Mathematical modeling.

Assignment 6, due March 8.

1. There are two kinds of light bulbs, normal and long life. We cannot tell them apart. However, the probability, p, of getting a long life bulb is very small. Let T be the failure time of the bulb. The probability density for T is $f(t) = \lambda e^{-\lambda t}$. Of course, this formula only holds if $t > 0$. It is impossible for T to be negative. The bulbs are distinguished by their failure rates. A long life bulb characterized by it's failure rate, λ_l, which is much smaller than the failure rate, λ_n, for normal bulbs. A particular randomly chosen bulb fails at time T. Compute the probability that bulb was a long life bulb, as a function of T. Question: about how long does the bolb have to last before you start believing that it is a long life bulb?

2. In a quiz show, we start with k contestants. In each round, the host asks each contestant a question. The contestant answers correctly with probability p. All questions and contestants are independent of each other. A contestant is eliminated as soon as he or she answers incorrectly. The rounds continue until the last contestant is eliminated. What is the probability that the last contestant is eliminated in round n?

3. We have n parts that may be broken. Each part is broken with probability p and all are independent. The probability that a part works is $q = 1 - p$. The number of broken parts is N. The probability that k of the parts are broken is

$$f(k) = \Pr(N = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k q^{n-k} ,$$

where

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$$

is the binomial coefficient, called "n choose k". The "normal approximation" to $f(k)$ is given by

$$f(k) \approx g(k) = \frac{1}{\sqrt{2\pi npq}} e^{-(k-np)^2/2npq} .$$

This is supposed to be a good approximation when np and nq are both large. Use Matlab[1] to test this by plotting on the same graph f and g for various values of p and n. Type "`help nchoosek`" to save some time. If you take n too large, Matlab will be unable to compute n choose k if n is too big, as you will discover. You compute p^k in Matlab by typing "`p^k`". Experiment with a large number of p and n values but only hand in a few graphs.

---

[1] You may use another system, but then you're on your own in figuring out how to get n choose k.