1_plot_musicvenues_1124

December 11, 2023

[1]:
```python
import geopandas as gpd
import os
import pyarrow.feather
import pandas as pd
import folium
from folium import plugins
from folium.plugins import HeatMap

# set path
current_path = os.getcwd()
root_path = os.path.dirname(current_path)
raw_path = os.path.join(root_path, 'data', 'raw')
process_path = os.path.join(root_path, 'data', 'processed')
# print(raw_path)

input_path = os.path.join(raw_path,  'flat_musicvenues.feather')
output_path = os.path.join(raw_path,'musicvenues','musicvenues.shp')
```

```
/Users/rainylty/opt/anaconda3/envs/city8/lib/python3.10/site-
packages/geopandas/_compat.py:112: UserWarning: The Shapely GEOS version
(3.10.3-CAPI-1.16.1) is incompatible with the GEOS version PyGEOS was compiled
with (3.10.1-CAPI-1.16.0). Conversions between both will be slow.
  warnings.warn(
```

[2]:
```python
# read feather data with geopandas and turn it into a geodataframe
df = pd.read_feather(input_path)

# rename geometry column
df = df.rename(columns={'coordinates.latitude': 'lat'})
df = df.rename(columns={'coordinates.longitude': 'lon'})
df = df.rename(columns={'location.display_address': 'address'})

# turn object into string in price column
df['price'] = df['price'].astype(str)
# print(type(df['price'][0]))

gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.lon, df.lat))
gdf.head()
```

```
[2]:                      id                             alias  \
     0  AbAw6Iqjrhts4CFxJD6hDA            bar-margot-atlanta-2
     1  ZoFht0viJtWiAt4MeP6zvQ               kats-cafe-atlanta
     2  8CV0o1eU0aTD7nDaxPcwzw    domaine-nightclub-atlanta-2
     3  fMyqmv7MfjUR4HaA0w_5Ig       dome-in-the-city-atlanta
     4  KAMJigcGSquToNvU1hjqZQ  atlanta-symphony-hall-atlanta

                         name                                           image_url  \
     0             Bar Margot  https://s3-media2.fl.yelpcdn.com/bphoto/bT1Qdk…
     1             Kat's Cafe  https://s3-media3.fl.yelpcdn.com/bphoto/E6KWha…
     2      Domaine Nightclub  https://s3-media1.fl.yelpcdn.com/bphoto/ipdvNF…
     3       Dome In The City  https://s3-media2.fl.yelpcdn.com/bphoto/Dxt4eu…
     4  Atlanta Symphony Hall  https://s3-media3.fl.yelpcdn.com/bphoto/Bh9k4K…

        is_closed                                                url  review_count  \
     0      False  https://www.yelp.com/biz/bar-margot-atlanta-2?…           234
     1      False  https://www.yelp.com/biz/kats-cafe-atlanta?adj…           273
     2      False  https://www.yelp.com/biz/domaine-nightclub-atl…            24
     3      False  https://www.yelp.com/biz/dome-in-the-city-atla…             3
     4      False  https://www.yelp.com/biz/atlanta-symphony-hall…            13

                                              categories  rating transactions  \
     0                              Lounges, Music Venues     4.0     delivery
     1                         New American, Music Venues     4.0     delivery
     2                                       Music Venues     3.0
     3  Venues & Event Spaces, Stadiums & Arenas, Musi…     3.5
     4                                       Music Venues     4.0

        …         lon     location.address1 location.address2  \
     0  … -84.385511          75 14th St NE
     1  … -84.381030       970 Piedmont Ave
     2  … -84.384044  1150 Crescent Ave NE              Fl 1
     3  … -84.383650  1100 Peachtree St NE              None
     4  … -84.384719  1280 Peachtree St NE              None

                  location.address3  location.city  location.zip_code  \
     0  Four Seasons Hotel Atlanta        Atlanta              30309
     1                                    Atlanta              30309
     2                                    Atlanta              30309
     3                        None        Atlanta              30309
     4                        None        Atlanta              30309

       location.country location.state  \
     0               US             GA
     1               US             GA
     2               US             GA
     3               US             GA
```

```
4               US             GA

                                                 address  \
0  75 14th St NE, Four Seasons Hotel Atlanta, Atl…
1              970 Piedmont Ave, Atlanta, GA 30309
2      1150 Crescent Ave NE, Fl 1, Atlanta, GA 30309
3            1100 Peachtree St NE, Atlanta, GA 30309
4            1280 Peachtree St NE, Atlanta, GA 30309

                     geometry
0  POINT (-84.38551 33.78688)
1  POINT (-84.38103 33.78112)
2  POINT (-84.38404 33.78592)
3  POINT (-84.38365 33.78488)
4  POINT (-84.38472 33.78935)

[5 rows x 25 columns]
```

[3]:
```
gdf.describe()
gdf.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 56 entries, 0 to 55
Data columns (total 25 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 56 non-null     object
 1   alias              56 non-null     object
 2   name               56 non-null     object
 3   image_url          56 non-null     object
 4   is_closed          56 non-null     bool
 5   url                56 non-null     object
 6   review_count       56 non-null     int32
 7   categories         56 non-null     object
 8   rating             56 non-null     float64
 9   transactions       56 non-null     object
 10  price              56 non-null     object
 11  phone              56 non-null     object
 12  display_phone      56 non-null     object
 13  distance           56 non-null     float64
 14  lat                56 non-null     float64
 15  lon                56 non-null     float64
 16  location.address1  55 non-null     object
 17  location.address2  43 non-null     object
 18  location.address3  47 non-null     object
 19  location.city      56 non-null     object
 20  location.zip_code  56 non-null     object
 21  location.country   56 non-null     object
 22  location.state     56 non-null     object
 23  address            56 non-null     object
 24  geometry           56 non-null     geometry
dtypes: bool(1), float64(4), geometry(1), int32(1), object(18)
memory usage: 10.5+ KB
```

[4]:
```
# give gdf a crs, use WGS84 mercator
gdf.crs = {'init': 'epsg:4326'}
```

```
/Users/rainylty/opt/anaconda3/envs/city8/lib/python3.10/site-
packages/pyproj/crs/crs.py:141: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
  in_crs_string = _prepare_from_proj_string(in_crs_string)
```

[9]:
```
# plot the geodataframe with folium
m = folium.Map(location=[33.7868794367165, -84.3855107579268], zoom_start=11,␣
 ↪tiles='cartodb positron')
# folium.GeoJson(gdf, tooltip=folium.GeoJsonTooltip(fields=['name','price'])).
 ↪add_to(m)
# different colors for different price levels
folium.GeoJson(gdf,
                           tooltip=folium.
 ↪GeoJsonTooltip(fields=['name','price', 'rating',␣
 ↪'review_count','address']),
                           # style_function=lambda x: {'color': 'green' if␣
 ↪x['properties']['price'] == '$' else 'orange' if x['properties']['price'] ==␣
 ↪'$$' else 'red' if x['properties']['price'] == '$$$' else 'black'},
                           # different colors for different rating levels
                           style_function=lambda x: {'color': 'green' if␣
 ↪x['properties']['rating'] >= 4 else 'orange' if x['properties']['rating'] >=␣
 ↪3 else 'red' if x['properties']['rating'] >= 2 else 'black'},
                           ).add_to(m)


m
```

[9]:
```
<folium.folium.Map at 0x133d930a0>
```

[44]:
```
# save geodataframe as shapefile
gdf.to_file(output_path)
```

```
/var/folders/38/ttqg2y215g16g2ng7jd502_c0000gn/T/ipykernel_23326/1322296942.py:2
: UserWarning: Column names longer than 10 characters will be truncated when
saved to ESRI Shapefile.
  gdf.to_file(output_path)
```

```
[17]: # draw a heatmap with folium
      # make intersection of gdf and polygon
      polygon_path = os.path.join(raw_path,␣
        ↪'City_of_Atlanta_Neighborhood_Statistical_Areas/City_of_Atlanta_boundary.
        ↪geojson')
      polygon = gpd.read_file(polygon_path)
      polygon.crs = {'init': 'epsg:4326'}
      # make intersection of gdf and polygon
      gdf_intersect = gpd.overlay(gdf, polygon, how='intersection')
      # print(gdf_intersect.head())

      m_heat = folium.Map(location=[33.7868794367165, -84.3855107579268],␣
        ↪zoom_start=12, tiles='cartodb positron')
      m_heat.add_child(HeatMap(data=gdf_intersect[['lat', 'lon']], radius=20))
      folium.GeoJson(polygon).add_to(m_heat)
      # change the opacity of the heatmap

      folium.LayerControl().add_to(m_heat)

      m_heat
```

```
/Users/rainylty/opt/anaconda3/envs/city8/lib/python3.10/site-
packages/pyproj/crs/crs.py:141: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
  in_crs_string = _prepare_from_proj_string(in_crs_string)
```

```
[17]: <folium.folium.Map at 0x12eaa84c0>
```

```
[11]: # add polygon layer
      # read polygon data
      polygon_path = os.path.join(raw_path,␣
        ↪'City_of_Atlanta_Neighborhood_Statistical_Areas/
        ↪City_of_Atlanta_Neighborhood_Statistical_Areas.shp')
      polygon = gpd.read_file(polygon_path)
      polygon.head()
      # add polygon layer to m
      folium.GeoJson(polygon).add_to(m)
      m
```

```
[11]: <folium.folium.Map at 0x133d930a0>
```

```
[114]: # count the number of music venues in each neighborhood
       # join gdf and polygon
       # add a column 'count' to gdf
       gdf['count'] = 1
```

```
gdf_polygon = gpd.sjoin(gdf, polygon, how='right', op='within')
gdf_polygon.head()
gdf_polygon.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 130 entries, 0 to 101
Data columns (total 42 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   index_left         49 non-null     float64
 1   id                 49 non-null     object
 2   alias              49 non-null     object
 3   name               49 non-null     object
 4   image_url          49 non-null     object
 5   is_closed          49 non-null     object
 6   url                49 non-null     object
 7   review_count       49 non-null     float64
 8   categories         49 non-null     object
 9   rating             49 non-null     float64
 10  transactions       49 non-null     object
 11  price              49 non-null     object
 12  phone              49 non-null     object
 13  display_phone      49 non-null     object
 14  distance           49 non-null     float64
 15  lat                49 non-null     float64
 16  lon                49 non-null     float64
 17  location.address1  48 non-null     object
 18  location.address2  37 non-null     object
 19  location.address3  40 non-null     object
 20  location.city      49 non-null     object
 21  location.zip_code  49 non-null     object
 22  location.country   49 non-null     object
 23  location.state     49 non-null     object
 24  address            49 non-null     object
 25  count              49 non-null     float64
 26  OBJECTID           130 non-null    int64
 27  NPU                130 non-null    object
 28  STATISTICA         130 non-null    object
 29  POP2010            130 non-null    int64
 30  NEIGHBORHO         130 non-null    object
 31  URL                130 non-null    object
 32  A                  130 non-null    object
 33  pop                130 non-null    int64
 34  white              130 non-null    float64
 35  black              130 non-null    float64
 36  asian              130 non-null    float64
 37  other              130 non-null    float64
 38   hispanic           130 non-null    float64
 39   GlobalID           130 non-null    object
 40   last_edite         4 non-null      object
 41   geometry           130 non-null    geometry
dtypes: float64(12), geometry(1), int64(3), object(26)
memory usage: 43.7+ KB
```

```
/Users/rainylty/opt/anaconda3/envs/city8/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3318: FutureWarning: The `op`
parameter is deprecated and will be removed in a future release. Please use the
`predicate` parameter instead.
  if await self.run_code(code, result, async_=asy):
/var/folders/38/ttqg2y215g16g2ng7jd502_c0000gn/T/ipykernel_23326/2126710322.py:6
: UserWarning: CRS mismatch between the CRS of left geometries and the CRS of
right geometries.
Use `to_crs()` to reproject one of the input geometries to match the CRS of the
other.

Left CRS: +init=epsg:4326 +type=crs
Right CRS: EPSG:4326

  gdf_polygon = gpd.sjoin(gdf, polygon, how='right', op='within')
```

[95]:
```
# convert polygon to geojson
polygon.to_file(os.path.
 ↪join(raw_path,'City_of_Atlanta_Neighborhood_Statistical_Areas','City_of_Atlanta_Neighborhoo
 ↪geojson'), driver='GeoJSON')
```

2_parking_lot

December 11, 2023

[9]:
```python
import osmnx as ox
import geopandas as gpd
from shapely.geometry import MultiPoint, MultiPolygon
import folium
from folium import plugins
import os
```

[10]:
```python
# set path
current_path = os.getcwd()
root_path = os.path.dirname(current_path)
raw_path = os.path.join(root_path, 'data', 'raw')
process_path = os.path.join(root_path, 'data', 'processed')
# print(raw_path)
```

[5]:
```python
# Specify the name of the city and country
place_name = "Atlanta, USA"

# Download the point of interest data
pois = ox.features_from_place(place_name, tags={'amenity':'parking'})

# see how many features were returned
print(len(pois), 'points of interest')
```

```
1365 points of interest
```

[29]:
```python
# plot out the pois
# ax = ox.plot_footprints(pois)

# plot with folium

# plot the geodataframe with folium
m = folium.Map(location=[33.7868794367165, -84.3855107579268], zoom_start=11,␣
 ↪tiles='cartodb positron')
# add poi's 'name' column as pop-up labels for the markers
folium.features.GeoJson(pois,
                                                fill_color="red",␣
 ↪fill_opacity=0.5, stroke=False,
                                                tooltip=folium.
↪GeoJsonTooltip(fields=['parking','access','fee','capacity']),
                                                ).add_to(m)

polygon_path = os.path.join(raw_path,␣
↪'City_of_Atlanta_Neighborhood_Statistical_Areas/
↪City_of_Atlanta_Neighborhood_Statistical_Areas.shp')
polygon = gpd.read_file(polygon_path)
polygon.head()
# add polygon layer to m
folium.GeoJson(data=polygon, fill=False).add_to(m)

m
```

[29]:
```
<folium.folium.Map at 0x1507299d0>
```

[15]:
```
pois.head(10)
```

[15]:
```
                         amenity                     geometry       name  \
element_type osmid
node         496141022   parking  POINT (-84.39143 33.76207)        NaN
             496141023   parking  POINT (-84.39142 33.76115)        NaN
             497397032   parking  POINT (-84.32157 33.75499)        NaN
             534431138   parking  POINT (-84.38312 33.75679)        NaN
             567065914   parking  POINT (-84.35175 33.79235)        NaN
             600429864   parking  POINT (-84.39090 33.76088)  Interpark
             681262448   parking  POINT (-84.39408 33.75466)        NaN
             795904771   parking  POINT (-84.39821 33.79183)        NaN
             1127136673  parking  POINT (-84.38043 33.75540)        NaN
             1179861872  parking  POINT (-84.38051 33.75715)        NaN

                        old_name operator layer      parking access  fee  \
element_type osmid
node         496141022       NaN      NaN    -1  underground    NaN  NaN
             496141023       NaN      NaN    -1  underground    NaN  NaN
             497397032       NaN      NaN   NaN      surface    yes   no
             534431138       NaN      NaN   NaN          NaN    NaN  NaN
             567065914       NaN      NaN   NaN          NaN    NaN  NaN
             600429864       NaN      NaN   NaN          NaN    NaN  NaN
             681262448       NaN      NaN   NaN      surface    NaN  NaN
             795904771       NaN      NaN   NaN          NaN    NaN  NaN
             1127136673      NaN      NaN   NaN      surface    yes  yes
             1179861872      NaN      NaN   NaN          NaN    NaN  NaN

                        capacity  … phone smoothness access:conditional  \
element_type osmid                …
node         496141022       NaN  …   NaN        NaN                NaN
```

```
              496141023        NaN  …  NaN        NaN                 NaN
              497397032        NaN  …  NaN        NaN                 NaN
              534431138        100  …  NaN        NaN                 NaN
              567065914        NaN  …  NaN        NaN                 NaN
              600429864        NaN  …  NaN        NaN                 NaN
              681262448        NaN  …  NaN        NaN                 NaN
              795904771        NaN  …  NaN        NaN                 NaN
              1127136673        20  …  NaN        NaN                 NaN
              1179861872        20  …  NaN        NaN                 NaN

                         maxstay:conditional building:part email image ways  \
element_type osmid
node         496141022                   NaN           NaN   NaN   NaN  NaN
             496141023                   NaN           NaN   NaN   NaN  NaN
             497397032                   NaN           NaN   NaN   NaN  NaN
             534431138                   NaN           NaN   NaN   NaN  NaN
             567065914                   NaN           NaN   NaN   NaN  NaN
             600429864                   NaN           NaN   NaN   NaN  NaN
             681262448                   NaN           NaN   NaN   NaN  NaN
             795904771                   NaN           NaN   NaN   NaN  NaN
             1127136673                  NaN           NaN   NaN   NaN  NaN
             1179861872                  NaN           NaN   NaN   NaN  NaN

                         type roof:shape
element_type osmid
node         496141022   NaN        NaN
             496141023   NaN        NaN
             497397032   NaN        NaN
             534431138   NaN        NaN
             567065914   NaN        NaN
             600429864   NaN        NaN
             681262448   NaN        NaN
             795904771   NaN        NaN
             1127136673  NaN        NaN
             1179861872  NaN        NaN

[10 rows x 80 columns]
```

[ ]:
```
# Remove rows with empty geometries
pois = pois[pois.geometry.notnull()]

# Remove rows with invalid geometries
pois = pois[pois.geometry.is_valid]
pois.geometry = pois.geometry.apply(lambda x: x[0] if isinstance(x, MultiPoint)␣
 ↪else x)
# pois.geometry = pois.geometry.apply(lambda x: x[0] if isinstance(x,␣
 ↪MultiPolygon) else x)
```

```
# if the field is a list, drop the list and keep the first element
pois.geometry = pois.geometry.apply(lambda x: x[0] if isinstance(x, list) else␣
↪x)
# save the data as a geojson file
pois.to_file('../data/raw/parking.geojson', driver='GeoJSON')
```

[ ]:
```
# Remove rows with empty geometries
pois = pois[pois.geometry.notnull()]

# Remove rows with invalid geometries
pois = pois[pois.geometry.is_valid]
pois.geometry = pois.geometry.apply(lambda x: x[0] if isinstance(x, MultiPoint)␣
↪else x)
pois.to_file('../data/raw/parking.shp')
# Convert MultiPolygons to Polygons
# pois.geometry = pois.geometry.apply(lambda x: x[0] if isinstance(x,␣
↪MultiPolygon) else x)
```

[38]:
```
# read data in the data/final
gdf_final = gpd.read_file('../data/final/5finalists.geojson')

# add parkingn poi and gdf_final to m_final
m_final = folium.Map(location=[33.7868794367165, -84.3855107579268],␣
↪zoom_start=11, tiles='cartodb positron')
# add gdf_final to m_final
folium.features.GeoJson(gdf_final,
                                                fill_color="blue",␣
↪fill_opacity=0.3, stroke=True,
                                                tooltip=folium.
↪GeoJsonTooltip(fields=['NEIGHBORHO']),
                                                ).add_to(m_final)
# add polygon layer to m
# folium.GeoJson(data=polygon, fill=False).add_to(m_final)
# add poi's 'name' column as pop-up labels for the markers
folium.features.GeoJson(pois,
                                                fill_color="red",␣
↪fill_opacity=0.8, stroke=False,
                                                tooltip=folium.
↪GeoJsonTooltip(fields=['parking','access','fee','capacity']),
                                                ).add_to(m_final)
m_final
```

[38]:
```
<folium.folium.Map at 0x1542ba9d0>
```

3_score_calculation

December 11, 2023

```
[5]: import geopandas as gpd
     import pandas as pd
```

```
[ ]: gdf = gpd.read_file('../data/raw/transport_census.geojson')
```

```
[8]: # Perform the quantile cut on the 'monthly_cost' column
     gdf['monthly_housing_costE'] = gdf['monthly_housing_costE'].fillna(0)
     gdf['housing_cost_score'] = pd.qcut(gdf['monthly_housing_costE'], 5, labels=[1,␣
      ↪2, 3, 4, 5]).astype(int)

     gdf['hhincomeE'] = gdf['hhincomeE'].fillna(0)
     gdf['income_score'] = pd.qcut(gdf['hhincomeE'], 5, labels=[1, 2, 3, 4, 5]).
      ↪astype(int)
```

```
[11]: def assign_score(age):
          if 25 <= age < 35:
              return 4
          elif 18 <= age < 25:
              return 3
          elif 35 <= age < 44:
              return 2
          else:
              return 1


      gdf['age_score'] = gdf['median_ageE'].apply(assign_score)

      # add up the scores to get a final score
      gdf['score'] = gdf['age_score'] + gdf['income_score'] +␣
       ↪gdf['housing_cost_score']
```

```
[12]: # save back
      gdf.to_file("../data/processed/transport_census.geojson", driver='GeoJSON')
```

5_detailed_score

December 11, 2023

[3]:
```
# use pandas and seaborn to plot a stack chart
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


df = pd.read_csv('../data/final/5finalists_score.csv')
df.head()
```

[3]:
```
                NEIGHBORHO    pop  hhincomeE  owner_occupied_housingE  \
0             East Atlanta   5101     111759                     1467
1  Peachtree Heights West   4874      83243                     1279
2          Buckhead Forest   3372      83243                     1279
3                  Midtown  16218     109426                     1569
4               Inman Park   6196      78182                      412

   renter_occupied_housingE  public_transportE  monthly_housing_costE  \
0                       641                 54                   1589
1                      1792                 63                   1625
2                      1792                 63                   1625
3                      1863                276                   1914
4                       882                117                   1657

   drive_to_workE  demographic  housing cost score  demographic  age score  \
0            2929                                4                       4
1            2786                                4                       4
2            2786                                4                       4
3            2482                                5                       2
4            1237                                4                       4

   demographic income score  demographic total score  \
0                         5                       13
1                         4                       12
2                         4                       12
3                         5                       12
4                         4                       12

   transport score(service area)
```

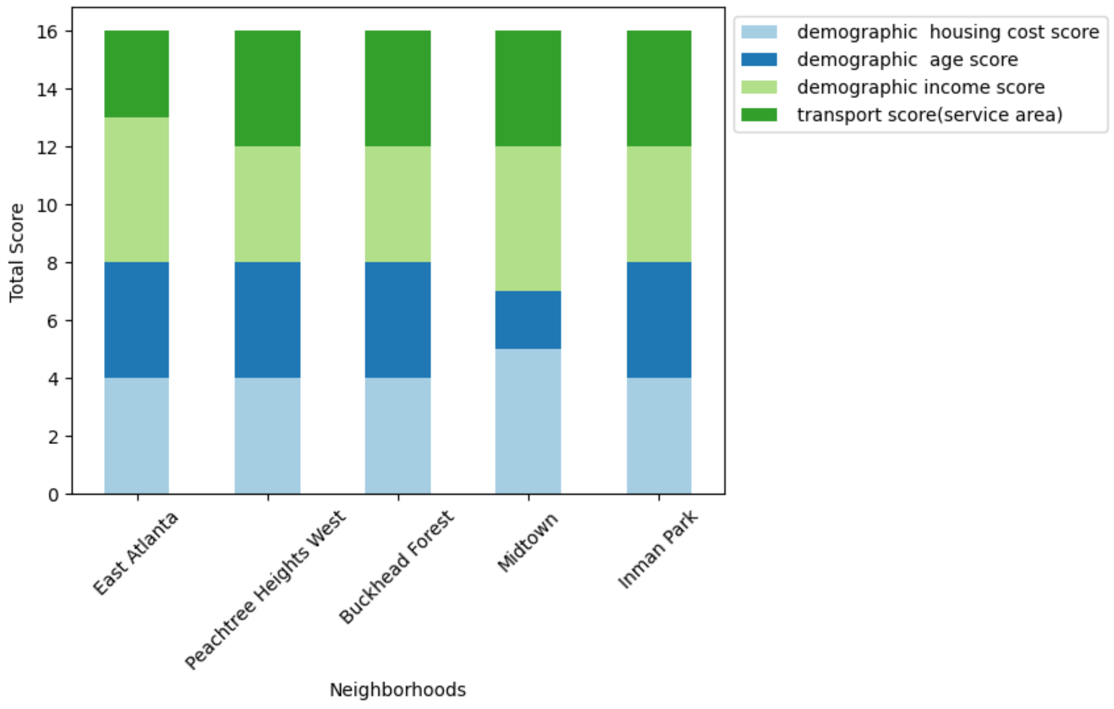
```
0                                 3
1                                 4
2                                 4
3                                 4
4                                 4
```

[ ]:
```
df = df.drop(df.columns[-2], axis=1)
```

[20]:
```
df.iloc[:, -4:].plot(kind='bar', stacked=True)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))

plt.xticks(range(len(df)), df.iloc[:, 0], rotation=45)
plt.xlabel('Neighborhoods')
plt.ylabel('Total Score')
sns.set_palette('Paired')
# plt.show()

plt.savefig('../map/plot/stacked_chart.png', dpi=300)
```


demographic housing cost score
demographic age score
demographic income score
transport score(service area)
Total Score
East Atlanta
Peachtree Heights West
Buckhead Forest
Midtown
Inman Park
Neighborhoods