Survival Data Analysis (BIOS:7210)
Breheny

Assignment 8
Due: Thursday, November 7

1. Implement a Bayesian Weibull regression model and apply it to the `pbc` data. Include treatment, stage, and hepatomegaly as predictors. Give $\boldsymbol{\beta}$ and γ reference priors as in the piecewise exponential model we used in class. Some JAGS syntax notes:

   - Unfortunately, JAGS parameterizes the Weibull distribution differently than we did in class. The simplest fix is to use the generalized gamma instead (the Weibull is a special case of the generalized gamma):

     `t[i] ~ dgen.gamma(1, rate, gamma)`

     means that T follows a Weibull distribution with rate λ and shape γ, where λ and γ follow our lecture definition and derivations.
   - `a <- x` means a deterministic assignment, while `a ~ dnorm` specifies a random distribution.
   - The normal distribution is parameterized in terms of a mean and *precision* $(1/\sigma^2)$. So, `a ~ dnorm(0, 10)` means that `a` follows a normal distribution with mean 0 and variance 0.1.
   - JAGS uses the same syntax for matrix multiplication as R: `eta <- X %*% beta`

   (a) Write a "methods" paragraph describing the model you fit and a "results" paragraph describing what the model tells you about the effects of treatment, stage, and hepatomegaly on survival. The results paragraph should be accompanied by a table or figure – **NOT** raw R output – that provides posterior means and 95% intervals for the relevant quantities. For the sake of this problem, use the AFT interpretation of the regression coefficients.

   (b) Suppose a physician is considering treating a stage 3, enlarged liver patient with D-penicillamine. Based on the above model, what is the likely impact on that patient's median survival time in terms of absolute number of years (i.e., "increase by 2 years", not "increase by 14%"). Provide an estimate and 95% posterior interval. Quietly ponder whether this providing the equivalent 95% frequentist confidence intervals would be easy to do or not.

2. Take the crude `R` code in http://myweb.uiowa.edu/pbreheny/7210/f19/notes/10-29.R (this is the same as the code we discussed in class) and improve it in three ways: (1) The code should not assume a fixed number of iterations; rather, it should check for convergence after every update. (2) The code should check whether the Newton update does, in fact, increase the Cox partial log-likelihood, and if it does not (and only then!), apply step-halving. (3) The code should be contained within a function, `cox`, that can be called like this: `b <- cox(X, d)`, where `X` is the design matrix, `d` is the vector of failure indicators, and `b` is a vector of estimated coefficients. You may assume that `X` and `d` are sorted by failure time and that there are no ties present.

   This problem does not require any written component. Rather, submit the function as a separate file on Dropbox; the file should contain the function definition and nothing else. Your score will be determined by whether your code executes correctly on a data set of my choosing. If your function does not run because you decided to change the name of the function or arguments, you will lose points.