Survival Data Analysis (BIOS 7210)
Breheny

Assignment 9
Due: Thursday, November 5

1. Take the crude `R` code in http://myweb.uiowa.edu/pbreheny/7210/f15/notes/10-27.R (this is the same as the code we discussed in class) and improve it in two 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-Raphson update does, in fact, increase the Cox partial log-likelihood, and if it does not (and only then!), apply step-halving.

   This problem does not require any written component. Rather, e-mail your code to me; your score will be determined by whether your code executes correctly on a data set of my choosing. As in the `R` code referenced above, assume that `X` is the design matrix and `d` is the vector of failure indicators, with `X` and `d` sorted by failure time with no ties present. Your code should not modify `X` or `d` in any way. Also, please do not send me 400 lines of code and require me to find the 20 lines of code that actually implement your algorithm.

2. This question concerns Wald and likelihood ratio tests for the global null hypothesis $\boldsymbol{\beta} = \mathbf{0}$ in the Cox model. Fit the following model to the GVHD data:

   ```
   fit <- coxph(Surv(Time, Status) ~ Group + LAF + Age, gvhd)
   ```

   (a) In terms of the MLE $\widehat{\boldsymbol{\beta}}$ and the information, $\mathbf{I}$, of the fitted model, provide a mathematical expression for the global Wald test statistic, and say what distribution it follows under the null.

   (b) Using `coef(fit)` and `vcov(fit)`, calculate the global Wald test statistic and verify that it equals the value in the output of `summary(fit)`.

   (c) Under the assumption of no ties among times on study, provide a mathematical expression for the log-likelihood of the null Cox model. The expression should be in terms of $\{d_i\}$ and $\{r_i\}$, the failure time rank for the subject i, only (i.e., it should not contain π or w terms; summations are allowed, however).

   (d) Using `Status`, `rank(Time)`, and `logLik(fit)`, calculate the global likelihood ratio test statistic and verify that it equals the value in the output of `summary(fit)`.