STAT 740 - Spring 2004 - Take Home 2

Due: Noon Wednesday May 5th

- Do not consult with anyone except me on these problems.
- Adequate comments and a modicum of efficiency are required for a perfect score.
- E-mail the code that is needed so that I can **exactly** reproduce your results.

Answer three of the following four questions:

1) Consider using the bisect function at `http://www.stat.sc.edu/~habing/courses/740/solve.txt` to find the maximum of a likelihood function by finding the zeros of the derivative of the log-likelihood.

a) Name two advantages that the bisection method has over Newton's method, and name one disadvantage.

b) Why is it if there is only one local extrema $\theta_{extreme}$ between the values a and b that the θ_{est} solved for by the bisect routine might not satisfy $|\theta_{extreme} - \theta_{est}| <$ `tol` ?

c) Modify the bisect code so that it runs until at least one of three stopping rules is met: the current stopping rule, a rule concerning $|\theta_{extreme} - \theta_{est}|$, and a maximum number of iterations. Make it so that the procedure outputs appropriate values to see how close it was to meeting the three stopping rules.

d) Consider a case where the likelihood function is similar to the posterior distribution we plotted in `http://www.stat.sc.edu/~habing/courses/740/methast.txt`. Even if the algorithm in (b) converged by all three criteria, why might we not have found the maximum likelihood estimate of θ? (The same problem can occur using Newton's method or the secant method as well.)

e) Suggest a way in which we could safeguard against the problem in c.

2) R has many built in functions that are fairly general. For example both `optim` and `adapt` will work with arbitrary functions. These functions should be expected to work fairly quickly since they are actually programmed in C or Fortran and called by R. However, their generality can slow them down sometimes. For example, the programming in C or Fortran needs to be general enough to work with the arbitrary functions and often has many safeguards that aren't needed in a particular circumstance.

The code at `http://www.stat.sc.edu/~habing/courses/740/finalint.txt` demonstrates using both a manually coded routine (the function `intfun`) and the built in integration (the function `intfun2`) to estimate the probability $P[X_1<a, X_2<b]$ for a bivariate normal distribution with mean=(0,0), variance=(1,1), and correlation=ρ.

Convert the `intfun` program to a Fortran subroutine and compile it as a `dll` file that can be called from R. Write an intfun3 function that calls this `dll` and briefly compare its speed to `intfun1` and `intfun2`.

3) An experiment is designed to measure the lifetimes of a new product. Each of the products was put into use at the same time, but the monitoring began and ended at different times. Say the lifetimes are distributed according to an exponential distribution with mean $1/\theta$, where the monitoring of product i began at time s_i and ended at time t_i. Say k of the products failed before their first observation time s_i, we have the exact failure time for m of the products, and n of the productss failed after their last observation time t_i. Construct the formulas for performing the E-M algorithm to estimate θ, code the algorithm in R, and use it to estimate θ for the 20 observations given below.

< 1.42	< 1.89	< 1.88				
1.26	1.76	2.01	2.01	2.42	3.49	3.89
4.16	4.33	4.82	6.67	7.00	8.50	8.52
> 5.36	> 5.83	>12.54				

Hint: If X is exponential with mean $1/\theta$, then

$$E[X \mid X \le s] = \frac{\frac{1}{\theta} - s(\exp(-\theta s)) - \left(\frac{1}{\theta}\right)\exp(-\theta s)}{1 - \exp(-\theta s)}$$