N.A. Fall 2021 Homework 1 Solutions

The following are partial solutions for homework 1, which should give you an idea of whether or not you were on the right track. Code listings are not included, but you should feel free to ask your TA or professor about them.

2) The expected value of a six-sided die is 3.5. The variance is

$$\text{var}(k) = \mathbb{E}[k^2] - (\mathbb{E}[k])^2 = \frac{6(6+1)(2\cdot 6+1)}{6\cdot 6} - 3.5^2 = \frac{35}{12}. \tag{1}$$

For the second part of the problem, your histogram should show that A_{1000} is approximately normally distributed. We also know that $\mathbb{E}[A_{1000}] = \mathbb{E}[k] = 3.5$, that $\text{var}(A_{1000}) = \text{var}(k)/1000 = \frac{35}{12000}$, and that $\text{std}(A_{1000}) = \sqrt{\text{var}(A_{1000})} = \sqrt{\frac{35}{12000}} \approx 0.054$.

4) If you do not change your guess, then the probability of winning is $\frac{3}{7}$, since that is the probability of their being a prize behind one of your three doors. If you do change, then the probability of winning is $\frac{4}{7}$, since you will win if and only if the prize was not behind one of your three original doors.

This problem is a little tricky to simulate because of the nature of the random numbers that you need to generate. To calculate these probabilities numerically, you can generate a random prize door by generating a random integer between 1 and 7 using the command random.randint(1,7) (requires the random package). You can generate three distinct door choices by choosing the first three entries of a random permutation of the numbers 1 to 7. You can generate the opened doors by forming a list of the doors without a prize that you have not already chosen, then choosing the first three entries in a random perturbation. You can then use these probabilities, and many trials, to estimate your probability of winning in each situation.

8) Suppose that Y is a random variable uniformly distributed between 0 and 1, then the random variable $X = Y^2$ has the desired cumulative distribution. To see this, we can compute

$$\text{Prob}(X \le a) = \text{Prob}(Y \le \sqrt{a}) = \sqrt{a}. \tag{2}$$

It is important to note that this idea is applicable to many other desired distributions. If we suppose that $g(x)$ is some monotone functions such that we can analytically compute it's inverse, then the random variable $Z = g^{-1}(Y)$ has the following cumulative distribution:

$$\text{Prob}(Z \le a) = \text{Prob}(g^{-1}(Y) \le a) = \text{Prob}(Y \le g(a)) = g(a) \tag{3}$$

12a) The integral can be factored, making it easy to evaluate:

$$\int_0^2 \int_0^2 \dots \int_0^2 x_1 x_2 \dots x_{10} dx_1 dx_2 \dots dx_{10} = \left(\int_0^2 x dx\right)^{10} = 2^{10} = 1024. \tag{4}$$

b) You can write a code that generates a random point $\boldsymbol{x} \in [0,2]^{10}$, multiplies the components, and then adds it to the total. Adding up many of these samples, dividing by the number of samples, and multiplying by the size of the domain, will give you the desired Monte Carlo estimator.

c) The central limit theorem tells us that the estimator should be approximately normally distributed. For a normally distributed random variable X with standard deviation σ, we have that

$$\mathbb{E}[|X - \mathbb{E}[X]|] = \frac{\sigma}{\sqrt{\pi}}. \tag{5}$$

Thus on average, the estimator should be about $\frac{1}{\sqrt{\pi}} \approx 0.56$ standard deviations from the true answer, no matter the size of N. The actual numbers that you get, will of course vary around this.