Kaitlyn Perham

Follow

Custom function - classifier

Reclassifying NH Lakes with a custom function

The object of this project was to create and use a custom function in Python. I created a function to classify lakes in New Hampshire in 2 classes: under 1 sq km, and those over 1 sq km.

```
import matplotlib.pyplot as plt
import geopandas as gpd
```

I'm going to use the lakes and ponds shapefile available on NH Granit.

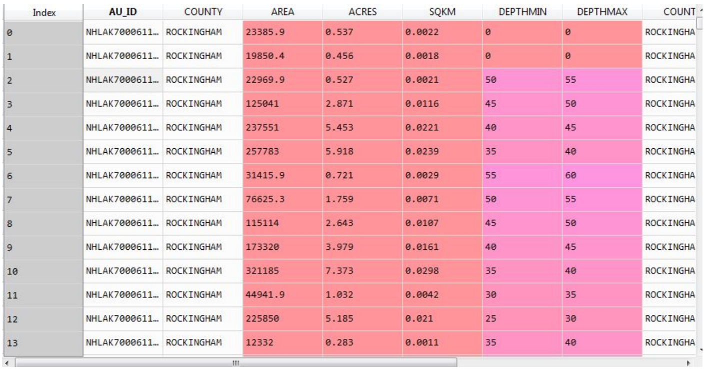
```
fp = "C:/Users/Kaitlyn/Desktop/GeoPython/Reclass/data/NH_lakes_polygons.shp"
```

```
data = gpd.read_file(fp)
```

Examining the columns: I'm going to get rid of irrelevant columns in the shapefile, selecting just the county, area, and Lake names for this analysis.

```
data.head(2)
```

```
selected_cols = ['COUNTY', 'AREA', 'LAKE', 'geometry']
data = data[selected_cols]
```



Index	AU_ID	COUNTY	AREA	ACRES	SQKM	DEPTHMIN	DEPTHMAX	COUNT
0	NHLAK7000611…	ROCKINGHAM	23385.9	0.537	0.0022	0	0	ROCKINGHA
1	NHLAK7000611…	ROCKINGHAM	19850.4	0.456	0.0018	0	0	ROCKINGHA
2	NHLAK7000611…	ROCKINGHAM	22969.9	0.527	0.0021	50	55	ROCKINGHA
3	NHLAK7000611…	ROCKINGHAM	125041	2.871	0.0116	45	50	ROCKINGHA
4	NHLAK7000611…	ROCKINGHAM	237551	5.453	0.0221	40	45	ROCKINGHA
5	NHLAK7000611…	ROCKINGHAM	257783	5.918	0.0239	35	40	ROCKINGHA
6	NHLAK7000611…	ROCKINGHAM	31415.9	0.721	0.0029	55	60	ROCKINGHA
7	NHLAK7000611…	ROCKINGHAM	76625.3	1.759	0.0071	50	55	ROCKINGHA
8	NHLAK7000611…	ROCKINGHAM	115114	2.643	0.0107	45	50	ROCKINGHA
9	NHLAK7000611…	ROCKINGHAM	173320	3.979	0.0161	40	45	ROCKINGHA
10	NHLAK7000611…	ROCKINGHAM	321185	7.373	0.0298	35	40	ROCKINGHA
11	NHLAK7000611…	ROCKINGHAM	44941.9	1.032	0.0042	30	35	ROCKINGHA
12	NHLAK7000611…	ROCKINGHAM	225850	5.185	0.021	25	30	ROCKINGHA
13	NHLAK7000611…	ROCKINGHAM	12332	0.283	0.0011	35	40	ROCKINGHA

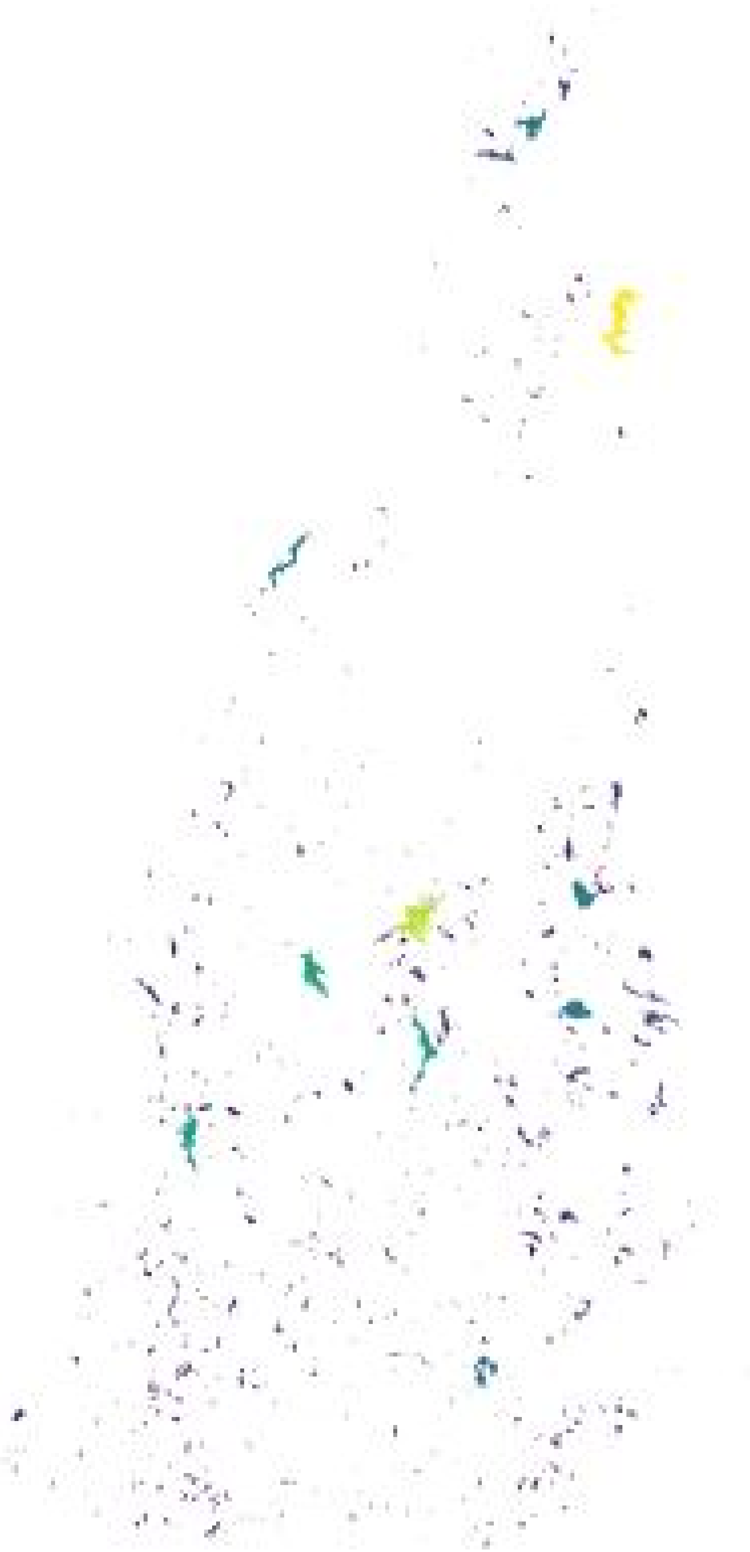
″

A plot of lakes, symbolized by area

```
data.plot(column='AREA', linewidth=0.05)
```

And remove extra white space

```
plt.tight_layout()
```



″

Now, to create a binary classifier - everything under 1 sq km in one class, everything over 1 sq mk is in another:

```
def binaryClassifier(row, source_col, output_col, threshold):
    if row[source_col] < threshold:
        row[output_col] = 0
    else:
        row[output_col] = 1
    return row
```

Calculating the area of lakes, and converting from meters into sq km

```
data['area_km2'] = data['AREA'] / 1000000
```

```
l_mean_size = data['area_km2'].mean()
```

Create an empty column for the classifier to reside in

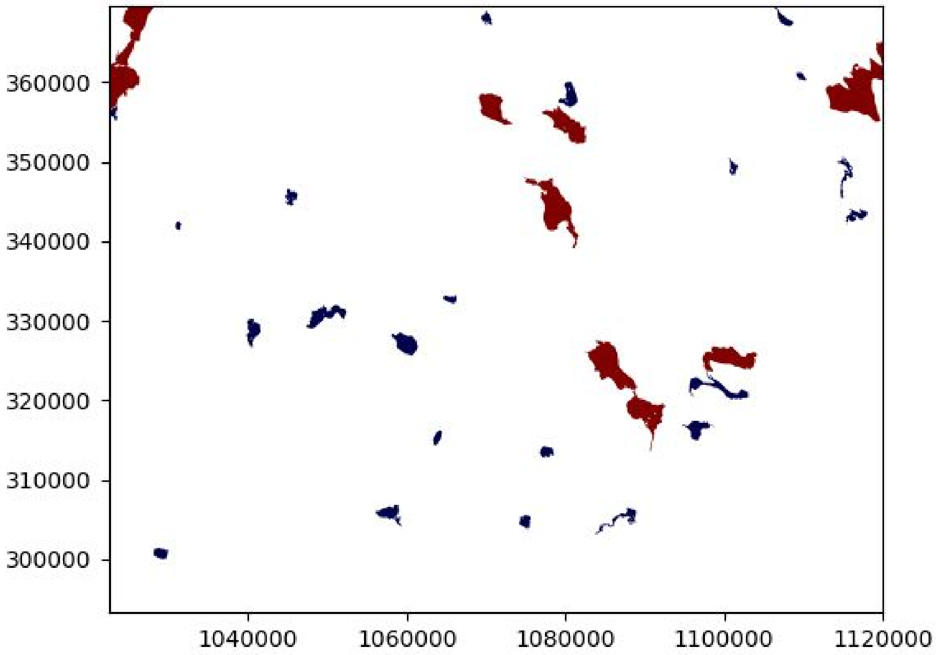
```
data['small_big'] = None
```

Now apply the classifier to the data:

```
data = data.apply(binaryClassifier, source_col='area_km2', output_col='small_big', threshold=l_mean_size, axis=1)
```

And here is the resulting plot (zoomed in for detail):

```
data.plot(column='small_big', linewidth=0.05, cmap="seismic")
```



″

Save the file as a new shapefile:

```
outfp_data = r"C:/Users/Kaitlyn/Desktop/GeoPython/Reclass/NH_lakes.shp"
data.to_file(outfp_data)
```

Previous Next