Assignment #6 (Due April 16)

1. (**Computing Problem:** Stochastic root-finding in drug design)
   a) Suppose that we are designing an over-the-counter drug, and are trying to decide the dosage. For an actual drug amount θ, the "effective dosage" is uncertain, but is modeled as θX, where the random variable X is uniformly distributed on the interval [1, 5] and represents uncertain biological factors (not affected by the drug) that vary from person to person. Suppose that the final drug concentration in the liver of a patient increases exponentially with effective dosage, and is given as $g(X,\theta) = e^{\theta X}$. Write a Python program that, with probability 95%, will estimate to within $\pm 1\%$ the value $\bar{\theta}$ that achieves an expected liver concentration equal to 20. You can use, e.g., scipy.optimize.brentq for rootfinding.
   b) A different drug will decrease user discomfort more and more up to a certain level but, beyond that level, discomfort will start to increase again. The model for the effect of dosage on discomfort is $g(X,\theta) = e^{-\theta X} + (\theta X / 2)$, where again the random variable X is uniformly distributed on the interval [1, 5] and represents uncertain biological factors. Write a Python program that will, with probability 95%, estimate to within $\pm 1\%$ the value $\bar{\theta}$ that minimizes the expected discomfort level. [Hint: for this problem, the derivative with respect to θ of $E[g(X,\theta)]$ can be brought inside the expectation operator. You should therefore be able to use the program from Part (a) with relatively little modification. You can also use scipy.optimize.minimize_scalar to compute both θ_n and g_n^{opt} (as in Part (c) below) at the same time.]
   c) (Extra credit) In Part (b), denote by g_n^{opt} the estimated minimum discomfort corresponding to θ_n. It is likely that your value of g_n^{opt} is less than the true minimum discomfort, which we denote by $g^{\text{opt}} \approx 0.8633$. (This toy problem is simple enough so that, with some work, you can compute the true optimal solution.) Prove that, in fact, for any choice of *g* and *X*, we have $E[g_n^{\text{opt}}] \le g^{\text{opt}}$, so that on average you always think that you have a better solution than you actually do; this bias vanishes as *n* becomes large. [Hint: First observe that
   $$g_n^{\text{opt}} = \min_\theta \frac{1}{n}\sum_{i=1}^n g(X_i,\theta) \le \frac{1}{n}\sum_{i=1}^n g(X_i,\theta^*)$$
   for any given value θ^*.]

2. Consider a simulation of a database system. Denote by *U* the average processing time for a database query over the course of a day, and by *V* the fraction (in %) of database queries during the day that access a particular type of data (say, image data). Suppose that we obtain the following 10 samples of *U* and *V*, based on 10 i.i.d. simulation replications.

Run #:	1	2	3	4	5	6	7	8	9	10
U:	12	18	15	35	8	16	2	12	20	6
V:	20	38	29	55	12	36	10	47	40	10

Compute a 95% confidence interval for the squared coefficient of correlation between *U* and *V*, i.e.,

$$\mathrm{Corr}[U,V]^2 = \frac{\mathrm{Cov}[U,V]^2}{\mathrm{Var}[U]\mathrm{Var}[V]}$$

a) Using the Taylor-Series method (i.e., the Delta method)

b) Using the jackknife method [Feel free to use a spreadsheet to manage the calculations]

[Hint: write $\mathrm{Corr}[U,V]^2 = g(\mu_1,\mu_2,\mu_3,\mu_4,\mu_5)$ for an appropriate function g where $\mu_1 = E[U]$, $\mu_2 = E[V]$, $\mu_3 = E[U^2]$, $\mu_4 = E[V^2]$, and $\mu_5 = E[UV]$. When computing quantities such as $\partial g(x_1,x_2,x_3,x_4,x_5)/\partial x_i$ for $i = 1,2,3,4,5$, it might be helpful to write $g(x_1,x_2,x_3,x_4,x_5) = a^2/(bc)$, where each of a, b, and c is a function of $x_1,x_2,\ldots,x_5$. Compute the values of a, b, and c and then use the chain rule of differentiation to write each partial derivative as a function both of a, b, and c and of the derivatives of a, b, and c.]

3. (Multiple performance measures) Suppose that you are simultaneously interested in $k > 1$ different performance measures for the system of interest (as is common in practice), and that, for each of $n > 0$ i.i.d. simulation replications you record each of the k measures. For each measure, you construct a $100(1-\alpha)\%$ confidence interval. Since the different performance measures were computed from the same set of replications, the interval estimates are statistically dependent
.
   a) What is the expected number $E[N]$ of confidence intervals that will *not* bracket the true value?
   b) Using Bonferroni's inequality, give a simple procedure for determining a value α^* such that if you individually compute a $100(1-\alpha^*)\%$ confidence interval for each of the k performance measures, then the probability that all k intervals simultaneously bracket their true values is at least $100(1-\alpha)\%$.

   [Hint: consider events of the form A_i = "i^{th} confidence interval brackets the true value of the i^{th} performance measure", as well as the indicator functions for such events.]

4. Suppose that $(X(t): t \geq 0)$ is a GSMP having regeneration points $T_0 = 0, T_1, T_2, \ldots$ and that rewards accrue continuously at rate $q(s)$ whenever the current state is $s \in S$. Also suppose that the performance measure of interest is the β-discounted reward r, defined as

$$r = E\left[\int_0^\infty e^{-\beta u} q\big(X(u)\big)\,du\right].$$

   a) Re-express the reward r in the form $r = E[X]/E[Y]$, where each of X and Y is a random variable that is defined in terms of the first regenerative cycle of the GSMP (just like the random variables Y_1 and τ_1 in the standard regenerative method). [Hint: write

$$\int_0^\infty e^{-\beta u} q\big(X(u)\big)\,du = \int_0^{T_1} e^{-\beta u} q\big(X(u)\big)\,du + e^{-\beta T_1}\int_{T_1}^\infty e^{-\beta(u-T_1)} q\big(X(u)\big)\,du$$

   and note that $e^{-\beta T_1}$ is independent of $\int_{T_1}^\infty e^{-\beta(u-T_1)} q\big(X(u)\big)\,du$ by the regenerative property.]

b) Using the above representation, we can apply the regenerative method to estimate r, based on n cycles. In the i^{th} cycle (where $1 \le i \le n$), we obtain a pair of observations (X_i, Y_i). Give explicit formulas for X_i and Y_i. (In the first part of the problem, you basically gave a formula for the first cycle; the second part of the problem is asking for the general formula.)