Using Python Programming to Calculate the Area of an Irregular Polygon

By Matt Palmer

Python Coding Language

- Python is a programming language that can be used among many applications
  - **Geographic Information Science and Technology**
  - Web and Internet Development
  - Scientific and Numeric Computing
  - **Educational purposes**
  - **Software development**
  - Business applications

Literature Reviews

“”Cartesian formulas for curvature, circumradius, and circumcenter for any three two-dimensional points”

- Finding three different values:
  - Curvature
  - Circumradius
  - Circumcenter
- Do not include these values into my function but Shoelace Formula is used
- This study uses three points in their calculations; my study uses many more coordinate points but the function should work regardless
- Shoelace formula
  - Also called Gauss’s area formulas and Surveyor’s formula
  - Can result in negative number therefore use absolute value

“A new approach (extra vertex) and generalization of Shoelace Algorithm using in convex polygon (Point-in-Polygon)”

- Shoelace formula is used to find the area of polygons along the Cartesian coordinate system
  - Our study the coordinate points were converted to squared meters
- This article’s study used python programming language to test the formula by segmenting lines and polygons
- PiP (Point-in-Polygon)
  - Tool used in GIS
  - “With a given polygon and an arbitrary point, determine whether the point will be within the polygon.”

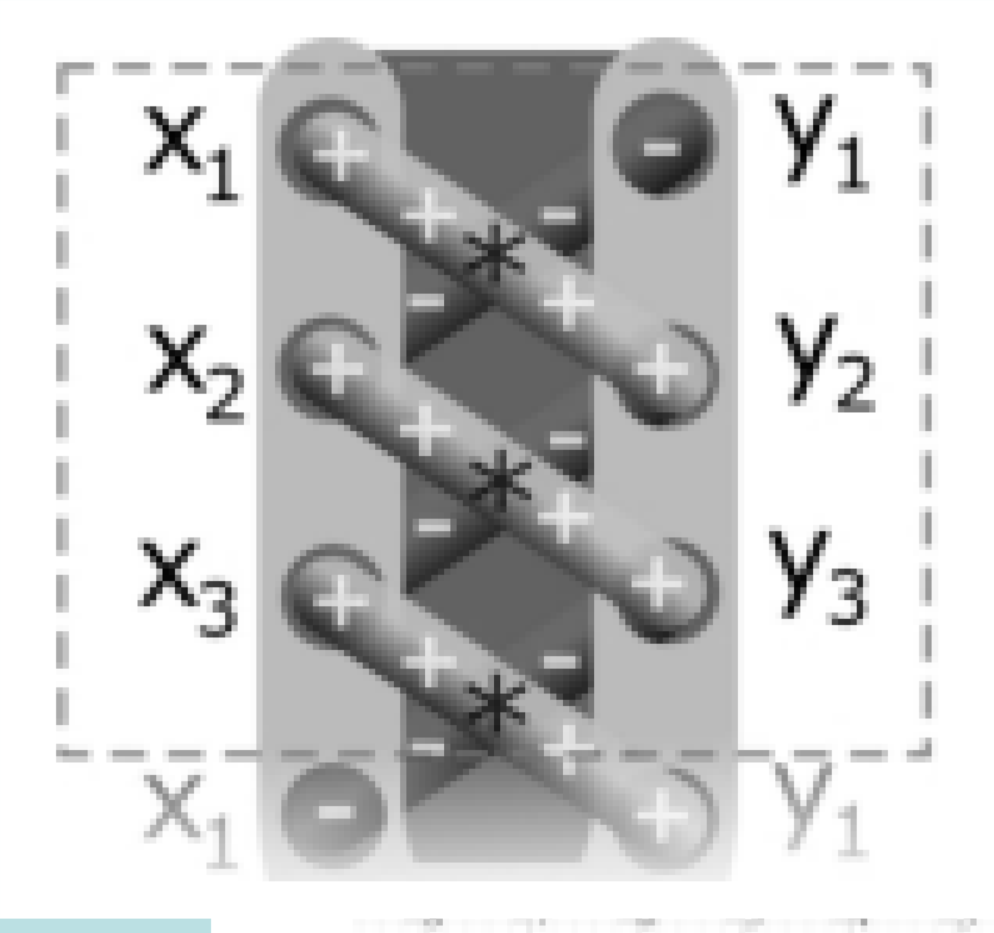
    -Ochilbek
  - Used in video gaming and with polygons that have high number of edges

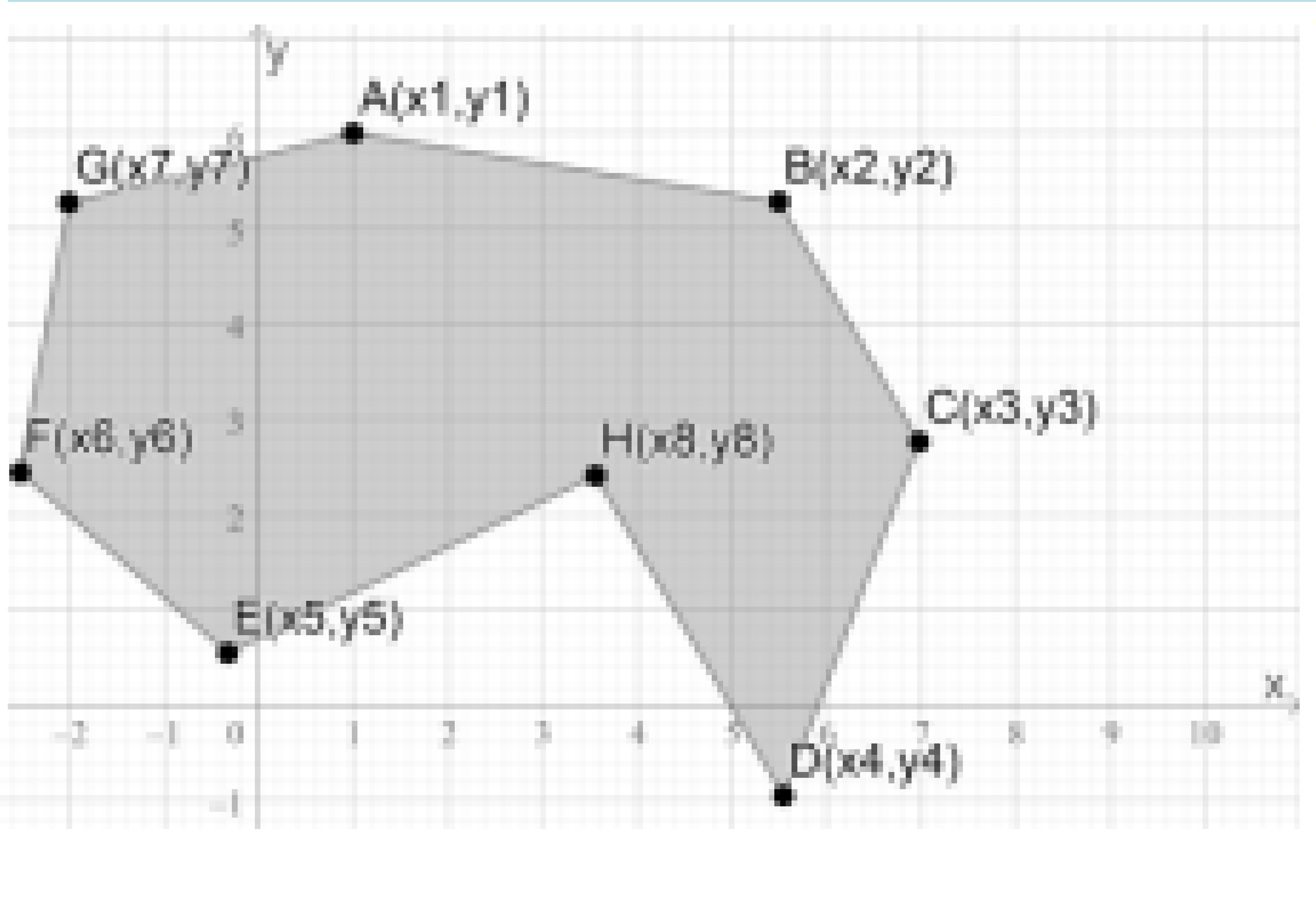
Research & Data

Shoelace Formula

- A formula used when trying to find the area of a polygon especially when the polygon consists on many sides or coordinate points
- When performing cross multiplication, the formula could be described as looking like a shoelace being tied up due to the order of operation

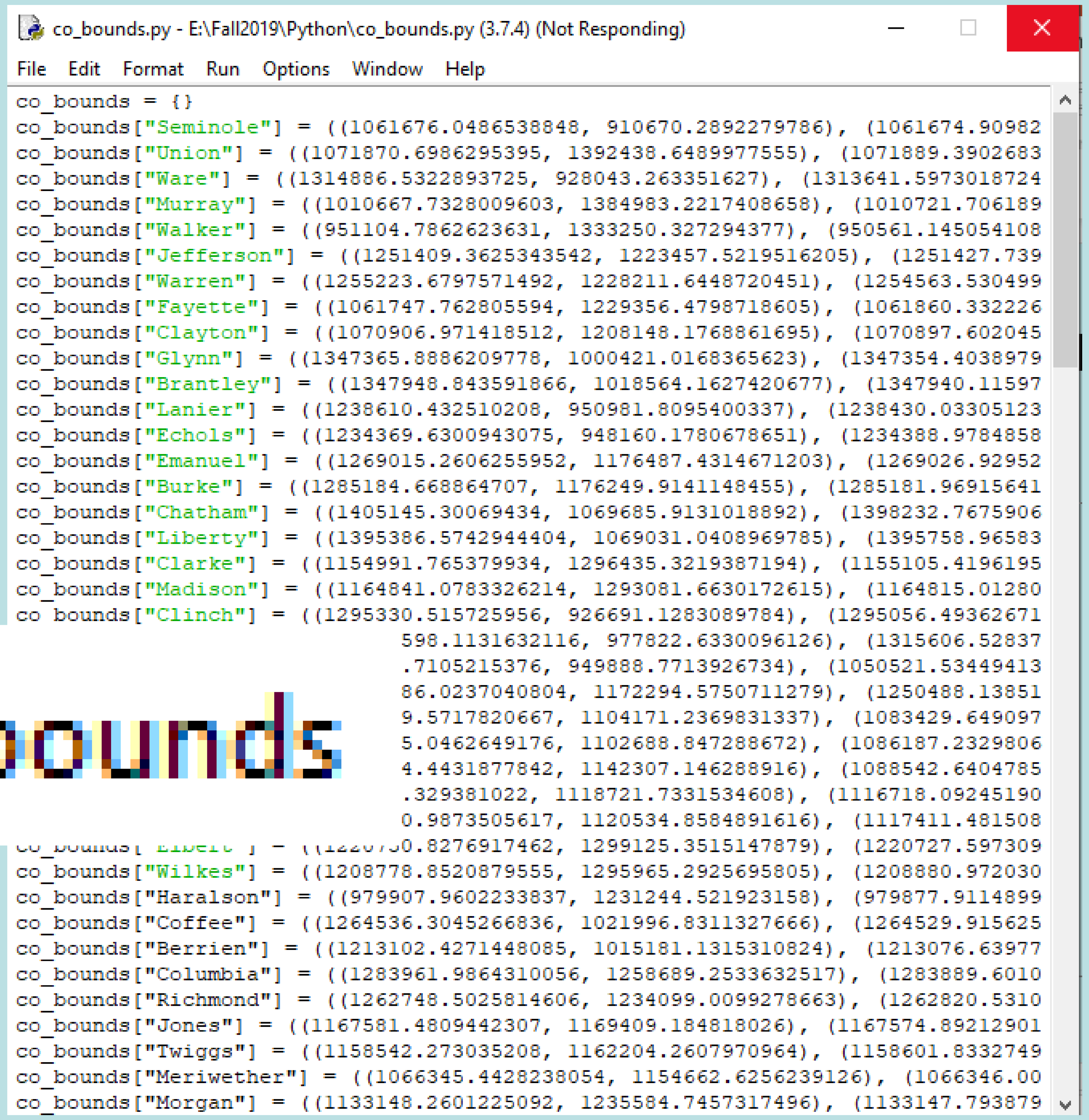
$$A = \frac{1}{2}\left|(x_1y_2 + x_2y_3 + \ldots + x_ny_1) - (y_1x_2 + y_2x_3 + \ldots + y_nx_1)\right|$$





GA County Data – Testing

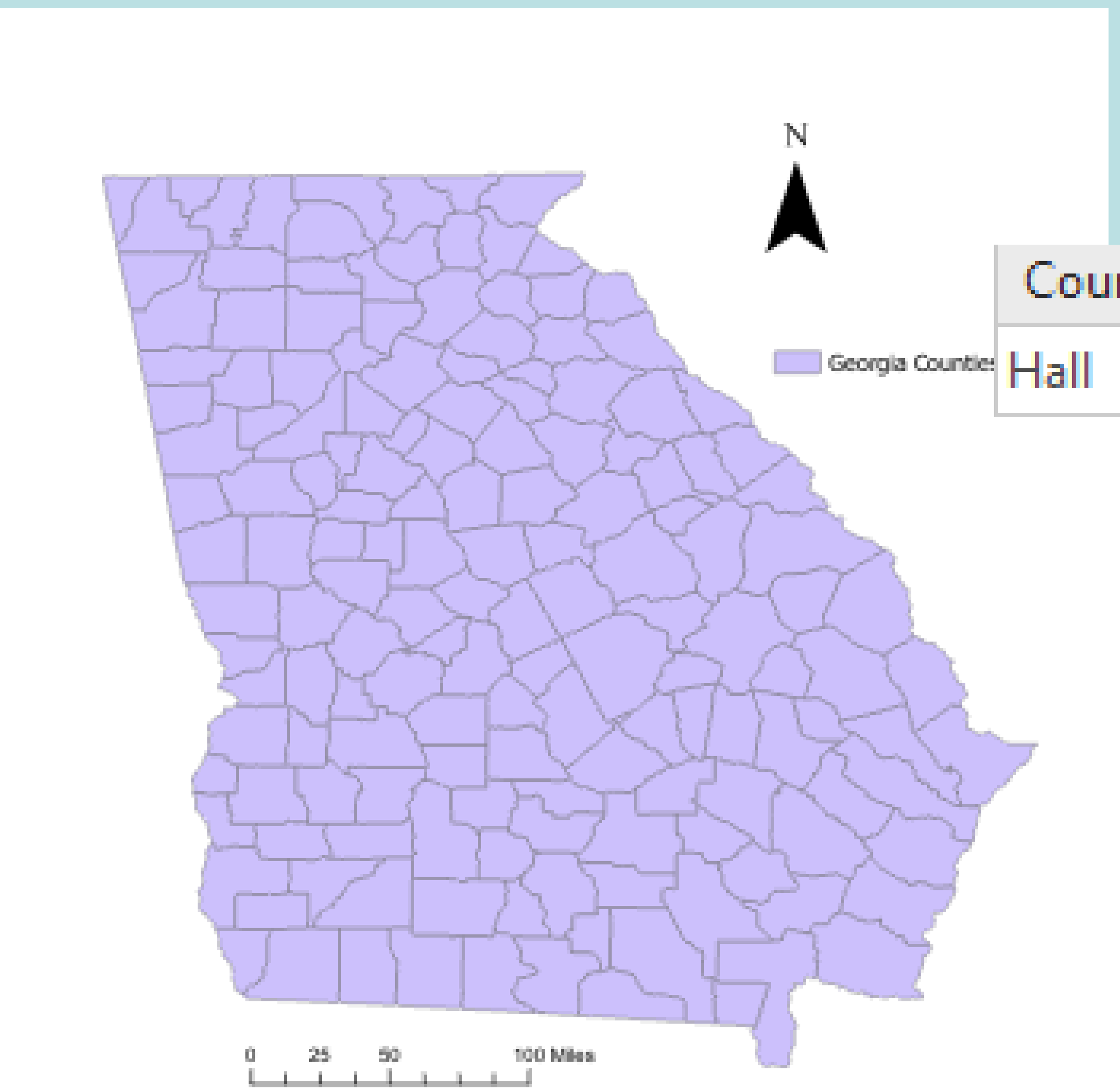
- File provided contained coordinate point data
- Used this to test the function written in python
- Need to convert these points into a polygon and calculate the area in squared meters



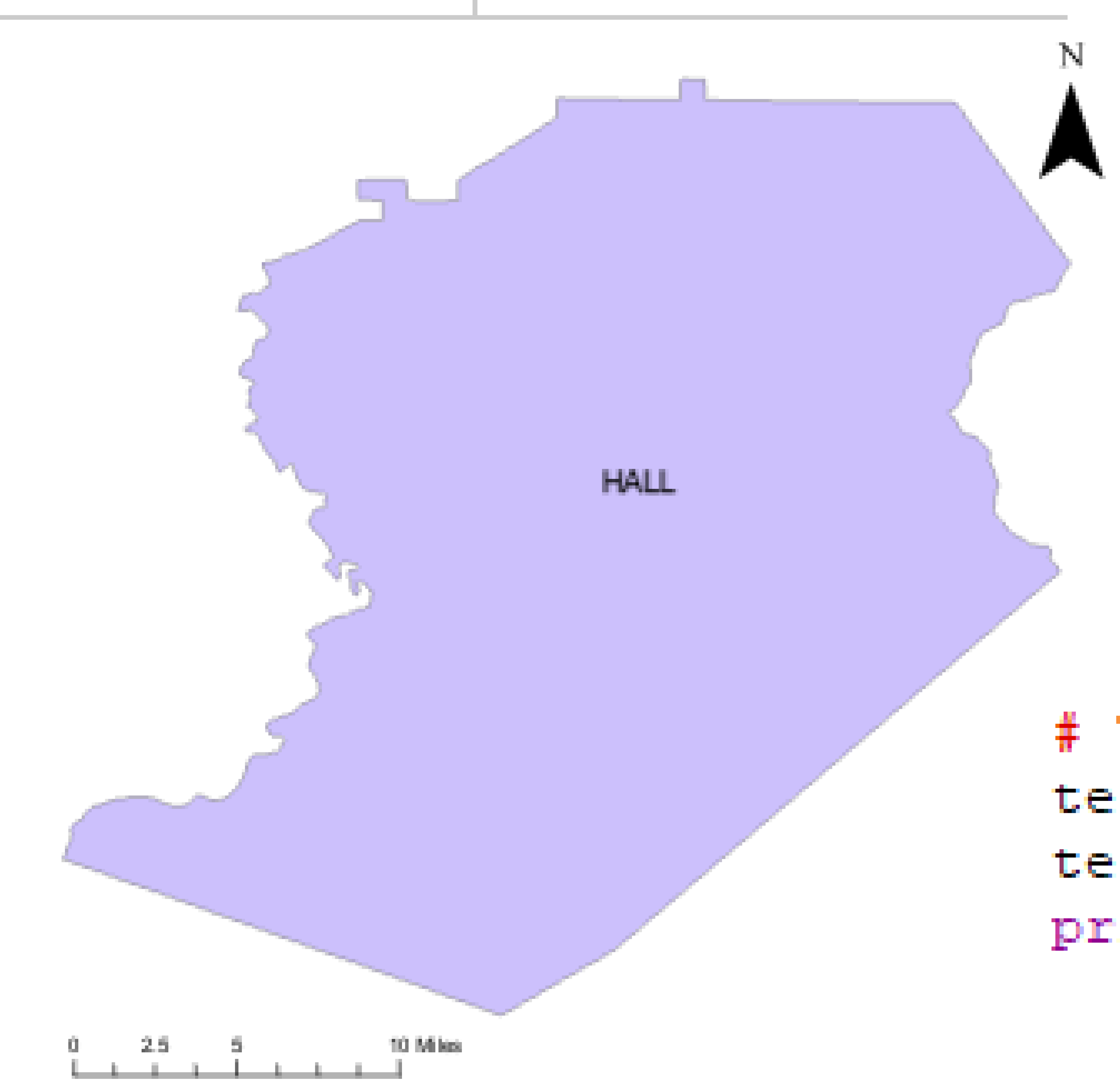
Research & Data

Study Area

- Georgia Counties
- County area in squared meters



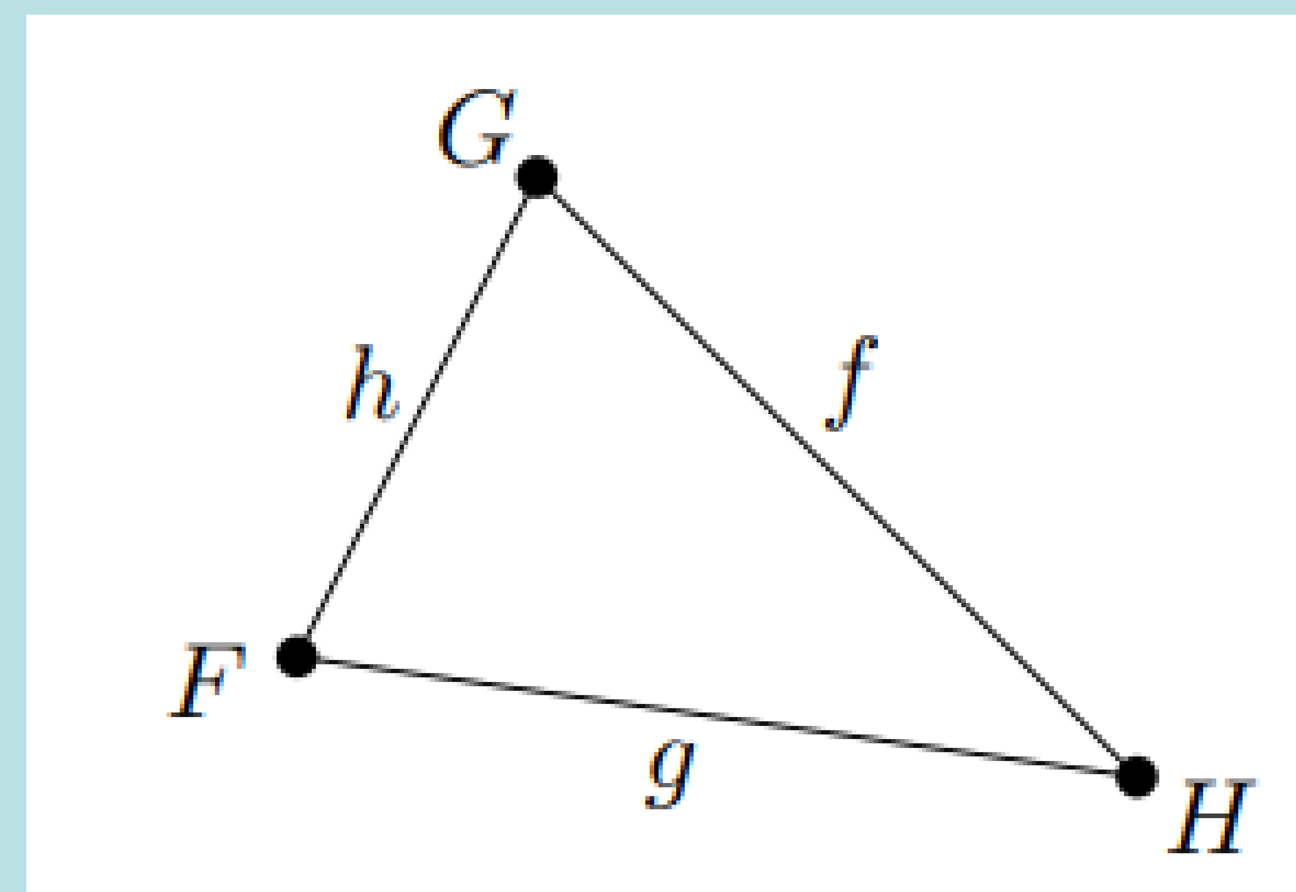
CountyName	CountyArea
Hall	1111864546.63348




```
# Run function on any county's coordinates
counties = cb.get_co_bounds()
county_area = calc_area(counties['Hall'])
print(county_area)
```

Other Uses

- The GA counties consist of many points
- Should be able to use the python function on simple polygons such as a triangle with three given points.
  - This is mentioned in literature reviews




```
# Test function on a simple triangle polygon
test_polygon = [(-2, -2), (-2, 2), (2, 2)]
test_area = calc_area(test_polygon)
print(test_area)
```

Methods

Step 1-Defining the function

- This is where we are going to create the function
- Part of the code where were are taking the polygon argument
- Called this function calc_area and referencing it back to points

```
def calc_area(points):
```

Step 2-Assign area to zero

- Having the starting value for the area assigned to zero provides easier workability
- We are working with (X,Y) coordinates therefore assigning y to the last tuple for a given county is applied to the code
- Indicate the last value of the list with a value of -1

```
# starting area value of zero
area = 0

# y value starts as last tuple for a given county
y = points[-1]
```

Methods

Step 3-Looping tuples

- This step makes it so that the function can loop through each tuple of tuple
- Each point has an (X,Y) pair so the Shoelace formula is applies here

```
#loop through each (x,y) pair
for x in points:
    area += (x[0] * y[1]) - (x[1] * y[0])
    y = x
```

Step 4-Return a number

- Finally, the program returns the area of the polygon
- Absolute value is applied to avoid negative numbers
- Dividing the area in half provides the answer to the equation
- Takes one tuple and returns one number

```
return abs(area / 2)
```

Preliminary Results

- INPUT

```
import co_bounds_converted as cb


def calc_area(points):

    # starting area value of zero
    area = 0

    # y value starts as last tuple for a given county
    y = points[-1]

    #loop through each (x,y) pair
    for x in points:
        area += (x[0] * y[1]) - (x[1] * y[0])
        y = x

    return abs(area / 2)


# Test function on a simple triangle polygon
test_polygon = [(-2, -2), (-2, 2), (2, 2)]
test_area = calc_area(test_polygon)
print(test_area)

# Run function on any county's coordinates
counties = cb.get_co_bounds()
county_area = calc_area(counties['Hall'])
print(county_area)
```

- OUTPUT
  - Returns the area (m^2) when points are input into the program
  - Returns the area of the county that are assigned to print

CountyName	CountyArea
Hall	1111864546.63348

```
8.0
1111868721.272461
>>>
```

Room for Growth

- Change the formatting of the program so that it is easier for the user to use
- Allow the program to add multiple county areas together
- Conversion from latitude/longitude to Cartesian coordinate system
- Convert to squared kilometers

```
def calc_area(points):
    area = 0
    y = points[-1]
    for x in points:
        area += (x[0] * y[1]) - (x[1] * y[0])
        y = x
    return abs(area/2)
```

References

- https://www.python.org/about/apps/
- Ratliff, Hunter. *Cartesian formulas for curvature, circumradius, and circumcenter for any three two-dimensional points*
- Ochilbek, Rakhmanov. (2018). *A new approach (extra vertex) and generalization of Shoelace Algorithm usage in convex polygon (Point-in-Polygon).* 2018 14th International Conference on Electronics Computer and Computation (ICECCO). 07 February 2019

UNG UNIVERSITY of NORTH GEORGIA
LEWIS F. ROGERS INSTITUTE FOR ENVIRONMENTAL AND SPATIAL ANALYSIS

Using Python Programming Language to Calculate the Area of an Irregular Polygon

Matt Palmer
University of North Georgia

Introduction

Python Programming Language is one of the most recent coding languages geared towards easy to learn coding applications. In the geospatial field, python can be implemented in many different such as allowing tools in ArcPro to run more quickly and efficiently. This research project focuses on implementing Python Programming into a program that will calculate the area of an irregular polygon. Irregular polygons are used frequently in geospatial technology due to many political boundaries not having the same shapes. This study focuses on the counties found in the state of Georgia to see if python coding can be created to help aid in this problem.

Literature Reviews

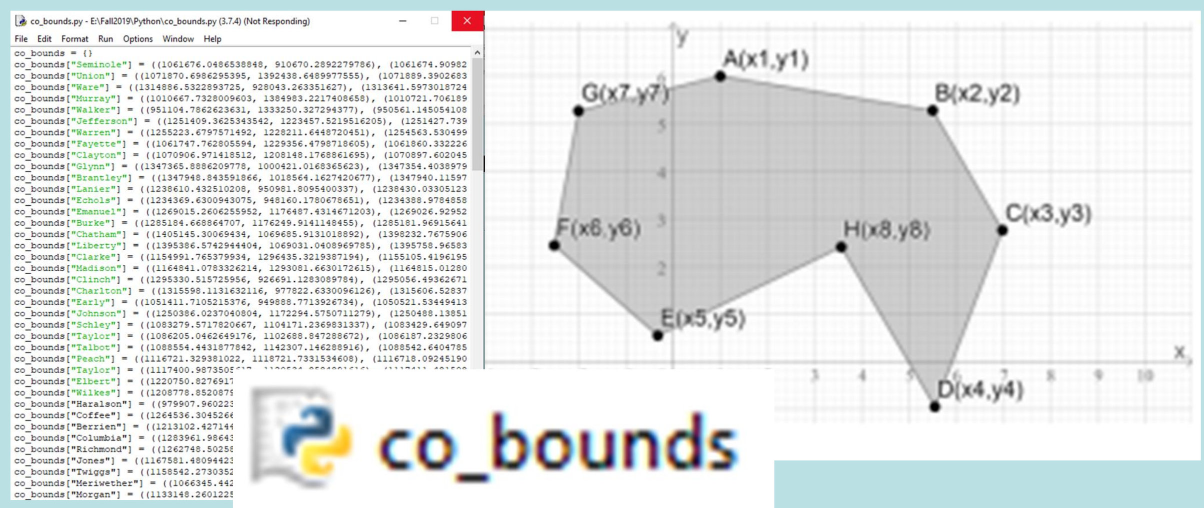
The literature reviews were used to help explain the Shoelace formula more in detail to help create the function for the code. In both the articles, they used this formula to help identify the area of the polygon being studied. It is mentioned that this type of formula is best used when the research involves polygons that have many different coordinate points to use to find the area. This algorithm can be applied to video game applications especially in the fighting games for applying parameters to characters that are being destroyed.

Applications

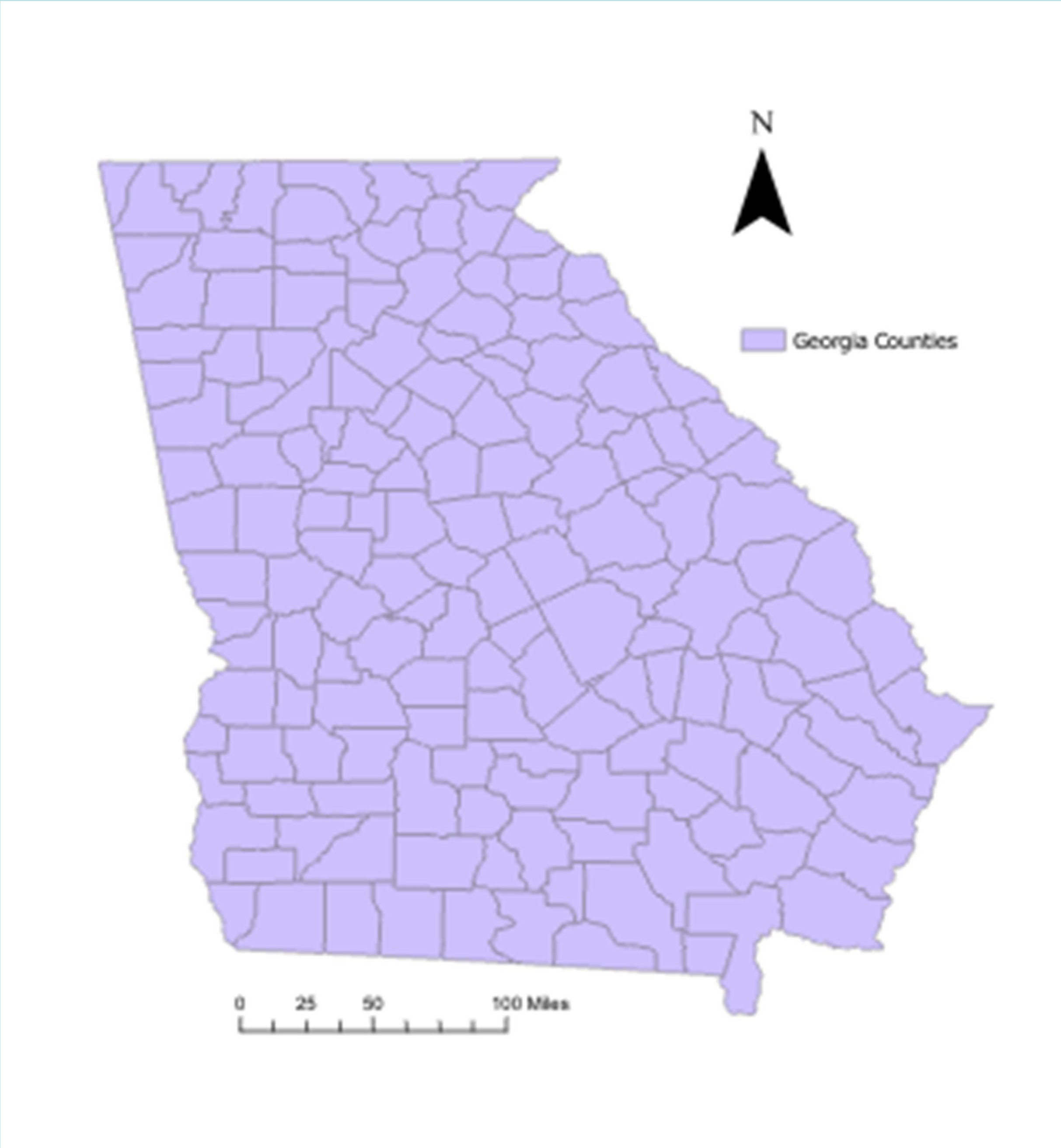
- Geospatial Technology
- Web and Internet Development
- Software Development
- Business Applications
- Educational Purposes

Data

The purpose of using the Georgia counties as a way to test the python code is to make sure that the code is running properly. The co_bounds.py file provided to the class contained coordinate points that reference to the individual counties within the states. Some of these values can have up to hundreds of coordinate points, therefore creating a code that can work with either just a handful of points or many points, such as those found in the file, is important to keep in mind when writing the program.



Study Area



Methods

Step 1-Defining the function

- This is where we are going to create the function

Step 2-Assign area to zero

- Having the starting value for the area assigned to zero provides easier workability
- We are working with (X,Y) coordinates therefore assigning y to the last tuple for a given county is applied to the code
- Indicate the last value of the list with a value of -1

Step 3-Looping tuples

- This step makes it so that the function can loop through each tuple of tuple
- Each point has an (X,Y) pair so the Shoelace formula is applies here

Step 4-Return a number

- Finally, the program returns the area of the polygon
- Absolute value is applied to avoid negative numbers
- Dividing the area in half provides the answer to the equation
- Takes one tuple and returns one number

INPUT/ OUTPUT

```
import co_bounds_converted as cb


def calc_area(points):

    # starting area value of zero
    area = 0

    # y value starts as last tuple for a given county
    y = points[-1]

    #loop through each (x,y) pair
    for x in points:
        area += (x[0] * y[1]) - (x[1] * y[0])
        y = x

    return abs(area / 2)


# Test function on a simple triangle polygon
test_polygon = [(-2, -2), (-2, 2), (2, 2)]
test_area = calc_area(test_polygon)
print(test_area)

# Run function on any county's coordinates
counties = cb.get_co_bounds()
county_area = calc_area(counties['Hall'])
print(county_area)
```

```
8.0
1111868721.272461
>>>
```

CountyName	CountyArea
Hall	1111864546.63348

Improvements

Formatting the program to make it more used friendly is something that can be added to this code to help improve the overall quality and use. Making it so that the user can add multiple different counties together to gather the area of multiple polygons is an improvement that can be made to this project. Adding a conversion factor to the code would make the interaction with the program more interesting as well. This would give the user a way to educate themselves more on the size of Georgia counties.

References

•https://www.python.org/about/apps/
•Ratliff, Hunter. Cartesian formulas for curvature, circumradius, and circumcenter for any three two-dimensional points
•Ochilbek, Rakhmanov. (2018). *A new approach (extra vertex) and generalization of Shoelace Algorithm usage in convex polygon (Point-in-Polygon).* 2018 14th International Conference on Electronics Computer and Computation (ICECCO). 07 February 2019