require(jagsUI)

library(rjags)

attach("C:/R files BHMRA/DS\_5\_3.Rdata")

#

# Model 1. Random Level plus Global Linear Trend

#

**cat("**model { for (t in 1:T){

# random level

M.cen[t] <- beta0+M[t]-M[1]

# global sea level

GSL[t] <- beta0+M[t]+b\*(t-1)-M[1]

# model for j=1,..,5 sites

for (j in 1:J) { mu[t,j] <- beta0+M[t]+b\*(t-1)-M[1]

y[t,j] ~ dnorm(mu[t,j],tau)

# replicates

ynew[t,j] ~ dnorm(mu[t,j],tau)

# test criteria

d1[t,j] <- pow(ynew[t,j]-y[t,j],2)

d2[t,j] <- pow(y[t,j]-mu[t,j],2)

d3[t,j] <- pow(ynew[t,j]-mu[t,j],2)}}

for (t in 2:T){ M[t] ~ dnorm(M[t-1],tau.M)}

# posterior predictive test

PPC <- step(sum(d3[1:81,])-sum(d2[1:81,]))

# first component of posterior predictive loss

PPL1 <- sum(d1[1:81,])\*1000/1001

# State Space Initial Condition

M1 ~ dnorm(6900,0.0001)

M[1] <- M1

# Priors

beta0 ~ dnorm(0,0.0000001)

kappa ~ dunif(0,1)

xi ~ dgamma(1,0.001)

tau <- kappa\*xi

tau.M <- (1-kappa)\*xi;

b ~ dnorm(0,1)

sig2[1] <- 1/tau

sig2[2] <- 1/tau.M}

**", file="model1.jag")**

# Initial values and estimation

init1=list(xi=1,M1=6900,beta0=6900,b=0.8)

init2=list(xi=0.001,M1=6900,beta0=6900,b=0.5)

inits=list(init1,init2)

pars <- c("b","sig2","M.cen","PPL1","PPC","GSL","beta0","ynew")

R1 <- autojags(DS\_5\_3, inits, pars,model.file="model1.jag",2,iter.increment=20000, n.burnin=500,Rhat.limit=1.1, max.iter=100000, seed=1234,codaOnly=c("ynew"))

R1$summary

**# Second component of posterior predictive loss**

J <- jags.model(inits=inits,data=DS\_5\_3,n.chains=2, file="model1.jag")

update(J,10000)

ynew <- coda.samples(J, c("ynew"),n.iter=1000)

ynew <- as.matrix(ynew)

postvar=apply(ynew,2,sd)^2

PPL2=sum(postvar)

#

# Model 2. Site Specific Linear Growth

#

**cat("**model { for (t in 1:T){

# M series

M.cen[t] <- beta0+M[t]-M[1]

# global sea level

GSL[t] <- beta0+M[t]+mu.b\*(t-1)-M[1]

# model for j=1,..,5 sites

for (j in 1:J) { mu[t,j] <- beta0+M[t]+b[j]\*(t-1)-M[1]

y[t,j] ~ dnorm(mu[t,j],tau)

ynew[t,j] ~ dnorm(mu[t,j],tau)

d1[t,j] <- pow(ynew[t,j]-y[t,j],2)

d2[t,j] <- pow(y[t,j]-mu[t,j],2)

d3[t,j] <- pow(ynew[t,j]-mu[t,j],2)}}

for (t in 2:T){ M[t] ~ dnorm(M[t-1],tau.M)}

# posterior predictive test

PPC <- step(sum(d3[1:81,])-sum(d2[1:81,]))

# first component of posterior predictive loss

PPL1 <- sum(d1[1:81,]) \*1000/1001

# State Space Initial Condition

M1 ~ dnorm(6900,0.0001)

M[1] <- M1

# Priors

for (j in 1:J) {b[j] ~ dnorm(mu.b,tau.b)}

tau.b ~ dexp(1);

mu.b ~ dnorm(0,1);

beta0 ~ dnorm(0,0.0000001)

kappa ~ dunif(0,1)

xi ~ dgamma(1,0.001)

tau <- kappa\*xi

tau.M <- (1-kappa)\*xi;

sig2[1] <- 1/tau

sig2[2] <- 1/tau.M

sig2[3] <- 1/tau.b}

**", file="model2.jag")**

# Initial values and estimation

init1= list(beta0=6900,xi=1,M1=6900,b=c(0,0,0,0,0),tau.b=1)

init2= list(beta0=6950,xi=0.001,M1=6900,b=c(0,0,0,0,0),tau.b=0.001)

inits=list(init1,init2)

pars <- c("b","sig2","mu.b","M.cen","PPL1","PPC","GSL","beta0")

R2 <- autojags(DS\_5\_3, inits, pars,model.file="model2.jag",2,iter.increment=10000, n.burnin=500,Rhat.limit=1.05, max.iter=100000, seed=1234)

R2$summary

# Second component of posterior predictive loss

J <- jags.model(inits=inits,data=DS\_5\_3,n.chains=2, file="model2.jag")

update(J,10000)

ynew <- coda.samples(J, c("ynew"),n.iter=1000)

ynew <- as.matrix(ynew)

postvar=apply(ynew,2,sd)^2

PPL2=sum(postvar)