encoding--need to learn more about this
ddot_onehot=pd.get_dummies(dc_venues[['Venue Category']], prefix='', prefix

#add neigh column back to df
ddot_onehot['Neighborhood']=dc_venues['Neighborhood']
#moved neigh to the 1st
#fixed_columns=[tdot_onehot.columns[-1]] + list(ddot_onehot.columns[:-1])
#ddot_onehot=ddot_onehot[fixed_columns]

ddot_onehot.head()
```

Out[94]:

ort ge	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	...	Weight Loss Center	Whisky Bar	Wine Bar	V
0	0	0	0	0	0	0	0	...	0	0	0	
0	0	0	0	0	0	0	0	...	0	0	0	
0	0	0	0	0	0	0	0	...	0	0	0	
0	0	0	0	0	0	0	0	...	0	0	0	
0	0	0	0	0	0	0	0	...	0	0	0	

```
In [95]: for col in ddot_grouped.columns:
          print(col)
```

Tiki Bar
 Tourist Information Center
 Toy / Game Store
 Track
 Trail
 Train Station
 Tunnel
 Turkish Restaurant
 Vegetarian / Vegan Restaurant
 Video Store
 Vietnamese Restaurant
 Volleyball Court
 Warehouse Store
 Weight Loss Center
 Whisky Bar
 Wine Bar
 Wine Shop
 Winery
 Wings Joint
 Women's Store

```
In [96]: ddot_grouped = ddot_onehot.groupby('Neighborhood').sum().reset_index()  
ddot_grouped
```

Out[96]:

	Neighborhood	Afghan Restaurant	African Restaurant	Airport Lounge	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Gall
0	16th Street Heights	0	0	0	0	0	0	0	
1	Adams Morgan	1	0	0	0	0	0	0	
2	American University Park	0	0	0	0	0	0	0	
3	Arboretum	0	0	0	0	0	0	0	
4	Barnaby Woods	0	0	0	0	0	0	0	
...
123	West End	0	0	0	5	0	0	0	
124	Woodland	0	0	0	0	0	0	0	
125	Woodland-Normanstone	0	0	0	0	0	0	0	
126	Woodley Park	0	0	0	0	0	0	0	
127	Woodridge	0	0	0	0	0	0	0	

128 rows × 310 columns

```
In [79]:
```

Neighborhood
Afghan Restaurant
African Restaurant
Airport Lounge
American Restaurant
Antique Shop
Arcade
Arepa Restaurant
Art Gallery
Art Museum
Arts & Crafts Store
Asian Restaurant
Athletics & Sports
Automotive Shop
BBQ Joint
Bagel Shop
Bakery
Bank
Bar
Basketball Field

```
In [105]: dfctest= ddot_grouped([ 'Neighborhood',
                                   'Afghan Restaurant',
                                   'African Restaurant',
                                   'American Restaurant' ])
dfctest['Resturant Total']=dfctest.sum(axis=1)
dfctest.head()
```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

"""

Out[105]:

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Resturant Total
0	16th Street Heights	0	0	0	0
1	Adams Morgan	1	0	0	1
2	American University Park	0	0	0	0
3	Arboretum	0	0	0	0
4	Barnaby Woods	0	0	0	0

```
In [107]: GrFood = ddot_grouped(['Neighborhood',
    'Afghan Restaurant',
    'African Restaurant',
    'Airport Lounge',
    'American Restaurant',
    'Arepa Restaurant',
    'Asian Restaurant',
    'BBQ Joint',
    'Bagel Shop',
    'Bakery',
    'Bar',
    'Bed & Breakfast',
    'Beer Bar',
    'Beer Garden',
    'Beer Store',
    'Belgian Restaurant',
    'Bistro',
    'Brazilian Restaurant',
    'Breakfast Spot',
    'Brewery',
    'Bubble Tea Shop',
    'Burger Joint',
    'Burrito Place',
    'Cafeteria',
    'Café',
    'Cajun / Creole Restaurant',
    'Cantonese Restaurant',
    'Caribbean Restaurant',
    'Cheese Shop',
    'Chinese Restaurant',
    'Cocktail Bar',
    'Coffee Shop',
    'Comfort Food Restaurant',
    'Cuban Restaurant',
    'Cupcake Shop',
    'Deli / Bodega',
    'Dessert Shop',
    'Diner',
    'Dive Bar',
    'Donut Shop',
    'Dumpling Restaurant',
    'Eastern European Restaurant',
    'Empanada Restaurant',
    'Ethiopian Restaurant',
    'Falafel Restaurant',
    'Fast Food Restaurant',
    'Filipino Restaurant',
    'Fish & Chips Shop',
    'Food',
    'Food & Drink Shop',
    'Food Court',
    'Food Service',
    'Food Stand',
    'Food Truck',
    'French Restaurant',
    'Fried Chicken Joint',
```

```
'Frozen Yogurt Shop',  
'Gastropub',  
'Gay Bar',  
'German Restaurant',  
'Gluten-free Restaurant',  
'Greek Restaurant',  
'Hookah Bar',  
'Hot Dog Joint',  
'Hotel Bar',  
'Ice Cream Shop',  
'Indian Restaurant',  
'Irish Pub',  
'Israeli Restaurant',  
'Italian Restaurant',  
'Japanese Restaurant',  
'Juice Bar',  
'Karaoke Bar',  
'Korean Restaurant',  
'Latin American Restaurant',  
'Lebanese Restaurant',  
'Lounge',  
'Market',  
'Mediterranean Restaurant',  
'Mexican Restaurant',  
'Middle Eastern Restaurant',  
'New American Restaurant',  
'Noodle House',  
'Peruvian Restaurant',  
'Pizza Place',  
'Poke Place',  
'Portuguese Restaurant',  
'Pub',  
'Ramen Restaurant',  
'Restaurant',  
'Roof Deck',  
'Russian Restaurant',  
'Salad Place',  
'Salvadoran Restaurant',  
'Sandwich Place',  
'Scandinavian Restaurant',  
'Seafood Restaurant',  
'Smoothie Shop',  
'Snack Place',  
'Soup Place',  
'South American Restaurant',  
'Southern / Soul Food Restaurant',  
'Spanish Restaurant',  
'Speakeasy',  
'Sports Bar',  
'Steakhouse',  
'Sushi Restaurant',  
'Taco Place',  
'Tapas Restaurant',  
'Tea Room',  
'Tex-Mex Restaurant',  
'Thai Restaurant',  
'Tiki Bar',
```

```

        'Turkish Restaurant',
        'Vegetarian / Vegan Restaurant',
        'Vietnamese Restaurant',
        'Whisky Bar',
        'Wine Bar',
        'Winery',
        'Wings Joint',
        'Xinjiang Restaurant']]
GrFood['Resturant Total']=GrFood.sum(axis=1)
GrFood.head()

```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:122: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[107]:

	Neighborhood	Afghan Restaurant	African Restaurant	Airport Lounge	American Restaurant	Arepa Restaurant	Asian Restaurant	BBQ Joint	Bage Sho
0	16th Street Heights	0	0	0	0	0	0	0	0
1	Adams Morgan	1	0	0	0	0	2	1	0
2	American University Park	0	0	0	0	0	0	0	0
3	Arboretum	0	0	0	0	0	0	0	0
4	Barnaby Woods	0	0	0	0	0	0	1	0

5 rows × 122 columns

```
In [108]: GrRec = ddot_grouped(['Neighborhood',  
                                'Arcade',  
                                'Art Gallery',  
                                'Art Museum',  
                                'Athletics & Sports',  
                                'Baseball Field',  
                                'Basketball Court',  
                                'Basketball Stadium',  
                                'Bike Rental / Bike Share',  
                                'Bike Shop',  
                                'Botanical Garden',  
                                'Boutique',  
                                'Bowling Green',  
                                'Comedy Club',  
                                'Concert Hall',  
                                'Cycle Studio',  
                                'Dance Studio',  
                                'Dog Run',  
                                'Event Space',  
                                'Exhibit',  
                                'Field',  
                                'Fountain',  
                                'Garden',  
                                'General Entertainment',  
                                'Golf Course',  
                                'Gym',  
                                'Gym / Fitness Center',  
                                'Gym Pool',  
                                'Gymnastics Gym',  
                                'Historic Site',  
                                'History Museum',  
                                'Indie Movie Theater',  
                                'Jazz Club',  
                                'Lake',  
                                'Martial Arts Dojo',  
                                'Massage Studio',  
                                'Monument / Landmark',  
                                'Movie Theater',  
                                'Museum',  
                                'Music Venue',  
                                'Nightclub',  
                                'Nightlife Spot',  
                                'Opera House',  
                                'Outdoor Sculpture',  
                                'Park',  
                                'Pedestrian Plaza',  
                                'Performing Arts Venue',  
                                'Pilates Studio',  
                                'Planetarium',  
                                'Playground',  
                                'Plaza',  
                                'Pool',  
                                'Public Art',  
                                'Recreation Center',  
                                'Reservoir',  
                                'River',
```



```

        'Rock Club',
        'Scenic Lookout',
        'Science Museum',
        'Sculpture Garden',
        'Skating Rink',
        'Soccer Field',
        'Soccer Stadium',
        'Spa',
        'Sporting Goods Shop',
        'Sports Club',
        'State / Provincial Park',
        'Tennis Court',
        'Theater',
        'Track',
        'Trail',
        'Volleyball Court',
        'Yoga Studio',
        'Zoo Exhibit']]
GrRec['Recreation Total']=GrRec.sum(axis=1)
GrRec.head()

```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:75: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[108]:

	Neighborhood	Arcade	Art Gallery	Art Museum	Athletics & Sports	Baseball Field	Basketball Court	Basketball Stadium	Bike Rental / Bike Share	Bi Sh
0	16th Street Heights	0	0	0	0	0	0	0	0	
1	Adams Morgan	0	1	0	0	0	0	0	0	
2	American University Park	0	0	0	0	0	0	0	0	
3	Arboretum	0	0	0	0	0	1	0	0	
4	Barnaby Woods	0	0	0	0	0	0	0	0	

5 rows x 75 columns

```
In [109]: GrServices = ddot_grouped(['Neighborhood',
    'Antique Shop',
    'Arts & Crafts Store',
    'Automotive Shop',
    'Bank',
    'Big Box Store',
    'Bookstore',
    'Business Service',
    'Camera Store',
    'Carpet Store',
    'Check Cashing Service',
    'Christmas Market',
    'Clothing Store',
    'Comic Shop',
    'Convenience Store',
    'Cosmetics Shop',
    'Credit Union',
    'Department Store',
    'Discount Store',
    'Drugstore',
    'Dry Cleaner',
    'Electronics Store',
    'Farmers Market',
    'Flea Market',
    'Flower Shop',
    'Frame Store',
    'Fruit & Vegetable Store',
    'Furniture / Home Store',
    'Gas Station',
    'Gift Shop',
    'Grocery Store',
    'Gourmet Shop',
    'Hardware Store',
    'Health & Beauty Service',
    'Herbs & Spices Store',
    'Home Service',
    'Hostel',
    'Hotel',
    'IT Services',
    'Jewelry Store',
    'Kitchen Supply Store',
    'Laundromat',
    'Laundry Service',
    'Leather Goods Store',
    'Liquor Store',
    'Mattress Store',
    'Men\'s Store',
    'Miscellaneous Shop',
    'Mobile Phone Shop',
    'Music Store',
    'Nail Salon',
    'Office',
    'Optical Shop',
    'Organic Grocery',
    'Other Repair Shop',
    'Paper / Office Supplies Store',
```

```

        'Pet Café',
        'Pet Service',
        'Pet Store',
        'Photography Lab',
        'Record Shop',
        'Rental Car Location',
        'Salon / Barbershop',
        'Shipping Store',
        'Shoe Repair',
        'Shoe Store',
        'Shop & Service',
        'Shopping Mall',
        'Shopping Plaza',
        'Smoke Shop',
        'Souvenir Shop',
        'Storage Facility',
        'Supermarket',
        'Supplement Shop',
        'Tailor Shop',
        'Thrift / Vintage Store',
        'Toy / Game Store',
        'Video Store',
        'Warehouse Store',
        'Wine Shop',
        'Women\'s Store']]
GrServices['Services Total']=GrServices.sum(axis=1)
GrServices.head()

```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:82: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[109]:

notive Shop	Bank	Big Box Store	Bookstore	Business Service	Camera Store	Carpet Store	...	Supermarket	Supplement Shop	Tailor Shop	Thrift / Vintage Store
0	0	0	0	0	0	0	...	0	0	0	0
0	0	0	0	0	0	0	...	1	0	0	0
0	0	0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	0	0	...	0	0	0	0
0	0	0	0	0	0	0	...	0	0	0	0

```
In [113]: Grcommute = ddot_grouped(['Neighborhood',
                                   'Boat or Ferry',
                                   'Bus Line',
                                   'Bus Station',
                                   'Bus Stop',
                                   'Light Rail Station',
                                   'Metro Station',
                                   'Taxi Stand',
                                   'Train Station',])
Grcommute['Commuting Total']=Grcommute.sum(axis=1)
Grcommute.head()
```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Remove the CWD from sys.path while we load stuff.

Out[113]:

	Neighborhood	Boat or Ferry	Bus Line	Bus Station	Bus Stop	Light Rail Station	Metro Station	Taxi Stand	Train Station	Commuting Total
0	16th Street Heights	0	0	0	1	0	0	0	0	1
1	Adams Morgan	0	0	0	0	0	0	0	0	0
2	American University Park	0	0	0	0	0	0	0	0	0
3	Arboretum	0	0	0	0	0	0	0	0	0
4	Barnaby Woods	0	0	0	0	0	0	0	0	0

```
In [116]: GrMedical = ddot_grouped[['Neighborhood',
                                   'Chiropractor',
                                   'Eye Doctor',
                                   'Hospital',
                                   'Pharmacy',
                                   'Weight Loss Center']]
GrMedical['Medical Total'] = GrMedical.sum(axis=1)
GrMedical.head()
```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
import sys
```

Out[116]:

	Neighborhood	Chiropractor	Eye Doctor	Hospital	Pharmacy	Weight Loss Center	Medical Total
0	16th Street Heights	0	0	0	0	0	0
1	Adams Morgan	0	0	0	0	0	0
2	American University Park	0	0	0	0	0	0
3	Arboretum	0	0	0	0	0	0
4	Barnaby Woods	0	0	0	0	0	0

```
In [117]: GrEdu = ddot_grouped(['Neighborhood',
                                'College Administrative Building',
                                'College Bookstore',
                                'College Cafeteria',
                                'College Library',
                                'College Quad',
                                'College Stadium',
                                'General College & University'])
GrEdu['Education Total'] = GrEdu.sum(axis=1)
GrEdu.head()
```

/Users/jeremymcdonald/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
if __name__ == '__main__':
```

Out[117]:

	Neighborhood	College Administrative Building	College Bookstore	College Cafeteria	College Library	College Quad	College Stadium	General College & University	Educ
0	16th Street Heights	0	0	0	0	0	0	0	
1	Adams Morgan	0	0	0	0	0	0	0	
2	American University Park	0	0	0	0	0	0	0	
3	Arboretum	0	0	0	0	0	0	0	
4	Barnaby Woods	0	0	0	0	0	0	0	

```
In [123]: dfFood=GrFood[['Neighborhood','Resturant Total']]
dfRec=GrRec[['Neighborhood','Recreation Total']]
dfServices=GrServices[['Neighborhood','Services Total']]
dfCommute=Grcommute[['Neighborhood','Commuting Total']]
dfEdu=GrEdu[['Neighborhood','Education Total']]
dfMed=GrMedical[['Neighborhood','Medical Total']]
```

```
In [137]: df_index_col=pd.merge(dfFood, dfRec, on='Neighborhood')
df_index_col
```

Out[137]:

	Neighborhood	Resturant Total	Recreation Total
0	16th Street Heights	10	3
1	Adams Morgan	55	11
2	American University Park	1	0
3	Arboretum	5	5
4	Barnaby Woods	1	2
...
123	West End	30	11
124	Woodland	1	5
125	Woodland-Normanstone	0	4
126	Woodley Park	7	5
127	Woodridge	2	0

128 rows × 3 columns

```
In [138]: df_index_1=pd.merge(df_merge_col, dfServices, on='Neighborhood')
df_index_1
```

Out[138]:

	Neighborhood	Resturant Total	Recreation Total	Services Total
0	16th Street Heights	10	3	1
1	Adams Morgan	55	11	9
2	American University Park	1	0	0
3	Arboretum	5	5	4
4	Barnaby Woods	1	2	0
...
123	West End	30	11	13
124	Woodland	1	5	0
125	Woodland-Normanstone	0	4	0
126	Woodley Park	7	5	5
127	Woodridge	2	0	2

128 rows × 4 columns

```
In [139]: df_index_2=pd.merge(df_index_1, dfCommute, on='Neighborhood')
df_index_2
```

Out[139]:

	Neighborhood	Resturant Total	Recreation Total	Services Total	Commuting Total
0	16th Street Heights	10	3	1	1
1	Adams Morgan	55	11	9	0
2	American University Park	1	0	0	0
3	Arboretum	5	5	4	0
4	Barnaby Woods	1	2	0	0
...
123	West End	30	11	13	0
124	Woodland	1	5	0	0
125	Woodland-Normanstone	0	4	0	0
126	Woodley Park	7	5	5	0
127	Woodridge	2	0	2	0

128 rows × 5 columns

```
In [140]: df_index_3=pd.merge(df_index_2, dfEdu, on='Neighborhood')
df_index_3
```

Out[140]:

	Neighborhood	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total
0	16th Street Heights	10	3	1	1	0
1	Adams Morgan	55	11	9	0	0
2	American University Park	1	0	0	0	0
3	Arboretum	5	5	4	0	0
4	Barnaby Woods	1	2	0	0	0
...
123	West End	30	11	13	0	0
124	Woodland	1	5	0	0	0
125	Woodland-Normanstone	0	4	0	0	0
126	Woodley Park	7	5	5	0	0
127	Woodridge	2	0	2	0	0

128 rows × 6 columns


```
In [142]: df_index_4=pd.merge(df_index_3, dfMed, on='Neighborhood')
```

```
df_index_4
```

```
Out[142]:
```

	Neighborhood	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total	Medical Total
0	16th Street Heights	10	3	1	1	0	0
1	Adams Morgan	55	11	9	0	0	0
2	American University Park	1	0	0	0	0	0
3	Arboretum	5	5	4	0	0	0
4	Barnaby Woods	1	2	0	0	0	0
...
123	West End	30	11	13	0	0	1
124	Woodland	1	5	0	0	0	0
125	Woodland-Normanstone	0	4	0	0	0	0
126	Woodley Park	7	5	5	0	0	0
127	Woodridge	2	0	2	0	0	0

128 rows × 7 columns

```
In [149]: #load locations to analyze and neighborhood connected to cluster
df_N= 'dfNeighborhood.csv'
Neighborhood_data=pd.read_csv(df_N)
```

```
dc_L='dfLocation.csv'
location_data=pd.read_csv(dc_L)
print ('locked and loaded')
```

locked and loaded

```
In [150]: #create a new dataframe to join the Foursquare data too.
dfMaster=df_index_4[ ['Neighborhood' ] ]

df_index_5=pd.merge(Neighborhood_data,dfMaster, on='Neighborhood')

df_index_5
```

Out[150]:

	OBJECTID	GIS_ID	Neighborhood	Cluster Name
0	1	nhood_050	Fort Stanton	Cluster 37
1	2	nhood_031	Congress Heights	Cluster 39
2	3	nhood_123	Washington Highlands	Cluster 39
3	4	nhood_008	Bellevue	Cluster 39
4	5	nhood_073	Knox Hill/Buena Vista	Cluster 36
...
123	127	nhood_082	Marshall Heights	Cluster 33
124	128	nhood_046	Fort Davis Park	Cluster 34
125	129	nhood_042	Fairfax Village	Cluster 35
126	130	nhood_064	Hillcrest	Cluster 35
127	132	nhood_089	Mount Vernon Square	Cluster 8

128 rows × 4 columns

```
In [152]: #add select locations to dfMaster
df_index_6=pd.merge(df_index_5, location_data, on='Cluster Name')
#clean up the end of the dataframe
df_index_6.drop(['Street', 'City', 'State', 'Zip', 'Latitude', 'Longitude'],
df_index_6
```

Out[152]:

	OBJECTID	GIS_ID	Neighborhood	Cluster Name	Location
0	2	nhood_031	Congress Heights	Cluster 39	Theta
1	3	nhood_123	Washington Highlands	Cluster 39	Theta
2	4	nhood_008	Bellevue	Cluster 39	Theta
3	11	nhood_021	Capitol Hill	Cluster 26	Gamma
4	65	nhood_135	Lincoln Park	Cluster 26	Gamma
5	116	nhood_079	Hill East	Cluster 26	Gamma
6	18	nhood_066	Historic Anacostia	Cluster 28	Alpha
7	20	nhood_080	Logan Circle/Shaw	Cluster 7	Epsilon
8	39	nhood_130	Penn Quarter	Cluster 8	Iota
9	40	nhood_131	Chinatown	Cluster 8	Iota
10	44	nhood_137	Downtown East	Cluster 8	Iota
11	77	nhood_139	Mount Vernon Triangle	Cluster 8	Iota
12	78	nhood_095	North Capitol Street	Cluster 8	Iota
13	79	nhood_036	Downtown	Cluster 8	Iota
14	132	nhood_089	Mount Vernon Square	Cluster 8	Iota
15	43	nhood_134	Langston	Cluster 23	Delta
16	73	nhood_069	Ivy City	Cluster 23	Delta
17	74	nhood_118	Trinidad	Cluster 23	Delta
18	75	nhood_005	Arboretum	Cluster 23	Delta
19	76	nhood_024	Carver	Cluster 23	Delta
20	48	nhood_116	Takoma	Cluster 17	Mu
21	52	nhood_081	Manor Park	Cluster 17	Mu
22	53	nhood_014	Brightwood	Cluster 17	Mu
23	50	nhood_102	Petworth	Cluster 18	Eta
24	51	nhood_015	Brightwood Park	Cluster 18	Eta
25	66	nhood_136	16th Street Heights	Cluster 18	Eta
26	61	nhood_016	Brookland	Cluster 22	Zeta
27	69	nhood_075	Langdon	Cluster 22	Zeta
28	70	nhood_013	Brentwood	Cluster 22	Zeta
29	92	nhood_054	Friendship Heights	Cluster 11	Kappa
30	94	nhood_117	Tenleytown	Cluster 11	Kappa

	OBJECTID	GIS_ID	Neighborhood	Cluster Name	Location
31	95	nhood_002	American University Park	Cluster 11	Kappa
32	113	nhood_093	Near Northeast	Cluster 25	Beta
33	114	nhood_115	Stanton Park	Cluster 25	Beta
34	115	nhood_072	Kingman Park	Cluster 25	Beta

```
In [170]: #Just looking at the data, I can see that the Commuting and Education data
#There are a bunch of metro stations and even more buslines
#DC is home to GW, GU, Catholic, American, and DC University-- plus many sa
#For the sake of simplicity-- we are going to stick with the "home teams"
#First we are going to get a list of Metro stations from WMATA's free JSON

import http.client, urllib.request, urllib.parse, urllib.error, base64

headers = {
    # Request headers
    'api_key': '38dcc021678b4ee99f3322264890a860',
}

params = urllib.parse.urlencode({
    # Request parameters
    'LineCode': '',
})

try:
    conn = http.client.HTTPSConnection('api.wmata.com')
    conn.request("GET", "/Rail.svc/json/jStations?%s" % params, "{body}", h
    response = conn.getresponse()
    json_data = response.read()
    print(json_data)
    conn.close()
except Exception as e:
    print("[Errno {0}] {1}".format(e.errno, e.strerror))
```

```
b'{"Stations":[{"Code":"A01","Name":"Metro Center","StationTogether1":"C0
1","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":nu
ll,"LineCode4":null,"Lat":38.898303,"Lon":-77.028099,"Address":{"Stree
t":"607 13th St. NW","City":"Washington","State":"DC","Zip":"20005"}},{
"Code":"A02","Name":"Farragut North","StationTogether1":"","StationTogether
2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":nul
l,"Lat":38.903192,"Lon":-77.039766,"Address":{"Street":"1001 Connecticut
Avenue NW","City":"Washington","State":"DC","Zip":"20036"}},{
"Code":"A03","Name":"Dupont Circle","StationTogether1":"","StationTogether2":"","Li
neCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":3
8.909499,"Lon":-77.04362,"Address":{"Street":"1525 20th St. NW","City":"W
ashington","State":"DC","Zip":"20036"}},{
"Code":"A04","Name":"Woodley Par
k-Zoo/Adams Morgan","StationTogether1":"","StationTogether2":"","LineCode
1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.92499
9,"Lon":-77.052648,"Address":{"Street":"2700 Connecticut Ave., NW","Cit
y":"Washington","State":"DC","Zip":"20008"}},{
"Code":"A05","Name":"Clevel
and Park","StationTogether1":"","StationTogether2":"","LineCode1":"RD","L
ineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.934703,"Lon":-7
7.058226,"Address":{"Street":"3599 Connecticut Avenue NW","City":"Washing
```

```
In [172]: json_data
```

```
Out[172]: b'{"Stations":[{"Code":"A01","Name":"Metro Center","StationTogether1":"C01","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.898303,"Lon":-77.028099,"Address":{"Street":"607 13th St. NW","City":"Washington","State":"DC","Zip":"20005"}},{ "Code":"A02","Name":"Farragut North","StationTogether1":"","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.903192,"Lon":-77.039766,"Address":{"Street":"1001 Connecticut Avenue NW","City":"Washington","State":"DC","Zip":"20036"}},{ "Code":"A03","Name":"Dupont Circle","StationTogether1":"","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.909499,"Lon":-77.04362,"Address":{"Street":"1525 20th St. NW","City":"Washington","State":"DC","Zip":"20036"}},{ "Code":"A04","Name":"Woodley Park-Zoo/Adams Morgan","StationTogether1":"","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.924999,"Lon":-77.052648,"Address":{"Street":"2700 Connecticut Ave., NW","City":"Washington","State":"DC","Zip":"20008"}},{ "Code":"A05","Name":"Cleveland Park","StationTogether1":"","StationTogether2":"","LineCode1":"RD","LineCode2":null,"LineCode3":null,"LineCode4":null,"Lat":38.934703,"Lon":-77.058226,"Address":{"Street":"3599 Connecticut Avenue NW","City":"Washington","State":"DC","Zip":"20036"}}]}
```

```
In [176]: wmata_data = json.loads(json_data)
          print(wmata_data)
```

```
{'Stations': [{'Code': 'A01', 'Name': 'Metro Center', 'StationTogether1': 'C01', 'StationTogether2': '', 'LineCode1': 'RD', 'LineCode2': None, 'LineCode3': None, 'LineCode4': None, 'Lat': 38.898303, 'Lon': -77.028099, 'Address': {'Street': '607 13th St. NW', 'City': 'Washington', 'State': 'DC', 'Zip': '20005'}}, {'Code': 'A02', 'Name': 'Farragut North', 'StationTogether1': '', 'StationTogether2': '', 'LineCode1': 'RD', 'LineCode2': None, 'LineCode3': None, 'LineCode4': None, 'Lat': 38.903192, 'Lon': -77.039766, 'Address': {'Street': '1001 Connecticut Avenue NW', 'City': 'Washington', 'State': 'DC', 'Zip': '20036'}}, {'Code': 'A03', 'Name': 'Dupont Circle', 'StationTogether1': '', 'StationTogether2': '', 'LineCode1': 'RD', 'LineCode2': None, 'LineCode3': None, 'LineCode4': None, 'Lat': 38.909499, 'Lon': -77.04362, 'Address': {'Street': '1525 20th St. NW', 'City': 'Washington', 'State': 'DC', 'Zip': '20036'}}, {'Code': 'A04', 'Name': 'Woodley Park-Zoo/Adams Morgan', 'StationTogether1': '', 'StationTogether2': '', 'LineCode1': 'RD', 'LineCode2': None, 'LineCode3': None, 'LineCode4': None, 'Lat': 38.924999, 'Lon': -77.052648, 'Address': {'Street': '2700 Connecticut Ave., NW', 'City': 'Washington', 'State': 'DC', 'Zip': '20008'}}, {'Code': 'A05', 'Name': 'Cleveland Park', 'StationTogether1': '', 'StationTogether2': '', 'LineCode1': 'RD', 'LineCode2': None, 'LineCode3': None, 'LineCode4': None, 'Lat': 38.934703, 'Lon': -77.058226, 'Address': {'Street': '3599 Connecticut Avenue NW', 'City': 'Washington', 'State': 'DC', 'Zip': '20036'}}]}
```

```
In [177]: print(json.dumps(wmata_data, indent = 4, sort_keys=True))
```

```
{
  "Stations": [
    {
      "Address": {
        "City": "Washington",
        "State": "DC",
        "Street": "607 13th St. NW",
        "Zip": "20005"
      },
      "Code": "A01",
      "Lat": 38.898303,
      "LineCode1": "RD",
      "LineCode2": null,
      "LineCode3": null,
      "LineCode4": null,
      "Lon": -77.028099,
      "Name": "Metro Center",
      "StationTogether1": "C01",
      "StationTogether2": ""
    },
    ,

```

```
In [187]: neighborhoods_data = wmata_data['Stations']
```

```
In [188]: neighborhoods_data
```

```
Out[188]: [{'Code': 'A01',
  'Name': 'Metro Center',
  'StationTogether1': 'C01',
  'StationTogether2': '',
  'LineCode1': 'RD',
  'LineCode2': None,
  'LineCode3': None,
  'LineCode4': None,
  'Lat': 38.898303,
  'Lon': -77.028099,
  'Address': {'Street': '607 13th St. NW',
    'City': 'Washington',
    'State': 'DC',
    'Zip': '20005'}},
  {'Code': 'A02',
  'Name': 'Farragut North',
  'StationTogether1': '',
  'StationTogether2': '',
  'LineCode1': 'RD',
  'LineCode2': None,
  'LineCode3': None,
  'LineCode4': None,
  'Lat': 38.904571,
  'Lon': -77.036371,
  'Address': {'Street': '14th St. NW',
    'City': 'Washington',
    'State': 'DC',
    'Zip': '20004'}}
```

```
In [198]: # define the dataframe column
column_names = ['Name', 'Lat', 'Lon', 'Code']

# instantiate the dataframe
stations = pd.DataFrame(columns=column_names)

stations
```

Out[198]:

	Name	Lat	Lon	Code
--	------	-----	-----	------

```
In [199]: for data in neighborhoods_data:
    borough = data['Name']
    neighborhood_name = data['Code']
    neighborhood_lat = data['Lat']
    neighborhood_lon = data['Lon']

    stations = stations.append({'Name': borough,
                                'Code': neighborhood_name,
                                'Lat': neighborhood_lat,
                                'Lon': neighborhood_lon}, ignore_
```

```
In [200]: stations.head()
```

Out[200]:

	Name	Lat	Lon	Code
0	Metro Center	38.898303	-77.028099	A01
1	Farragut North	38.903192	-77.039766	A02
2	Dupont Circle	38.909499	-77.043620	A03
3	Woodley Park-Zoo/Adams Morgan	38.924999	-77.052648	A04
4	Cleveland Park	38.934703	-77.058226	A05

```
In [211]: export_Metro = stations.to_csv(r'Root_dir', index = None, header = True)

print('Exported')
```

Exported

```
In [220]: stations_data=pd.read_csv('WMATA_Cluster.csv')  
  
stations_data.head()
```

Out[220]:

	Name	Lat	Lon	Code	Cluster Name	Metro Stops
0	Tenleytown-AU	38.947808	-77.079615	A07	Cluster 11	1
1	Friendship Heights	38.960744	-77.085969	A08	Cluster 11	1
2	Van Ness-UDC	38.943620	-77.063511	A06	Cluster 12	1
3	Woodley Park-Zoo/Adams Morgan	38.924999	-77.052648	A04	Cluster 15	1
4	Cleveland Park	38.934703	-77.058226	A05	Cluster 15	1


```
In [254]: stations_final=stations_data.groupby('Cluster Name').sum()  
#drop columns no longer needed  
stations_final.drop(['Lat', 'Lon'], axis=1, inplace=True)  
stations_final.head(26)
```

Out[254]:

Metro Stops	
Cluster Name	
Cluster 11	2
Cluster 12	1
Cluster 15	2
Cluster 17	1
Cluster 18	1
Cluster 19	2
Cluster 2	1
Cluster 20	1
Cluster 22	1
Cluster 25	2
Cluster 26	4
Cluster 27	1
Cluster 3	1
Cluster 30	1
Cluster 31	1
Cluster 32	1
Cluster 33	1
Cluster 37	1
Cluster 43	1
Cluster 45	2
Cluster 5	1
Cluster 6	3
Cluster 7	1
Cluster 8	8
Cluster 9	4

```
In [223]: #File showing neighborhood with counts and location.
df_index_7=pd.merge(df_index_6, df_index_4, on='Neighborhood')
#More Cleanup of dataframe
df_index_7.drop(['OBJECTID', 'GIS_ID'], axis=1, inplace=True)
df_index_7
```

Out[223]:

	Neighborhood	Cluster Name	Location	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total	Med Ti
0	Congress Heights	Cluster 39	Theta	5	2	4	0	0	
1	Washington Highlands	Cluster 39	Theta	3	1	3	0	0	
2	Bellevue	Cluster 39	Theta	2	0	2	0	0	
3	Capitol Hill	Cluster 26	Gamma	60	17	17	0	0	
4	Lincoln Park	Cluster 26	Gamma	1	2	3	0	0	
5	Hill East	Cluster 26	Gamma	1	0	0	0	0	
6	Historic Anacostia	Cluster 28	Alpha	7	4	2	0	0	
7	Logan Circle/Shaw	Cluster 7	Epsilon	32	8	4	1	0	
8	Penn Quarter	Cluster 8	Iota	64	27	9	0	0	
9	Chinatown	Cluster 8	Iota	66	21	10	0	1	
10	Downtown East	Cluster 8	Iota	20	10	5	0	0	
11	Mount Vernon Triangle	Cluster 8	Iota	31	8	4	1	1	
12	North Capitol Street	Cluster 8	Iota	12	11	3	2	0	
13	Downtown	Cluster 8	Iota	61	15	23	0	0	
14	Mount Vernon Square	Cluster 8	Iota	64	7	23	0	1	
15	Langston	Cluster 23	Delta	5	4	5	3	0	
16	Ivy City	Cluster 23	Delta	3	1	5	1	0	
17	Trinidad	Cluster 23	Delta	3	1	0	0	0	
18	Arboretum	Cluster 23	Delta	5	5	4	0	0	
19	Carver	Cluster 23	Delta	8	2	9	2	0	

	Neighborhood	Cluster Name	Location	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total	Med Tr
20	Takoma	Cluster 17	Mu	9	8	5	2	0	
21	Manor Park	Cluster 17	Mu	1	1	4	0	0	
22	Brightwood	Cluster 17	Mu	14	2	10	1	0	
23	Petworth	Cluster 18	Eta	10	1	4	0	0	
24	Brightwood Park	Cluster 18	Eta	15	2	4	1	0	
25	16th Street Heights	Cluster 18	Eta	10	3	1	1	0	
26	Brookland	Cluster 22	Zeta	0	1	0	0	0	
27	Langdon	Cluster 22	Zeta	0	3	0	0	0	
28	Brentwood	Cluster 22	Zeta	5	4	11	0	0	
29	Friendship Heights	Cluster 11	Kappa	12	5	16	0	0	
30	Tenleytown	Cluster 11	Kappa	27	8	16	1	0	
31	American University Park	Cluster 11	Kappa	1	0	0	0	0	
32	Near Northeast	Cluster 25	Beta	28	8	11	2	0	
33	Stanton Park	Cluster 25	Beta	1	3	2	0	0	
34	Kingman Park	Cluster 25	Beta	1	3	1	0	0	

```
In [224]: #combine locations into single row
df_Final=df_index_7.groupby('Location').sum()
#Dropping Commuting total because the number is off-- instead I am going to
df_Final.drop(['Commuting Total'], axis=1, inplace=True)
df_Final
```

Out[224]:

	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total	Medical Total
Location						
Alpha	7	4	2	0	0	0
Beta	30	14	14	2	0	1
Delta	24	13	23	6	0	1
Epsilon	32	8	4	1	0	0
Eta	35	6	9	2	0	1
Gamma	62	19	20	0	0	1
Iota	318	99	77	3	3	6
Kappa	40	13	32	1	0	1
Mu	24	11	19	3	0	2
Theta	10	3	9	0	0	0
Zeta	5	8	11	0	0	0

```
In [251]: #I am creating a table that has location and cluster # so I can add final t
loc=location_data

#loc.drop(['Street', 'City','State','Zip', 'Latitude', 'Longitude'], axis=1
loc
```

Out[251]:

	Location	Cluster Name
0	Alpha	Cluster 28
1	Beta	Cluster 25
2	Gamma	Cluster 26
3	Delta	Cluster 23
4	Epsilon	Cluster 7
5	Zeta	Cluster 22
6	Eta	Cluster 18
7	Theta	Cluster 39
8	Iota	Cluster 8
9	Kappa	Cluster 11
10	Mu	Cluster 17

```
In [252]: #add Metro stop data
df_Final2=pd.merge(loc, df_Final, on='Location')
#Add metro
df_Final3=pd.merge(df_Final2, stations_final, on='Cluster Name')
#clean up the end of the dataframe

df_Final3
```

Out[252]:

	Location	Cluster Name	Resturant Total	Recreation Total	Services Total	Commuting Total	Education Total	Medical Total	Metro Stops
0	Beta	Cluster 25	30	14	14	2	0	1	2
1	Gamma	Cluster 26	62	19	20	0	0	1	4
2	Epsilon	Cluster 7	32	8	4	1	0	0	1
3	Zeta	Cluster 22	5	8	11	0	0	0	1
4	Eta	Cluster 18	35	6	9	2	0	1	1
5	Iota	Cluster 8	318	99	77	3	3	6	8
6	Kappa	Cluster 11	40	13	32	1	0	1	2
7	Mu	Cluster 17	24	11	19	3	0	2	1

In []: