KEVLAR NO SPOOLS LOGNORMAL PRIOR II

```
##########################################################################################
###############  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 KevlarNoSpoolsLognormalPrior1to2;
{

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

intercept ~ dnorm(0,0.001)                          # Intecept prior
beta.stress ~ dnorm(0,0.001)                        # Fixed stress effect

    ################ Creates a lognormal prior for the shape parameter of the Weibull ################
    ################       where most certainly (Beta) is between 1 and 2          ################

anot <- -0.3466
mu <- anot * -1
bnot <- 0.1345
tau <- pow(bnot,-2)                                 # tau = 1/(bnot^2)
r ~ dlnorm(mu,tau)                                  # Weibull shape parameter
##########################################################################################

###############  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])   # 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 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 for a given spool       ###############

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

eta225 <- exp(intercept + beta.stress * log(22.5))  # Eta values at 22.5 MPa stress
quan225 <- (eta225 * pow((-log(1 - 0.5)),(1/r))) / 1000   # 50th percentile at 22.5 MPa stress (/ 1000)
```

```
lambda225 <- pow(eta225,-r)                              # Rescale into lambda parameterization
y.225new ~ dweib(r,lambda225)                            # Predicted distribution for 22.5 MPa stress
probability225 <- 1 - exp(-(pow((1000/eta225),r)))       # Prob of failure at 1000 hours for each spool
##########################################################################################

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

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

list(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,
               0,0,0,0,0,0,0,0,0,0,0,0,41000,41000,41000,
               41000,41000,41000,41000,41000,41000,41000,41000),

    t = c(2.2,4,4,4.6,6.1,6.7,7,7.9,8.3,8.5,9.1,10.2,12.5,13.3,14,14.6,15,
               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)            # initial values based on Crowder et al. (1991)
list(intercept = 1, beta.stress = -1,r = 1)
################                    End of initial values                    ################
#########################################################################################################
```