Assigned: 21 August

Homework #1

EE 541: Fall 2023

Due: Thursday, 31 August at 23:59. Submission instructions will follow separately on canvas.

Use only Python standard library modules (https://docs.python.org/3/library/) and `matplotlib` for this assignment, i.e. do not `import numpy`, `scikit`, or any other non-standard package.

1. Simulate tossing a biased coin (a Bernoulli trial) where $P[\text{HEAD}] = 0.70$.

   (a) Count the number of heads in 50 trials. Record the longest run of heads.

   (b) Repeat the 50-flip experiment 20, 100, 200, and 1000 times. Use matplotlib to generate a histogram showing the observed number of heads for each case. Comment on the limit of the histogram.

   (c) Simulate tossing the coin 500 times. Generate a histogram showing the heads run lengths.

2. Define the random variable $N = \min\{n : \sum_{i=1}^{n} X_i > 4\}$ as the smallest number of standard uniform random samples whose sum exceeds four. Generate a histogram using 100, 1000, and 10000 realizations of N. Comment on the expected value $E[N]$.

3. The secant method is an iterative root-finding algorithm. It uses a sequence of secant line roots to approximate c such that $f(c) = 0$ for a continuous function f. Unlike Newton's method it does not require knowledge or evaluation of the derivative f'. The secant method is defined by the recurrence:

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

   Write a python script that uses the secant method to approximate the root of a continuous function f in the interval $[a, b]$. You may assume that f has at most one root in $[a, b]$. Use $|x_{k+1} - x_k| < 10^{-10}$ as the convergence criterion. Let N be the number of iterations to reach convergence. Output N followed by the three root approximations x_{N-2}, x_{N-1}, x_N. **Output each number to its own line** and use precision sufficient to show convergence.

   Import the function f from a file named `func.py` in the same directory as your script — *i.e.,* `from func import f`. You may assume that f is continuous on $[a, b]$ and that `func.f(x)` returns a scalar float for all $x \in [a, b]$.

   Your script should accept `a` and `b` as two numeric command line arguments, *i.e.,* `python hw3p1.py "1.1" "1.4"`. Your script must validate that `a` and `b` are numeric, verify that `a < b`, and check that `f(a)f(b) < 0` — see Bolzano's Theorem. Write "`Range error`" to `STDERR` (standard error) if any of these three conditions fail and immediately terminate.

   Your script should not produce any output except as described above.