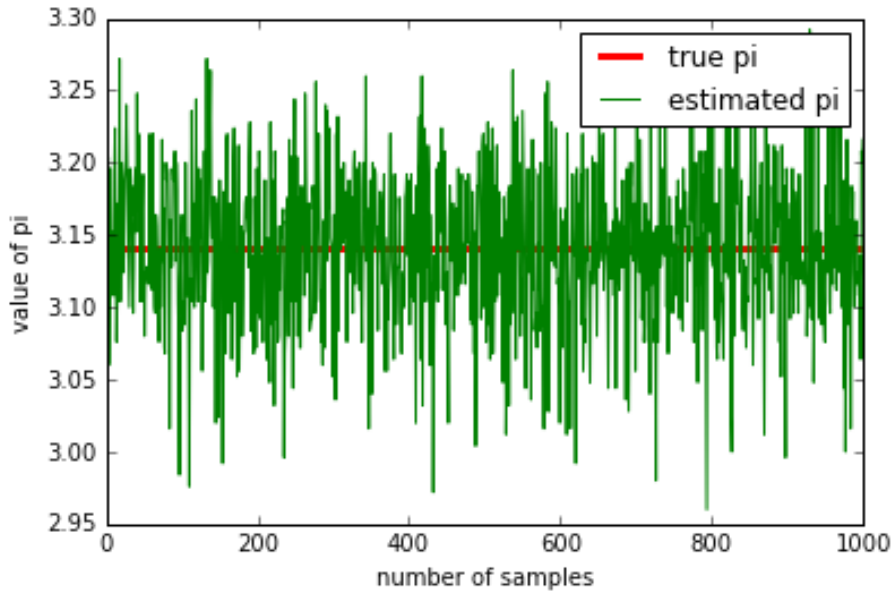
Assignment 2

```
import random
import numpy as np
import math
import matplotlib.pyplot as plt
%matplotlib inline
def pi_calc(n):
      radius = 1
      length = float(radius * 2)
      holder_list = []
      #Assuming length of rectangle is 2. The center of circle is at
(1,1)
      # The vertex of rectangles are at (1,1), (1,-1), (-1,1), (-1,-1)
      for i in range (1000):
      land_in_count = 0
      for i in range(n):
            x = random.uniform(-1,1)
            y = random.uniform(-1,1)
            if (x**2 + y**2) <= 1.0:
                  land_in_count += 1
      rectangle_area = length ** 2
      #Circle area would be 'pi'
      probab_land_in = land_in_count / float(n)
      pi = probab_land_in * rectangle_area
      holder_list.append(pi)
      plt.plot([math.pi for i in range(n)], color = "red",linewidth=3,
label = "true pi")
      plt.plot(holder_list, color = "green", label = "estimated pi")
      plt.legend()
      plt.xlabel("number of samples")
      plt.ylabel("value of pi")
      plt.show()

pi_calc(1000)
```



1. Error bars as a function of N
    a. The error bars in big Oh notation are O(1/sqrt(N)). Looking at the standard error formula, we get SE = (height x width x sqrt((ratio(1-ratio))/N))/radiusˆ2. So, we see that the error reduces as 2 sqrt(N) increases.
2. Instead of putting the circle in any-sized rectangle, specifically put the circle in a square and make sure that the diameter of the circle = width of the square. If we do this, we only need ONE input i.e. the diameter = width, and we can find pi more easily because the area ratio we obtain will simply = (pi x (diameter/2)2)/(diameter2) = ((pi x diameter2)/4)/(diameter2) = (pi x diameterˆ2)/(4 x diameterˆ2) = pi/4. So, pi = 4 x ratio. We will not have to deal with the different heights and widths.
3. You cannot keep the estimate accurate within a given number of samples. This is because you randomly throw darts at the dartboard-- whether or not your ratio of area of circle to area of rectangle is accurate depends totally on chance. You CAN however, repeat the simulation multiple times and obtain a mean-- which will almost definitely be more accurate than single samples