

```
In [1]: #Import all needed programs

from bs4 import BeautifulSoup
import pandas as pd
import urllib.request
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
import requests # library to handle requests
# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors
import json # library to handle JSON files
import numpy as np
```

```
In [2]: #find lat,long of knoxville for center of map
address = 'Knoxville, TN'
geolocator = Nominatim(user_agent="knoxville")
location = geolocator.geocode(address)
latitude1 = location.latitude
longitude1 = location.longitude
print('The geograpical coordinate of Knoxville are {}, {}'.format(latitude1, longitude1))
```

The geograpical coordinate of Knoxville are 35.9603948, -83.9210261.

```
In [5]: #read wikipedia page to find knoxville neighborhoods
url = 'https://en.wikipedia.org/wiki/Category:Neighborhoods_in_Knoxville,_Tennessee'
#url = 'https://en.wikipedia.org/wiki/Knoxville,_Tennessee'
req = urllib.request.urlopen(url)
article = req.read().decode()

with open('Neighborhoods_in_Knoxville_Tennessee', 'w') as fo:
    fo.write(article)
```

```
In [3]: #read in Knoxville neighborhoods and find lat, long for each one. Only keep those with valid lat longs found
neighborhood=[]
latitude=[]
longitude=[]
geolocator = Nominatim(user_agent="knoxville")
article = open('Neighborhoods_in_Knoxville_Tennessee').read()
soup = BeautifulSoup(article, 'html.parser')
for i in range(5,33):
    paragraphs = soup.find_all("a")[i].text

    try:
        location = geolocator.geocode(paragraphs)
        latitude.append(location.latitude)
        longitude.append(location.longitude)
        neighborhood.append(paragraphs)
#neighborhood.replace('North Hills Historic District (Knoxville)', 'North Hills Historic District Knoxville')
    except:
        "skipped"
neighborhood
```

```
Out[3]: ['Bearden, Knoxville',
        'Chilhowee Park, Knoxville',
        'Colonial Village, Knoxville',
        'Cumberland Estates',
        'Downtown Knoxville',
        'East Knoxville',
        'Forest Heights, Knoxville',
        'Fort Sanders, Knoxville',
        'Fountain City, Knoxville',
        'Fourth and Gill, Knoxville',
        'Island Home Park, Knoxville',
        'Lake Forest, Knoxville',
        'Lindbergh Forest',
        'Lonsdale, Knoxville, Tennessee',
        'Mechanicsville, Knoxville',
        'North Knoxville, Knoxville, Tennessee',
        'Norwood, Knoxville',
        'Oakwood-Lincoln Park',
        'Old City, Knoxville',
        'Old North Knoxville',
        'Parkridge, Knoxville',
        'Rocky Hill, Knoxville',
        'Sequoyah Hills, Knoxville',
        'South Knoxville',
        'University of Tennessee',
        'West Hills, Knoxville',
        'West Knoxville']
```

```
In [4]: #add Knoxville neighborhoods with lat and longs to a dataframe
df1=pd.DataFrame()
df1['Neighborhood']=neighborhood
df1['latitude']=latitude
df1['longitude']=longitude
df1
```

Out[4]:

	Neighborhood	latitude	longitude
0	Bearden, Knoxville	35.932859	-84.002686
1	Chilhowee Park, Knoxville	35.998366	-83.882996
2	Colonial Village, Knoxville	35.919528	-83.882681
3	Cumberland Estates	40.184257	-77.018033
4	Downtown Knoxville	35.963200	-83.874300
5	East Knoxville	35.960395	-83.921026
6	Forest Heights, Knoxville	35.827578	-84.235752
7	Fort Sanders, Knoxville	35.959170	-83.934895
8	Fountain City, Knoxville	36.032026	-83.937408
9	Fourth and Gill, Knoxville	35.985916	-83.921851
10	Island Home Park, Knoxville	35.961106	-83.884009
11	Lake Forest, Knoxville	35.926751	-83.867681
12	Lindbergh Forest	41.110639	-75.943811
13	Lonsdale, Knoxville, Tennessee	35.985638	-83.957130
14	Mechanicsville, Knoxville	35.969145	-83.931845
15	North Knoxville, Knoxville, Tennessee	36.002027	-83.912407
16	Norwood, Knoxville	36.004248	-83.981576
17	Oakwood-Lincoln Park	36.266533	-86.744087
18	Old City, Knoxville	35.960395	-83.921026
19	Old North Knoxville	36.002027	-83.912407
20	Parkridge, Knoxville	35.981203	-83.906144
21	Rocky Hill, Knoxville	35.902859	-84.021298
22	Sequoyah Hills, Knoxville	35.940194	-83.969397
23	South Knoxville	35.943139	-83.898794
24	University of Tennessee	35.951439	-83.930679
25	West Hills, Knoxville	35.695915	-84.020739
26	West Knoxville	35.970058	-83.955371

```
In [154]: #read wikipedia page to find chattanooga neighborhoods

url = 'https://en.wikipedia.org/wiki/Category:Neighborhoods_in_Chattanooga,_Tennessee'
req = urllib.request.urlopen(url)
article = req.read().decode()

with open('Neighborhoods_in_Chattanooga_Tennessee', 'w') as fo:
    fo.write(article)
```

```

In [5]: #read in Chattanooga neighborhoods and find lat, long for each one. Only keep those with valid lat longs found
neighborhood=[]
latitude=[]
longitude=[]
geocator = Nominatim(user_agent="chattanooga")
article = open('Neighborhoods_in_Chattanooga_Tennessee').read()
soup = BeautifulSoup(article, 'html.parser')
for i in range(6,18):
    paragraphs = soup.find_all("a")[i].text
    #for a in soup.find_all("a")[a].text
    #for a in soup.findAll('a',href=True, attrs={'class':'mw-category-group'}) :
        # neighbor=a.find('title', attrs={'class':'mw-category-group'})
    #price=a.find('div', attrs={'class':'_1vC4OE _2rQ-NK'})
    #rating=a.find('div', attrs={'class':'hGSR34 _2beYZw'})
    #products.append(name.text)
    #prices.append(price.text)
    #ratings.append(rating.text)
        try:
            location = geocator.geocode(paragraphs)
            latitude.append(location.latitude)
            longitude.append(location.longitude)
            neighborhood.append(paragraphs)
    #neighborhood.replace('North Hills Historic District (Knoxville)', 'North Hills Historic District Knoxville')
        except:
            "skipped"
neighborhood

```

```

Out[5]: ['East Brainerd, Tennessee',
'Highland Park, Chattanooga',
'Hixson, Tennessee',
'Lookout Valley, Tennessee',
'Lupton City, Chattanooga',
'Ridgeside, Tennessee',
'Tiftonia, Tennessee',
'Tyner, Tennessee']

```

```
In [6]: #add Chattanooga neighborhoods with lat and longs to a dataframe
df2=pd.DataFrame()
df2['Neighborhood']=neighborhood
df2['latitude']=latitude
df2['longitude']=longitude
df2
```

Out[6]:

	Neighborhood	latitude	longitude
0	East Brainerd, Tennessee	34.995908	-85.150230
1	Highland Park, Chattanooga	35.028685	-85.272179
2	Hixson, Tennessee	35.217807	-85.286116
3	Lookout Valley, Tennessee	35.020353	-85.366070
4	Lupton City, Chattanooga	35.105713	-85.264264
5	Ridgeside, Tennessee	35.034202	-85.246754
6	Tiftonia, Tennessee	35.049266	-85.356523
7	Tyner, Tennessee	35.062852	-85.157176

```
In [7]: #set up needed ids for Foursquare account
CLIENT_ID = 'WFYV1ZBH5RH0W0VYLIFPXWYMKZXXEV3FCAQMPTQSDUS1LBVE' # your
Foursquare ID
CLIENT_SECRET = 'V43KV1QHBR4JXX4WYTYXGP1SX4CHXOVVCTMH4W05IHFR3TAA' # y
our Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: ' + CLIENT_SECRET)
LIMIT = 200
radius = 500
```

Your credentails:

CLIENT_ID: WFYV1ZBH5RH0W0VYLIFPXWYMKZXXEV3FCAQMPTQSDUS1LBVE
 CLIENT_SECRET: V43KV1QHBR4JXX4WYTYXGP1SX4CHXOVVCTMH4W05IHFR3TAA

```

In [8]: # set up for loop to get venues nearby each neighborhood

def getNearbyVenues(names, latitudes, longitudes, CATID, 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={}&v={}&ll={},{}&radius={}&limit={}&categoryId={}'.f
ormat(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT,
            CATID)

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

        # 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 f
or item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                            'Neighborhood Latitude',
                            'Neighborhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)

```



```
In [9]: # call loop to get venues for Chattanooga with category id for healthcare
         chattanooga_healthcare_venues = getNearbyVenues(names=df2['Neighborhood'],
                                                         latitudes=df2['latitude'],
                                                         longitudes=df2['longitude'],
                                                         CATID='4bf58dd8d48988d104941735'
                                                         )
```

East Brainerd, Tennessee
Highland Park, Chattanooga
Hixson, Tennessee
Lookout Valley, Tennessee
Lupton City, Chattanooga
Ridgeside, Tennessee
Tiftonia, Tennessee
Tyner, Tennessee

```
In [10]: # call loop to get venues for Knoxville with category id for healthcare
         knoxville_healthcare_venues = getNearbyVenues(names=df1['Neighborhood'],
                                                         latitudes=df1['latitude'],
                                                         longitudes=df1['longitude'],
                                                         CATID='4bf58dd8d48988d104941735'
                                                         )
```

Bearden, Knoxville
 Chilhowee Park, Knoxville
 Colonial Village, Knoxville
 Cumberland Estates
 Downtown Knoxville
 East Knoxville
 Forest Heights, Knoxville
 Fort Sanders, Knoxville
 Fountain City, Knoxville
 Fourth and Gill, Knoxville
 Island Home Park, Knoxville
 Lake Forest, Knoxville
 Lindbergh Forest
 Lonsdale, Knoxville, Tennessee
 Mechanicsville, Knoxville
 North Knoxville, Knoxville, Tennessee
 Norwood, Knoxville
 Oakwood-Lincoln Park
 Old City, Knoxville
 Old North Knoxville
 Parkridge, Knoxville
 Rocky Hill, Knoxville
 Sequoyah Hills, Knoxville
 South Knoxville
 University of Tennessee
 West Hills, Knoxville
 West Knoxville

```

In [11]: # call loop to get venues for Chattanooga with category id for recreat
         ion facilities
         chattanooga_Outdoor_venues = getNearbyVenues(names=df2['Neighborhood']
         ,
                                     latitudes=df2['latitude'],
                                     longitudes=df2['longitude'],
                                     CATID='4d4b7105d754a06377d81259'
         )
  
```

East Brainerd, Tennessee
 Highland Park, Chattanooga
 Hixson, Tennessee
 Lookout Valley, Tennessee
 Lupton City, Chattanooga
 Ridgeside, Tennessee
 Tiftonia, Tennessee
 Tyner, Tennessee

```
In [12]: # call loop to get venues for Knoxville with category id for recreation facilities
knoxville_Outdoor_venues = getNearbyVenues(names=df1['Neighborhood'],
                                             latitudes=df1['latitude'],
                                             longitudes=df1['longitude'],
                                             CATID='4d4b7105d754a06377d81259'
                                             )
```

Bearden, Knoxville
 Chilhowee Park, Knoxville
 Colonial Village, Knoxville
 Cumberland Estates
 Downtown Knoxville
 East Knoxville
 Forest Heights, Knoxville
 Fort Sanders, Knoxville
 Fountain City, Knoxville
 Fourth and Gill, Knoxville
 Island Home Park, Knoxville
 Lake Forest, Knoxville
 Lindbergh Forest
 Lonsdale, Knoxville, Tennessee
 Mechanicsville, Knoxville
 North Knoxville, Knoxville, Tennessee
 Norwood, Knoxville
 Oakwood-Lincoln Park
 Old City, Knoxville
 Old North Knoxville
 Parkridge, Knoxville
 Rocky Hill, Knoxville
 Sequoyah Hills, Knoxville
 South Knoxville
 University of Tennessee
 West Hills, Knoxville
 West Knoxville

```
In [13]: # Group Chattanooga healthcare venues by Neighborhood and count
chattanooga_healthcare_venues.groupby('Neighborhood').count()
```

Out[13]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Highland Park, Chattanooga	2	2	2	2	2	2
Lookout Valley, Tennessee	1	1	1	1	1	1

```
In [14]: # Group Knoxville healthcare venues by Neighborhood and count
knoxville_healthcare_venues.groupby('Neighborhood').count()
```

Out[14]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Bearden, Knoxville	31	31	31	31	31	31
Chilhowee Park, Knoxville	1	1	1	1	1	1
East Knoxville	4	4	4	4	4	4
Fort Sanders, Knoxville	50	50	50	50	50	50
Fountain City, Knoxville	5	5	5	5	5	5
Fourth and Gill, Knoxville	2	2	2	2	2	2
Lake Forest, Knoxville	2	2	2	2	2	2
Mechanicsville, Knoxville	2	2	2	2	2	2
North Knoxville, Knoxville, Tennessee	1	1	1	1	1	1
Norwood, Knoxville	8	8	8	8	8	8
Old City, Knoxville	4	4	4	4	4	4
Old North Knoxville	1	1	1	1	1	1
Rocky Hill, Knoxville	4	4	4	4	4	4
South Knoxville	1	1	1	1	1	1
University of Tennessee	3	3	3	3	3	3

```
In [208]: # Group Chattanooga recreation venues by Neighborhood and count
chattanooga_Outdoor_venues.groupby('Neighborhood').count()
```

Out[208]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Highland Park, Chattanooga	1	1	1	1	1	1
Lookout Valley, Tennessee	4	4	4	4	4	4
Lupton City, Chattanooga	2	2	2	2	2	2
Ridgeside, Tennessee	1	1	1	1	1	1

```
In [15]: # Group Knoxville recreation venues by Neighborhood and count
knoxville_Outdoor_venues.groupby('Neighborhood').count()
```

Out[15]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Bearden, Knoxville	3	3	3	3	3	3
Chilhowee Park, Knoxville	5	5	5	5	5	5
Colonial Village, Knoxville	1	1	1	1	1	1
Cumberland Estates	1	1	1	1	1	1
East Knoxville	20	20	20	20	20	20
Fort Sanders, Knoxville	9	9	9	9	9	9
Fountain City, Knoxville	1	1	1	1	1	1
Fourth and Gill, Knoxville	3	3	3	3	3	3
Island Home Park, Knoxville	2	2	2	2	2	2
Lake Forest, Knoxville	5	5	5	5	5	5
Lindbergh Forest	1	1	1	1	1	1

Mechanicsville, Knoxville	5	5	5	5	5	5
North Knoxville, Knoxville, Tennessee	2	2	2	2	2	2
Norwood, Knoxville	3	3	3	3	3	3
Oakwood-Lincoln Park	1	1	1	1	1	1
Old City, Knoxville	20	20	20	20	20	20
Old North Knoxville	2	2	2	2	2	2
Parkridge, Knoxville	5	5	5	5	5	5
Rocky Hill, Knoxville	2	2	2	2	2	2
Sequoyah Hills, Knoxville	1	1	1	1	1	1
South Knoxville	1	1	1	1	1	1
University of Tennessee	7	7	7	7	7	7
West Hills, Knoxville	1	1	1	1	1	1

```
In [16]: #print the number of unique categories for Chattanooga healthcare venues
print('There are {} uniques categories.'.format(len(chattanooga_healthcare_venues['Venue Category'].unique())))
```

There are 2 uniques categories.

```
In [17]: #print the number of unique categories for Knoxville healthcare venues
print('There are {} uniques categories.'.format(len(knoxville_healthcare_venues['Venue Category'].unique())))
```

There are 13 uniques categories.

```
In [18]: #print the number of unique categories for Chattanooga recreation venues
print('There are {} uniques categories.'.format(len(chattanooga_outdoor_venues['Venue Category'].unique())))
```

There are 6 uniques categories.

```
In [19]: #print the number of unique categories for Knoxville recreation venues
print('There are {} uniques categories.'.format(len(knoxville_outdoor_venues['Venue Category'].unique())))
```

There are 27 uniques categories.

```

In [27]: #import folium to use in mapping
import folium # map rendering library
#set up array for rainbow colors
x = np.arange(5)
ys = [i + x + (i*x)**2 for i in range(5)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# map Knoxville neighborhoods with the highest healthcare and recreation counts using latitude and longitude values
map_knoxville = folium.Map(location=[latitude1, longitude1], zoom_start=10)

# add markers to map for four interesting neighborhoods
i=0
for lat, lng, neighborhood in zip(df1['latitude'], df1['longitude'], df1['Neighborhood']):
    if neighborhood in ('Bearden, Knoxville', 'Fort Sanders, Knoxville', 'East Knoxville', 'Old City, Knoxville'):
        i+=1
        label = '{}'.format(neighborhood)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color=rainbow[i],
            fill=True,
            fill_color=rainbow[i],
            fill_opacity=0.7,
            parse_html=False).add_to(map_knoxville)

map_knoxville

```

Out[27]:



In []: