

KEVLAR RANDOM EFFECTS GAMMA PRIOR

```
#####  
##### 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 KevlarRandomEffectsGammaPrior;  
{
```

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

```
taub ~ dgamma(0.001,0.001) # Precision parameter of normal random effect  
sigmasquare <- 1 / taub # Rescale into variance parameterization  
intercept ~ dnorm(0,0.001) # Intercept component  
beta.stress ~ dnorm(0,0.001) # Fixed stress effect  
r ~ dgamma(1,0.2) # Shape parameter of Weibull  
# inverse scale parameter that is used by  
# winBUGS parameterization for gamma
```

```
#####
```

```
##### 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 contains the likelihood failure times as exact or censored #####
```

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

```
##### End of loop #####
```

```
#####
```

```
##### Calculates failure quantiles and prediction intervals for a given spool #####
```

```
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 MPa stress for a new spool
quan234other <- eta234other * pow((-log(1 - 0.01)),(1/r)) # 1st percentile at 23.4 MPa 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 MPa 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,
                27.6,27.6,27.6,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,
                25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,25.5,
                25.5,25.5,25.5,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,23.4,
                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,
               4,4,4,3,3,3,2,3,2,2,1,2,6,6,2,8,1,2,4,1,6,4,4,1,8,4,6,7,3,2,2,2,2,3,8,

```

