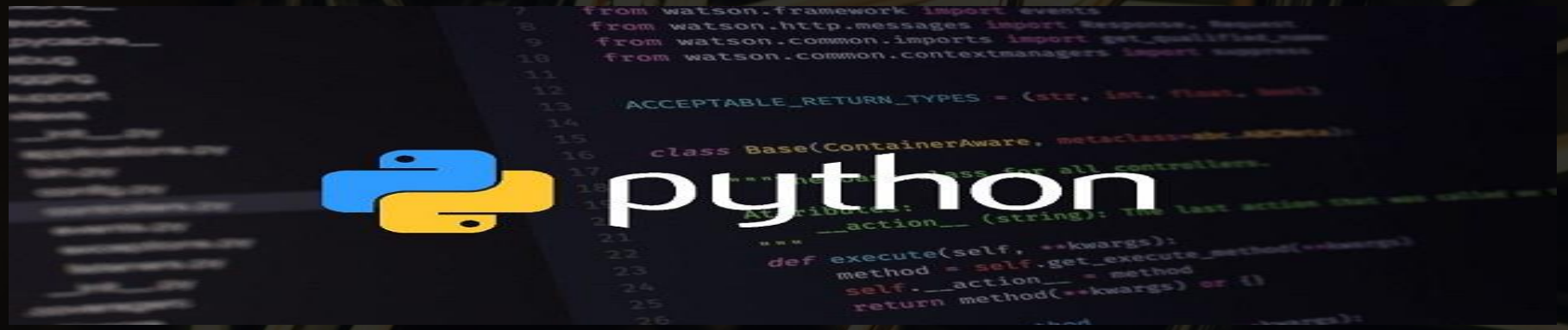
Calculating Area of an Irregular Polygon in python

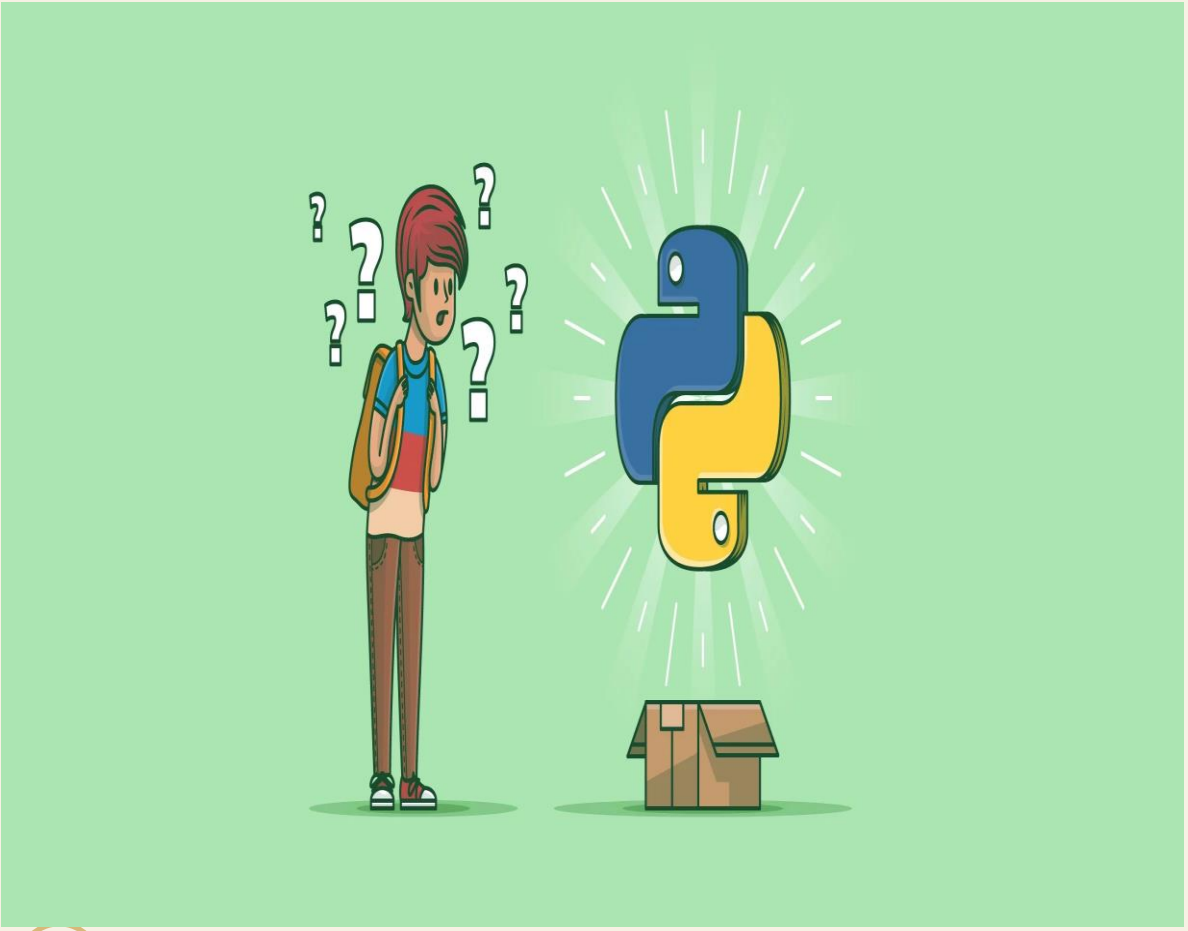
By: Devon Elam



Introduction

What is Python ?

- Python is one of the many high-leveled programming languages that was created in the late 1980s by Guido van Rossum.
- The objective of this project is to create a function that will compute the area of the file co_bounds script that has coordinates attached to them.



Research Data

Shoelace Formula

- Also known as the shoelace algorithm is a mathematical algorithm to determine the area of a simple polygon whose vertices are described by their Cartesian coordinates in the plane.
- Consists of cross-multiplying corresponding coordinates of the different vertices of a polygon to find its area.

$$\text{Area} = \frac{1}{2}\left|\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n\right|$$

$$\text{Area} = \frac{1}{2}\left|x_1 y_2 + x_2 y_3 + \cdots + x_{n-1} y_n + x_n y_1 - x_2 y_1 - x_3 y_2 - \cdots - x_n y_{n-1} - x_1 y_n\right|$$

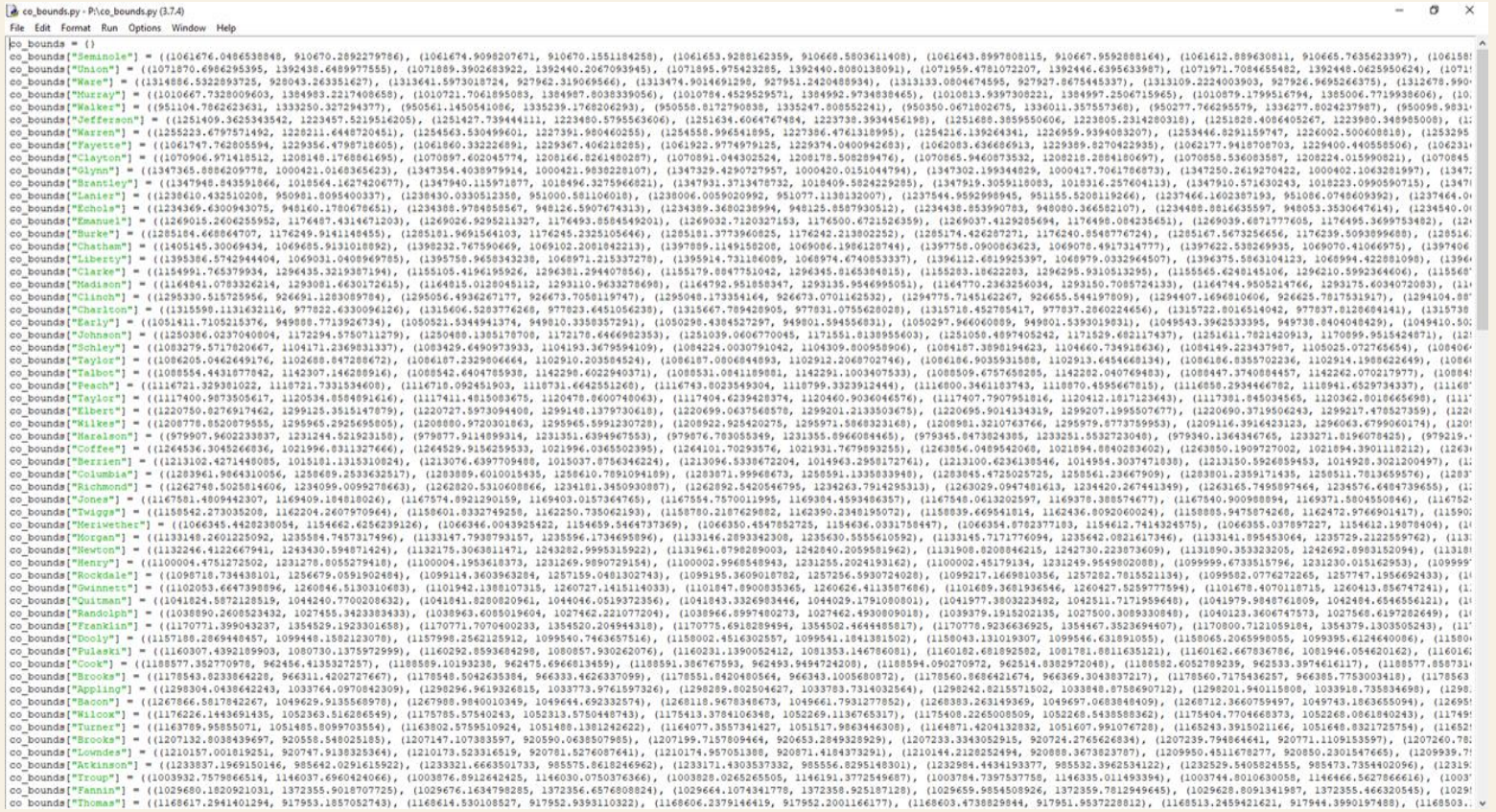
Vertices	x	y
A	2	7
B	10	1
C	8	6
D	11	7
E	7	10

Research Data

GA Counties .PY

co_bounds.py

- Python code contains a list of all Georgia counties and the coordinate data of each one.
- Use the code to test your function , but don't implement it into your code.
- The data is in meters, and the objective is to make the output in meters squared.



Methodology

1) First step is you must import the co_bounds function from the co_bounds module.

```
from co_bounds import co_bounds
```

2) Next, you begin creating the function; you define calc_area(poly) so that the function will calculate the area of a polygon.(poly is the source that the function references too)

```
#area of polygon = abs(((x1y2-y1x2) + (x2y3-y2x3))
+ (xn-1)(yn)+(xn)(y1)-(xn)(yn-1)-(x1)(yn)/2)))
```

3) Set the area to zero since that is what we are looking for, and makes the function simple to calculate.

```
area = 0
#Initialize the area
```

Methodology Cont..

3) Now, we begin the for loop for the function; the values in the tuples are stored as the Cartesian coordinates are cross multiplied with the corresponding coordinates in order to find the area encompassing the polygon and then subtracting it from the surrounding polygon.

```
for i in range(len(poly)-1):
    #Creating a range that corresponds to the index
    #find the area encompassing the polygon and then subtracting it from the surrounding polygon
    #Loop through the polygon points
```

```
p1 = poly[i]
#Set the first point variable
p2 = poly[i+1]
#Set the second point variable
area = area +(p1[0] * p2[1] - p1[1] * p2[0])
#calculate x1y2-y1x2 and add it to the area
```

4) Instead of x,y, I use the variables p1, and p2 just to stay in commonality of poly.

Methodology

5) Now, the for loop is now finished, area is printed and the. The final value will be written as an absolute value and since the area cannot be negative and then in order to find the area within it is divided by two.

```
area = abs(area/2)
#This ends the for loop
#Return the absolute value of the area
return area
```

6) Below, I have set poly to equal random vertices of a quadratic to test the function running outside of co_bounds module.

```
#Test out function with random vertices
#Initialize the points
poly = [(2, 3), (11, 8), (12, 25), (4, 9)]
#Run the function to print
print("The area is",calc_area(poly))
```

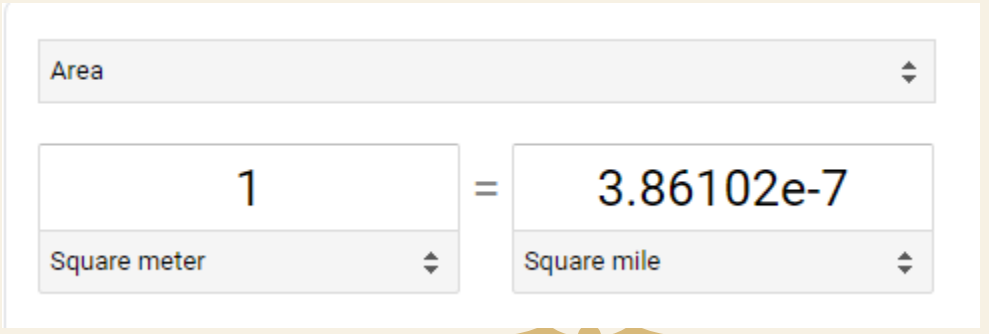
```
The area is 82
```

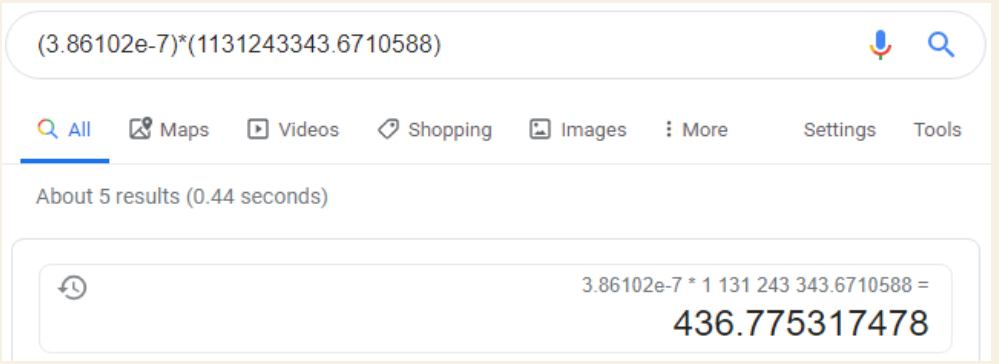
Results

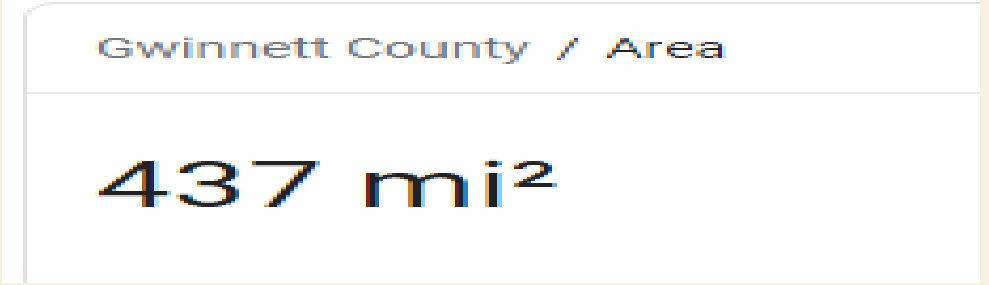
File Edit Format Run Options Window Help

```
#Import the co_bounds function from the co_bounds module
from co_bounds import co_bounds

def calc_area(poly):
    #area of polygon = abs(((x1y2-y1x2) + (x2y3-y2x3)) + (xn-1)(yn)+(xn)(y1)-(xn)(yn-1)-(x1)(yn)/2)

    area = 0
    #Initialize the area is zero beacsue thats what were looking to find

    for i in range(len(poly)-1):
        #Creating a range thacorresponds to the indexes
        #find the area encompassing the polygon and then subtracting it from the surrounding polygon
        #Loop through the polygon points
        p1 = poly[i]
        #Set variable p as a point
        p2 = poly[i+1]
        #Set variable p2 as another point
        area = area +(p1[0]*p2[1]-p1[1]*p2[0])
        #calculate x1y2-y1x2 and add it to the area

    area = abs(area/2)
    #This ends the for loop
    #Return the absolute value of the area
    return area

print("The area is", calc_area(co_bounds["Gwinnett"]), 'meters')

#prints the area
```

```
================== RE:
1131243343.6715088
```







Ways to Improve

- Be able to let the script format so that the output is able to convert to square miles itself.
- Let you be able to type the name of the city and it already identifies the county and states info.
- Run two county scripts and allow it to compare area to each other.

Works Cited

- "Area of a Polygon (Coordinate Geometry)." Area of Any Polygon (Coordinate Geometry) - Math Open Reference, www.mathopenref.com/coordpolygonarea.html.
- "Generate Points That Lie inside Polygon." *Geographic Information Systems Stack Exchange*, 1 Apr. 1961, gis.stackexchange.com/questions/6412/generate-points-that-lie-inside-polygon.
- Marsh, Charles. "Computational Geometry in Python: From Theory to Application." Toptal Engineering Blog, Toptal, 21 Jan. 2014, www.toptal.com/python/computational-geometry-in-python-from-theory-to-implementation.

e program language
ctive of this project
nction that will
a of the file
has coordinates of
a counties attached
e coordinates are
a is set to be in
his project you need
ea into meter's

earch/Data

ace Formula
the shoelace algorithm
l algorithm to determine
ple polygon whose
ribed by their Cartesian
e plane.
ss-multiplying
ordinates of the
of a polygon to find its

File Edit Format Run Options Window Help

```
#Import the co_bounds function from the co_bounds module
from co_bounds import co_bounds

def calc_area(poly):
    #area of polygon = abs(((x1y2-y1x2) + (x2y3-y2x3)) + (xn-1)(yn)+(xn)(y1)-(xn)(yn-1)-(x1)(yn)/2)

    area = 0
    #Initialize the area is zero beacsue thats what were looking to find

    for i in range(len(poly)-1):
        #Creating a range thacorresponds to the indexes
        #find the area encompassing the polygon and then subtracting it from the surrounding polygon
        #Loop through the polygon points
        p1 = poly[i]
        #Set variable p as a point
        p2 = poly[i+1]
        #Set variable p2 as another point
        area = area +(p1[0]*p2[1]-p1[1]*p2[0])
        #calculate x1y2-y1x2 and add it to the area

    area = abs(area/2)
    #This ends the for loop
    #Return the absolute value of the area
    return area

print("The area is", calc_area(co_bounds["Gwinnett"]), 'meters')

#prints the area
```

```
1131243343.6715088
```

Gwinnett County / Area

1) First step is you must in
co_bounds function fro
module.
2) Next, you begin creatin
define calc_area(poly) s
will calculate the area of
the source that the funct
3) Set the area to zero sin
are looking for, and mak
simple to calculate.
3) Now, we begin the for lo
the values in the tuples a
Cartesian coordinates a
with the corresponding c
to find the area encompa
and then subtracting it fr
surrounding polygon.
4) Instead of x,y, I use the
p2 just to stay in commo
5) Now, the for loop is now
printed and the. The fin
written as an absolute v
area cannot be negative
to find the area within it
6) Below, I have set poly t
vertices of a quadratic to
running outside of co_bo