library(rjags); setwd("C:/R files BHMRA")

options(scipen=999)

attach("DS\_7\_8.Rdata")

attach("DS\_7\_8\_DM.Rdata")

# Variance regression model

cat("model { for (i in 1:113) {

y[i] ~ dnorm(beta[1]+beta[2]\*x[i], exp(-gam[1]-gam[2]\*x[i]))

ynew[i] ~ dnorm(beta[1]+beta[2]\*x[i], exp(-gam[1]-gam[2]\*x[i]))

**# predictive discrepancy**

predF[i] <- pow(y[i]-ynew[i],2)}

**# priors**

for (j in 1:2) {gam[j] ~ dnorm(0,0.001)

beta[j] ~ dnorm(0,0.001)}

C.F <- sum(predF[])}

", file="assay1.jag")

**# Initial values and Estimation**

ini1 <- list(beta=c(0,0),gam=c(0,0))

ini2 <- list(beta=c(-20,20),gam=c(1,0))

ini <- list(ini1,ini2)

M1 <- jags.model(inits=ini,data= DS\_7\_8,n.chains=2, file="assay1.jag",n.adapt=500)

S1.1 <- coda.samples(M1,c("beta","gam"),n.iter=5000)

gelman.diag(S1.1)

S1.2 <- coda.samples(M1, c("beta","gam","C.F","ynew"),n.iter=5000)

summary(S1.2)

**# predictive variance**

M=as.matrix(S1.2)

M1=M[1:10000,6:118]

predvar=apply(M1,2,sd)^2

sum(predvar)

**# Variance power model**

cat("model { for (i in 1:113) {y[i] ~ dnorm(eta[i], 1/(s[i]\*s[i]))

ynew[i] ~ dnorm(eta[i], 1/(s[i]\*s[i]))

**# predictive discrepancy**

predF[i] <- pow(y[i]-ynew[i],2)

eta[i] <- beta[1]+beta[2]\*x[i]

s[i] <- sig\*pow(1+abs(eta[i]),lam)}

**# Priors**

sig ~ dunif(0,250)

lam ~ dunif(-2,2)

for (j in 1:2) {beta[j] ~ dnorm(0,0.001)}

C.F <- sum(predF[])}

", file="assay2.jag")

**# Initial values and Estimation**

ini1 <- list(beta=c(-20,20),sig=1,lam=1); ini2 <- list(beta=c(-15,15),sig=2,lam=0.5)

ini <- list(ini1,ini2)

M2 <- jags.model(inits=ini,data= DS\_7\_8,n.chains=2, file="assay2.jag",n.adapt=500)

S2.1 <- coda.samples(M2,c("beta","lam"),n.iter=10000)

gelman.diag(S2.1)

S2.2 <- coda.samples(M2, c("beta","lam","C.F","ynew"),n.iter=5000)

summary(S2.2)

**# predictive variance**

M=as.matrix(S2.2)

M2=M[1:10000,5:117]

predvar=apply(M2,2,sd)^2

sum(predvar)

**# Student t scale mixture**

cat("model { for (i in 1:113) {y[i] ~ dnorm(beta[1]+beta[2]\*x[i],kappa[i]/sig^2);

ynew[i] ~ dnorm(beta[1]+beta[2]\*x[i], kappa[i]/sig^2)

**# predictive discrepancy**

predF[i] <- pow(y[i]-ynew[i],2)

**# Student t precision adjustments**

kappa[i] ~ dgamma(nu/2,nu/2)}

for (j in 1:2) {beta[j] ~ dnorm(0,0.001)}

nu <- 1/inv.nu

inv.nu ~ dunif(0.01,1)

sig ~ dunif(0,250)

C.F <- sum(predF[])}

", file="assay3.jag")

**# Initial values and Estimation**

ini1 <- list(beta=c(0,0),sig=10,inv.nu=0.5); ini2 <- list(beta=c(-20,20),sig=5,inv.nu=0.5)

ini <- list(ini1,ini2)

M3 <- jags.model(inits=ini,data= DS\_7\_8,n.chains=2, file="assay3.jag",n.adapt=500)

S3.1 <- coda.samples(M3,c("beta","Tau","nu","C.F"),n.iter=10000)

gelman.diag(S3.1)

S3.2 <- coda.samples(M3, c("beta","nu","C.F","ynew","kappa"),n.iter=5000); summary(S3.2)

**# predictive variance**

M=as.matrix(S3.2)

M3=M[1:10000,118:230]

predvar=apply(M3,2,sd)^2

sum(predvar)

**# discrete mixture, K components**

KM = DS\_7\_8\_DM$K-1

K = DS\_7\_8\_DM$K

cat("model { for (i in 1:N) {y[i] ~ dnorm(alph[G[i]]+beta[G[i]]\*x[i], 1/sig[G[i]]^2)

r[i] <- (y[i]- alph[G[i]]-beta[G[i]]\*x[i])/sig[G[i]]

**# log-likelihood**

LL[i] <- -0.5\*r[i]\*r[i]-0.5\*log(sig[G[i]])

ynew[i] ~ dnorm(alph[G[i]]+beta[G[i]]\*x[i], 1/sig[G[i]]^2)

**# predictive discrepancy**

predF[i] <- pow(y[i]-ynew[i],2)

**# latent component index**

G[i] ~ dcat(pi[1:K])

**# monitor index to obtain posterior mean allocation probabilities**

for (k in 1:K) {index[i,k] <- equals(G[i],k)}}

pi[1:K] ~ ddirch(w[1:K])

for (j in 1:K) {w[j] <- 1

alph[j] ~ dnorm(0,0.001)

sig[j] ~ dunif(0,1000)}

**# ascending beta prior**

for (k in 1:K) {beta1[k] ~ dnorm(0,0.01)}

beta=sort(beta1)

**# Fit measures**

BIC <- -2\*sum(LL[])+(p\*K+2\*K)\*log(N)

C.F <- sum(predF[])}

",file="assay4.jag")

**# Initial values and Estimation**

ini1 <- list(beta1=rep(0,K), sig=rep(75,K),alph= rep(-50,K))

ini2 <- list(beta1=rep(0,K), sig=rep(50,K),alph= rep(-25,K))

ini <- list(ini1,ini2)

M4 <- jags.model(inits=ini,data= DS\_7\_8\_DM,n.chains=2, file="assay4.jag",n.adapt=500)

S4.1 <- coda.samples(M4,c("alph","beta" ,"sig","C.F","BIC","pi"),n.iter=25000)

gelman.diag(S4.1,multivariate=F)

summary(S4.1)

S4.2 <- coda.samples(M4, c("alph","beta","sig","C.F","BIC","ynew","pi"),n.iter=5000)

summary(S4.2)