Ahmad Hijazi
Monte-Carlo

The goal of this homework was to use Monte Carlo sampling to estimate π. The first step was to generate random points and check how many fell inside the unit circle. This gave me some trouble at first, since generating random points and plotting them individually via a for-loop is very inefficient. Using NumPy vectorization, as a classmate recommended, made this exponentially faster (15 seconds for one result to instant results).

After plotting the results with Matplotlib and comparing the estimates based on different values of n, we can notice that the estimates of π got closer to the true value as the number of points increased. Repeating the experiment 500 times for a fixed number of points, the distribution of the estimates formed a clear Gaussian curve centered around π, just as the Central Limit Theorem predicts. Pretty satisfying to see a normal distribution simply from random sampling.

Even with only one thousand points, the estimate was surprisingly close—I never would have guessed we could get that close with such few points. Increasing the number of points shrank the spread, and the estimates lined up almost perfectly with pi. Overall, I learned a lot, and it was cool to see convergence, random sampling, and Gaussian distributions with code I wrote.