KEVLAR RANDOM EFFECTS INFORMATIVE LOGNORMAL PRIOR I

```
#############################################################################################
###############    This program represents the collaborative efforts of Avery Ashby,       ###############
###############     Ramon Leon, and Jayanth Thyagarajan, This is an inclusive odc file     ###############
###############         containing the model, data, and initial values. March 20, 2002.    ###############
#############################################################################################

#############################################################################################
###############                                  MODEL                                      ###############
#############################################################################################

model KevlarRandomEffectsLognormalPrior1to5;
{

###############                      Generation of prior values                             ###############

taub ~ dgamma(0.001,0.001)                                    # Precision parameter of normal random effect
                                                              #     (vague)
sigmasquare <- 1 /  taub                                      # Rescale into variance parameterization
intercept ~ dnorm(0,0.001)                                    # Intercept component  (vague)
beta.stress ~ dnorm(0,0.001)                                  # Fixed stress effect  (vague)

    ###############  Creates a lognormal prior for the shape parameter of the Weibull  ###############
    ###############           where most certainly r (Beta) is between 1 and 5          ###############

anot <- -0.80472
mu <- anot * -1
bnot <- 0.31241
tau <- pow(bnot,-2)                                           # tau = 1/(bnot^2)
r ~ dlnorm(mu,tau)                                            # Weibull shape parameter
#############################################################################################

###############          This loop creates the random effect for the spools              ###############

for(i in 1:N ) {                                              # N is the number of spools (8)
  b[i] ~ dnorm(0,taub)                                        # random effect of spool
  }
###############                          End of loop                                        ###############
#############################################################################################

###############   This loop reads in the data and calculates Weibull scale parameter     ###############

for(j in 1:M ) {                                              # M is the number of rows in the data (108)
  log(eta[j]) <- intercept + beta.stress * log(stress[j]) + b[spool[j]] # This is the function for mu in the Weibull
  lambda[j] <- pow(eta[j],-r)                                 # Rescale into lambda parameterization
  }                                                           #         for use in winBUGS 1.4
###############                          End of loop                                        ###############
#############################################################################################

###############   This loop gives the likelihood for  failure times as exact or censored    ###############

for(j in 1:M ) {
  t[j] ~ dweib(r,lambda[j])I(cen[j],)                         # failure times are Weibull or censored
  }
###############                          End of loop                                        ###############
#############################################################################################
```

```
################       Calculates specified failure quantiles and prediction intervals       ################

for(i in 1:N) {
   eta234[i] <- exp(intercept + beta.stress * log(23.4) + b[i])      # eta values at 23.4 MPa stress for each spool
   quan234[i] <- eta234[i] * pow((-log(1 - 0.01)),(1/r))             # 1st percentile at 23.4 MPa stress for each spool
   lambda234[i] <- pow(eta234[i],-r)                                 # Rescale into lambda parameterization
   y.234new[i] ~ dweib(r,lambda234[i])                               # Predicted distribution for 23.4 MPa stress
                                                                     #     for each spool
   probability234[i] <- 1 - exp(-(pow((1000/eta234[i]),r)))          # Prob of failure at 1000 hours for each spool

   eta225[i] <- exp(intercept + beta.stress * log(22.5) + b[i])      # eta values at 22.5 MPa stress for each spool
   quan225[i] <- (eta225[i] * pow((-log(1 - 0.5)),(1/r))) / 1000     # 50th percentile at 22.5 MPa stress for each spool
   lambda225[i] <- pow(eta225[i],-r)                                 # Rescale into lambda parameterization
   y.225new[i] ~ dweib(r,lambda225[i])                               # Predicted distribution for 22.5 MPa stress
                                                                     #     for each spool
   probability225[i] <- 1 - exp(-(pow((1000/eta225[i]),r)))          # Prob of failure at 1000 hours for each spool
   }
################                                End of loop                                ################
#########################################################################################################

################   Represents a new observation from a random spool in the population  ################

other ~ dnorm(0,taub)                                                # Choose a spool at random from population
eta234other <- exp(intercept + beta.stress * log(23.4) + other)      # eta values at 23.4 stress for a new spool
quan234other <- eta234other * pow((-log(1 - 0.01)),(1/r))            # 1st percentile at 23.4 stress for a new spool
lambda234other <- pow(eta234other,-r)                                # Rescale into lambda parameterization
y.234newother ~ dweib(r,lambda234other)                              # Predicted distribution for 23.4 MPa stress
probability234other <- 1 - exp(-(pow((1000/eta234other),r)))         # Probability of failure at 1000 hours for a new obs
                                                                     #         at 23.4 MPa

eta225other <- exp(intercept + beta.stress * log(22.5) + other)      # eta values at 22.5 stress for a new spool
quan225other <- (eta225other * pow((-log(1 - 0.5)),(1/r))) / 1000    # 50th percentile at 22.5 MPa stress for new spool
lambda225other <- pow(eta225other,-r)                                # Rescale into lambda parameterization
y.225newother ~ dweib(r,lambda225other)                              # Predicted distribution for 22.5 MPa stress
probability225other <- 1 - exp(-(pow((1000/eta225other),r)))         # Probability of failure at 1000 hours for a new obs
                                                                     #         at 22.5 MPa
#########################################################################################################


 }
################                                End of program model                                ################
#########################################################################################################

#########################################################################################################
################                                DATA                                ################
#########################################################################################################

list(N=8, M=108,
    stress = c(29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,
               29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,
               29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,29.7,
               29.7,29.7,29.7,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,
               27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,27.6,
               27.6,27.6,27.6,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,
               25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,
               25.5,25.5,25.5,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,
```

```
            23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4),

    spool  = c(2,7,7,7,7,6,7,5,2,2,2,3,5,7,3,6,3,2,7,2,7,5,8,3,2,6,2,5,4,1,8,8,1,1,1,4,
            4,4,4,3,3,3,2,3,2,2,2,1,2,6,6,2,8,1,2,4,1,6,4,4,1,8,4,6,7,3,2,2,2,2,3,8,
            2,8,2,6,8,8,6,8,8,1,5,4,1,4,1,7,7,6,3,5,2,6,5,6,5,1,1,1,1,4,4,4,4,8,8,8),

    cen = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
            0,0,0,0,0,0,0,0,0,0,0,0,41000,41000,41000,
            41000,41000,41000,41000,41000,41000,41000,41000),

    t = c(2.2,4,4,4.6,6.1,6.7,7.9,8.3,8.5,9.1,10.2,12.5,13.3,14,14.6,15,
            18.7,22.1,45.9,55.4,61.2,87.5,98.2,101,111.4,144,158.7,243.9,
            254.1,444.4,590.4,638.2,755.2,952.2,1108.2,1148.5,1569.3,
            1750.6,1802.1,19.1,24.3,69.8,71.2,136,199.1,403.7,432.2,453.4,
            514.1,514.2,541.6,544.9,554.2,664.5,694.1,876.7,930.4,1254.9,
            1275.6,1536.8,1755.5,2046.2,6177.5,225.2,503.6,1087.7,1134.3,
            1824.3,1920.1,2383,2442.5,2974.6,3708.9,4908.9,5556,6271.1,
            7332,7918.7,7996,9240.3,9973,11487.3,11727.1,13501.3,14032,
            29808,31008,4000,5376,7320,8616,9120,14400,16104,20231,
            20233,35880,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA))

################                End of data                ################
#############################################################################################

#############################################################################################
################                INITIAL VALUES                ################
#############################################################################################

list(intercept = 84.1, beta.stress = -23.1,r = 1.21, taub = 1)        # initial values based on Crowder et al. (1991)
list(intercept = 1, beta.stress = -1,r = 1, taub = 1)
################                End of initial values                ################
#############################################################################################
```