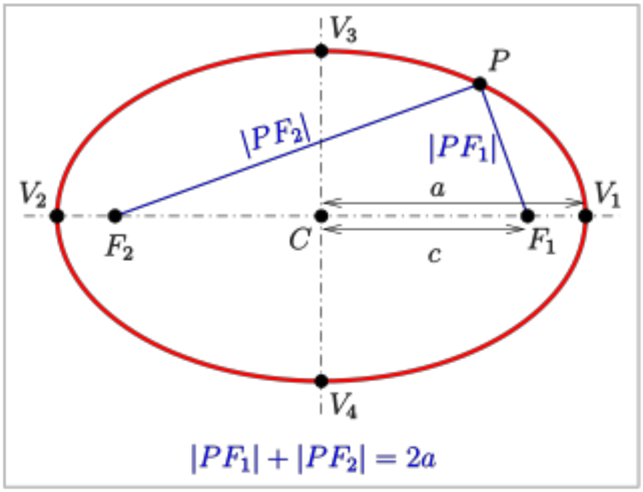
Overlapping Ellipses

An ellipse is a curve in a plane surrounding two focal points such that the sum of the distances to the two focal points is constant for every point on the curve. – Wikipedia.



In this assignment you will use a Monte Carlo Simulation to estimate the overlap between two Ellipses.

The assignments is divided into 4 parts:

1. Ellipse Class
2. Structure Chart
3. Implementation
4. Unit Testing

Part 1: Ellipse Class

A. Write a class to capture the necessary information of an ellipse.
B. Write code to thoroughly test your implementation of the ellipse class including:
    a. A constructor that accepts no variables.
    b. A constructor that accepts all appropriate variables.
    c. Setting and getting the appropriate variables.
    d. A function that returns the area of the ellipse
    e. A function that returns the circumference of the ellipse.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, “I have not given or received any unauthorized assistance on this assignment.” Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.

Part 2: Structure Chart

Using the top-down strategy discussed in class, produce a structure chart to capture the necessary steps for computing the overlap of two ellipses using the Monte Carlo Technique.

A. Each block of the chart should capture the requirements of a function.
B. Label each edge with the input and output of the function, similar to the examples given in class.

NOTE: DO NOT ATTEMPT TO IMPLEMENT YOUR SOLUTION UNTIL YOU THOROUGHLY UNDERSTAND THE PROBLEM AND HAVE MAPPED OUT YOUR SOLUTION.

Submission: Submit a single .pdf file containing the structure chart. Do not zip or archive the file. Your submission must include your name, date and the honor statement, "I have not given or received any unauthorized assistance on this assignment."

Grading: There is no partial credit for this problem.

Part 3: Implementation

Implement your solution described by your structure chart. Your function should accept two Ellipse objects (defined in part 1) and return the area of their overlap. Use a Monte Carlo simulation to estimate the overlap. If they do not overlap, return 0.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, “I have not given or received any unauthorized assistance on this assignment.” Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.

Part 4: Unit Testing

For this part, you will thoroughly test your implementation. For EVERY function, beginning with the lowest level functions:

A. Print the name of the function.
B. Print the doctstring of the function.
C. Print the rationale for the test (e.g. “Testing when a point falls within an Ellipse.”)
D. Print the specific input used to evaluate that rationale.
E. Print the result.
F. Repeat C – E until all relevant cases are tested for that function.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, “I have not given or received any unauthorized assistance on this assignment.” Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.