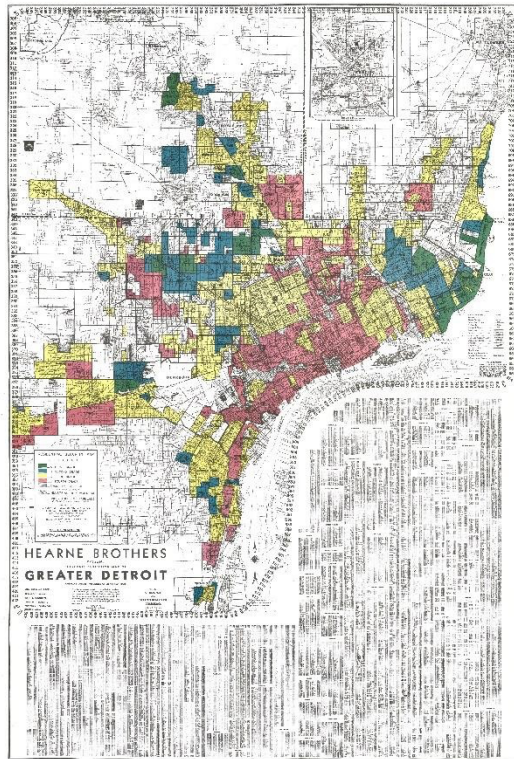


#### HW 4 – Using Text Data, JSON & APIs in the wild. Redlining



In the prior homework we replicated a simulation of residential segregation that ‘explained’ segregation through individual preference. However, this model did not account for the active policy interventions that have led to segregation in the United States. One such policy intervention is the use of ‘residential security maps’ to determine which neighborhoods would be granted government-backed low interest rate mortgages. These maps were used explicitly to exclude neighborhoods with high African American representation from receiving government investment. Those neighborhoods, were marked in red, and the practice has become known as redlining. All of the non-python readings linked in this HW are entirely optional (i.e. not required).

(<https://www.smartcitiesdive.com/ex/sustainablecitiescollective/short-history-redlining/1162160/>)

In today’s homework we will examine the original redlining map of Detroit as we practice the following technical skills:

- Working with text files
- Parsing new JSON files
- Navigating new APIs
- General python skills including list comprehensions, dictionaries etc.

**This is the first hw/project for which we are not providing starter code. Instead we will provide a series of instructions that you should follow.**

**Step 1:** Using the python methods we used in class obtain the json file located at:

<https://dsl.richmond.edu/panorama/redlining/static/downloads/geojson/MIDetroit1939.geojson> and deserialize it into a usable python object named **RedliningData**

This was previewed in the Week 6 Consolidation Lecture/Class Activity. Please refer to that video if you are having trouble

**Background** The json is digitized version of the original dataset used in 1936. The file divides Detroit into **238** districts. Each district has a set of latitude and longitude coordinates, a letter grade (A, B, C, or D), and a text description of the demographics of the neighborhoods. The district grades are associated with the following colors and designations: green for the “Best,” blue for “Still Desirable,” yellow for “Definitely Declining,” and red for “Hazardous.” These grades were determined primarily by the racial composition of the districts (<https://www.esri.com/arcgis-blog/products/arcgis-livingatlas/announcements/redlining-data-now-in-arcgis-living-atlas/>)

**Content warning** – the text descriptions are from 1936 and may contain offensive language and ideas. Reading each those text descriptions is NOT needed for this assignment.

## Step 2:

- Develop a mental map of the data structure of the redlining data. (is it a list of dictionaries? What is the structure of the values for each key in the dictionary etc)
  - This is previewed in the consolidation lecture recording from week 6
- Define a class called **DetroitDistrict**
  - This class should have the following **attributes**
    - **"Coordinates"**: a list of lists of coordinates from the json file for each district
      - Note that some districts are non-contiguous, which may effect the structure of this attribute
    - **"HolcGrade"**: a string with the letter Grade from the redlining data for each district
    - **"HolcColor"**: a string with the appropriate color for each district based on the instructions below
      - Districts with grade A should be assigned the color 'darkgreen'
      - Districts with grade B should be assigned the color 'cornflowerblue'
      - Districts with grade C should be assigned the color 'gold'
      - Districts with grade D should be assigned the color 'maroon'
    - **"name"**: a string or integer with a name of the district. This is up to you, you might use a number or other iterator, or you may extract a neighborhood name from the qualitative portion in the redlining data
    - **"Qualitative Description"**: a string with the text description called Section 8 for each district.
    - **RandomLat**: A random latitude point within the district (this should be empty for now, we will need a later step to do this)
    - **RandomLong**: A random Longitude point within the district (this is not available from the JSON and will be filled in in a later step)
    - **Median Income**: Median 2018 household Income at the random lat/long in that district (filled in later)
    - **CensusTract**: A code for the census for the Tract that contains the point defined by RandomLat & RandomLong
- Use a list comprehension to create 238 objects of the class **DetroitDistrict** from the redlining data in a list called **Districts**

- Note that these must be created in the order they are present in the redlining data

**Step 3.** One of the learning objectives of this course is to be able to use documentation to figure out how to use packages you have not been taught. This will be a critical skill in your internships and your year 2 courses. In line with this objective, you will be asked to complete the following code in order to produce a redlining map of Detroit.

The starter code provided only needs 2 lines to be filled in – see the comments in red. You must use the matplotlib documentation to figure out how to create a polygon for each district, and give it the color appropriate to that district. Keep the borders of the district as black. Matplotlib documentation can be found here:

<https://matplotlib.org/3.1.1/index.html> Please note that **polygon** is a specific method within in matplotlib

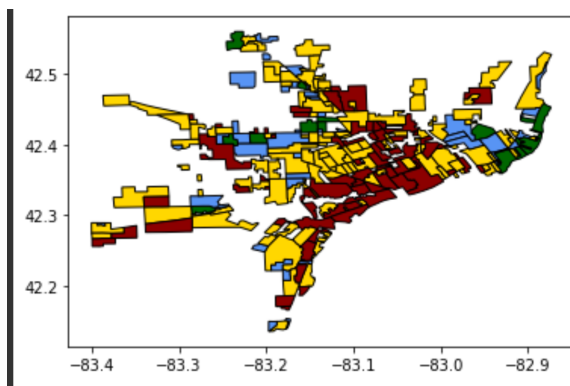
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

fig, ax = plt.subplots()

for ____: # what kind of for loop makes sense?
    ax.add_patch(.....) # add arguments here
    ax.autoscale()
    plt.rcParams["figure.figsize"] = (15,15)

plt.show()
```

The graph your produce should look like this:



#### Step 4.

The following code will pick a latitude and longitude coordinate from each of the districts

```
import random as random
from matplotlib.path import Path
import numpy as np
random.seed(17)

xgrid = np.arange(-83.5,-82.8,.004)
ygrid = np.arange(42.1, 42.6, .004)
xmesh, ymesh = np.meshgrid(xgrid,ygrid)

points = np.vstack((xmesh.flatten(),ymesh.flatten())).T

for j in Districts:
    p = Path(j.Coordinates[0])
    grid = p.contains_points(points)
    print(j," : ", points[random.choice(np.where(grid)[0])])
    point = points[random.choice(np.where(grid)[0])]
    j.RandomLong = point[0]
    j.RandomLat = point[1]
```

Comment this code so you understand it. Search documentation to understand the parts you are not familiar with. This is an essential skill!

**Step 5:** Now each district has 1 latitude/longitude point that represents it. We want to use that point to find the present-day median household income of that district. To do this we need to identify the census tract code for the random point from each district. This can be obtained using the following API:

<https://geo.fcc.gov/api/census/>

Add the tract code to the appropriate attribute in our class.

### **Step 6 Background Information.**

Redlining played a critical role in preserving and exacerbating racial wealth inequality in the United States. Appreciation of property value is one of the key mechanisms of intergenerational wealth transfer in the United States. By blocking access to home-ownership by limiting access to government insured mortgages (the alternatives were twice as expensive or more), African American citizens were not allowed to benefit from the infrastructure investments of the United States government. This mechanism is detailed in the following article:

<https://www.npr.org/2017/05/03/526655831/a-forgotten-history-of-how-the-u-s-government-segregated-america>. Additionally, it is important to know that while official government redlining was banned in the 1960s, there is some evidence of algorithmic redlining going on in the present.

- <https://revealnews.org/article/for-people-of-color-banks-are-shutting-the-door-to-homeownership/>
- <https://www.washingtonpost.com/news/wonk/wp/2018/03/14/the-senate-rolls-back-rules-meant-to-root-out-discrimination-by-mortgage-lenders/>

**Now** we will check for evidence of the legacy of redlining in Detroit.

- A. Sign up for an API key for the US Census at : [https://api.census.gov/data/key\\_signup.html](https://api.census.gov/data/key_signup.html)
- B. Examine the documentation at <https://api.census.gov/data/2018/acs/acs5/variables.html> and <https://api.census.gov/data/2018/acs/acs5/examples.html> and use your census tract info to find the MEDIAN HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2015 INFLATION-ADJUSTED DOLLARS) for each tract. You will need to use your API and JSON skills from this course.
  - a. Note you can get all of them at once by just specifying the state and then parse the larger json file to get the specific data you need for each district.**
- C. Add the median household income to the appropriate attribute.

### **Step 7.**

For each district grade (A, B, C, D). Calculate the mean and median of the median household income data that you gathered. Assign those to the variables that are descriptively named with the following pattern:

A\_mean\_income

A\_median\_income

B\_mean\_income

B\_median\_income

### Step 7:

Use a list comprehension or other method to combine all the qualitative description strings for each district category. You should have 4 strings, (one each for A, B, C, and D). For each of these strings use a regex operation to split the strings into words.

Find the 10 most common words that are unique to each category (meaning don't include words that are common to all 4 categories). The best answers will find a way to exclude filler words such as 'the', 'of', 'and' etc.

Assign those most common words to the following variables as a list

A\_10\_Most\_Common

B\_10\_Most\_Common

C\_10\_Most\_Common

D\_10\_Most\_Common

### Bonus 1: 5 points

Modify the map code above so that the edgecolor is color of the district grade and the district is filled in with a color based on its median income. Use the matplotlib documentation and/or google to figure out how to create use a color gradient.

**Bonus 2: 5 points** Write 1 paragraph that responds to one of the following prompts. All good faith (in terms of effort) responses, regardless of content or point of view will receive the bonus points.

Did the results in Step 6 or 7 surprise you?

Have these 2 HW assignments changed your prior understanding of segregation in any way? What systemic effects would you want to simulate with the python skills you have now?