Challenge #1: Use of Monte Carlo

Master in Informatics and Computing Engineering
Programming Fundamentals
Instance: 2018/2019

0. Introduction

The challenge is extra work for more advanced Python programmers that easily solve the regular exercises.

*The **goal** of this challenge is to calculate the value of π using the Monte Carlo method.*

The text is taken from the book (page 200): Guttag, John. Introduction to Computation and Programming Using Python: With Application to Understanding Data, Second Edition. MIT Press, 2016. ISBN: 9780262529624

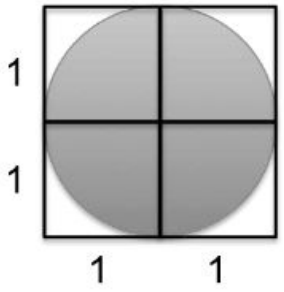
TIP: Do not see the solution before trying.

1. Monte Carlo

It is easy to see how Monte Carlo simulation is useful for tackling problems in which nondeterminism plays a role. Interestingly, however, Monte Carlo simulation (and randomized algorithms in general) can be used to solve problems that are not inherently stochastic, i.e., for which there is no uncertainty about outcomes.

2. Calculate π

Long before computers were invented, the French mathematicians Buffon (1707-1788) and Laplace (1749-1827) proposed using a stochastic simulation to estimate the value of π. Think about inscribing a circle in a square with sides of length 2, so that the radius, r, of the circle is of length 1 .



By the definition of π, area = π r^2 . Since r is 1 , π = area .

But what's the area of the circle?

Buffon suggested that he could estimate the area of a circle by a dropping a large number of needles (which he argued would follow a random path as they fell) in the vicinity of the square. The ratio of the number of needles with tips lying within the square to the number of needles with tips lying within the circle could then be used to estimate the area of the circle.

If the locations of the needles are truly random, we know that:

$$\frac{needles\ in\ circle}{needles\ in\ square} = \frac{area\ of\ circle}{area\ of\ square}$$

solving for the area of the circle,

$$\textit{area of the circle} = \frac{area\ of\ square * needles\ in\ circle}{needles\ in\ square}$$

3. Let's do it in Python.

Start by throwing 1000 needles for each estimation of π. Get 100 estimates of π, use the average, and double the number of needles used until the standard deviation (`statistics.stdev`) of the 100 estimates is less than 0.005 (as in the book).

The end.

FPRO, 2018/19