
MATH 20 FALL 14 ASSIGNMENT 2

Do each of the following exercises. Hand in the solutions to the book problems on paper at the beginning of class on Friday, 9/26. Send the code for 2 and 3 by email by the same time. Section 4 contains some Python information that may turn out useful.

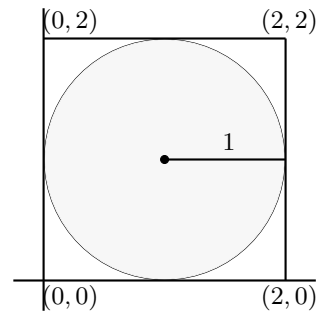
1 Book problems

Section 1.2: Problems 6, 14, 22, 23

Section 2.2: Problems 1, 2, 3, 7

2 Monte Carlo Sampling: estimating π

Write a program that will estimate π using the Monte Carlo Method on a unit circle inscribed in the square as in the illustration on the right. Your code should reliably return an answer such that the first 2 decimal places are correct (i.e. the answer is a number of the form 3.14...).



3 More about tossing coins

The following code tests something about a series of 10 tosses of a coin. What is it?

```
import random
X=0.0
for i in range(100000):
    A=0.0
    for j in range(10):
        B=random.randrange(2)
        if B==0:
            A=A+1
    if A==4:
        X=X+1
print X/100000
```

Write code that prints a list of length 11, where k th entry is an estimate for the probability that out of 10 tosses of a coin exactly k will come up heads. You can use the above code to help you.

4 (This is not an exercise) Some useful Python functions

Notice that the program in question 3 uses a double equality sign "`==`" to test if two data values are equal. That's because a single equality sign would simply change a value - to determine whether two things are already equal, we use a double.

Two new functions of the *random* package that can turn out useful are as follows:

`random.uniform(0,1)` draws a rational number between 0 and 1.

`random.randint(1,10)` draws an integer from 1 to 10, including 1 and 10.

You might notice that `randint` could have been used last week to simulate rolling dice just as well as `randrange`.

To test a point, you will need an `if` statement. That's a statement that executes instructions if a condition is satisfied. For example, the following code draws a natural number lower than 8, and tests whether it's bigger than 4:

```
import random
A=random.randrange(8)
if A>4:
    print "yes"
```

This you will probably not need for this assignment but it's a natural extension of an **if** statement: an **if/else** statement can include instructions what to do if the condition is NOT satisfied, and adding a statement **elif**, which means "else, if" allows for as many nested conditions as we want, here's an example:

```
import random
A=random.randrange(8)
if A>6:
    print "It's bigger than 6"
elif A>5:
    print "It's 6."
elif A>4:
    print "It's 5"
else:
    print "It's less than 5."
```