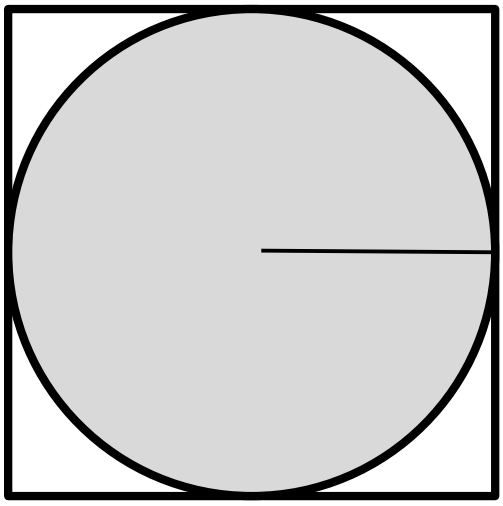
Monte Carlo Simulation: Calculating π

Background

In section today we will try to use random numbers to calculate π. To understand how a precise mathematical constant can be estimated using randomness we first need to refresh some basic geometry.

In the figure on the right the circle and the square have the following areas:



Area Circle: $A_C = \pi r^2$
Area Square: $A_S = (2r)^2 = 4r^2$

The ratio of the two areas is: $\frac{A_C}{A_S} = \frac{\pi r^2}{4r^2}$

And solving for π we get: $\pi = \frac{4A_C}{A_S}$

If we have an estimate for the ratio of the area of the circle to the area of the square we can solve for pi. The challenge becomes estimating this ratio.

Monte Carlo Simulations

This is where we can take advantage of how quickly a computer can generate pseudorandom numbers. There is a whole class of algorithms called Monte Carlo simulations that exploit randomness to estimate real world scenarios that would otherwise be difficult to explicitly calculate. We can use a Monte Carlo simulation to estimate the area ratio of the circle to the square.

Imagine you randomly drop grains of sand into the area of the square. By counting the total number of sand grains in the square (all of them since you're an accurate dropper) to the number of sand grains inside the circle we get this estimate. Multiply the estimated ratio by four and you get an estimate for π. The more sand grains you use the more accurate your estimate of π.

Today's Problem

Write a program that estimates π for $[10^0,10^1,10^2,10^3,10^4,10^5,10^6,10^7,10^8]$ random points. Your output should look something like this:

```
1x10^0: 4.0
1x10^1: 2.8
1x10^2: 3.36
1x10^3: 3.18
1x10^4: 3.17
1x10^5: 3.13904
1x10^6: 3.141276
1x10^7: 3.1417744
1x10^8: 3.1414218
```

Step 1: Plan It Out

What are some major design considerations you need to think about? How can the problem be broken into a series of smaller problems?

What are some useful functions to create? For each function write a doc string.

Hint: From the random module: random.random() returns a float between 0 and 1.

Step 2: Implement the smaller problems.

Step 3: Write the main function.

Monte Carlo Simulation: Calculating π
(Solutions)

Step 1: Plan It Out

Design Considerations:

- How general to make each function? Can the radius of the circle be changed?
- Where to center the square in the coordinate plane?

Some useful functions might include:

- random_around_zero(r): return a random float between [-r,r]
- random_throw(r): return the tuple (x,y) of a random point inside a square [(-r,r),(r,r),(r,-r),(-r,-r)]
- in_circle(r,x,y): return True if the point (x,y) is in a circle of radius r centered at (0,0). Return False otherwise.
- number_in_circle(n_total,r=0.5): test n_total number points and return the number that are in a circle of radius r (default r=0.5)

Step 2: Implement the smaller problems.

```
def random_around_zero(r):
    """
    Return a random float in the range [-r, r]
    """
    width = 2 * r
    return random.random() * width - r

def random_throw(r):
    """
    Return a random point in the square
    [(-r, r), (r, r), (r, -r), (-r, -r)]
    """
    x = random_around_zero(r)
    y = random_around_zero(r)
    return x, y

def in_circle(r, x, y):
    """
    Return True if (x, y) is in a circle with radius r
    centered at 0. Otherwise False.
    """
    distance_from_center = (x * x + y * y) ** 0.5
    return distance_from_center <= r

def number_in_circle(n_total, r=0.5):
    """
    Test n_total random points (x, y) in a square with
    sides 2r and return the number of points that are
    inside a circle of radius r.
    """
    n_circle = 0
    for i in xrange(n_total):
        x,y = random_throw(r)
        if in_circle(r, x, y):
            n_circle += 1
    return n_circle
```

Step 3: Write the main function.

```
def main():
    for i in range(9):
        n_total = 10 ** i
        n_circle = number_in_circle(n_total)
        pi = 4.0 * n_circle / n_total
        print '1x10^' + str(i) + ':', pi


if __name__ == '__main__':
    main()
```

Name:

Section:

Email:

Are you planning to work with a partner for the project? If yes, do you have a partner already? Who is it?

What are some ideas you have thought about for the project?

If you working with a partner, when will you meet next to work on Part 0?

Name:

Section:

Email:

Are you planning to work with a partner for the project? If yes, do you have a partner already? Who is it?

What are some ideas you have thought about for the project?

If you working with a partner, when will you meet next to work on Part 0?

Section 7: Monte Carlo Simulations

Pass Out:

- **Worksheet**
- **Attendance Question**

Agenda

- **5 min** explain the problem. You might want to draw a circle and square on the board and explain how to calculate pi if you had the area of both.
- **5 min** let students work on the Step 1 in pairs or by themselves.
- **5 -7 min** go over Step 1 writing function definitions and basic doc strings on laptop (the answer does not need to be the one given but should still work).
- **10 min** let the students work on Step 2 in pairs or by themselves. Walk around and answer questions.
- **10 min** go over Step 2 writing up the function bodies in a file.
- **5 min** let students work on Step 3 in pairs or by themselves.
- **5-7 min** go over the solution to Step 3 in your file.
    - Review the if __name__ == '__main__': syntax
- **5 min** run the program
    - It might take a while to calculate the last couple values. Use this time to ask if there are any questions about HW5/project