setwd("C:/R files BHMRA")

require(mcmcse); require(loo); library(MCMCvis)

library(ggplot2); library(reshape); options(scipen=999)

attach("DS\_12\_3.Rdata")

library(jagsUI)

**# MODEL 1**

cat("model { for (i in 1:400) {y[i] ~ dnorm(mu[i],invsig2)

mu[i] <- alph+beta[1]\*(fsm[i]-mean(fsm[]))

+beta[2]\*(elp[i]-mean(elp[]))

+beta[3]\*(emcred[i]-mean(emcred[]))

LL[i] <- 0.5\*log(invsig2/6.283)-0.5\*invsig2\*resid[i]^2

resid[i] <- y[i]-mu[i]}

for (j in 1:3) {beta[j] ~ dnorm(0,0.1)}

alph ~ dnorm(0,0.00001)

invsig2~dgamma(1,0.001)

sig2 <- 1/invsig2}

", file="model1.jag")

**# initial values and estimation**

init1 <- list(invsig2=0.001,alph=500,beta=c(0,0,0))

init2 <- list(invsig2=0.0005,alph=650,beta=c(-3,-0.9,-1.6))

inits <- list(init1,init2)

pars <- c("beta","sig2","LL","mu","resid")

R1 = autojags(DS\_12\_3, inits, pars,model.file="model1.jag",2,iter.increment=2500, n.burnin=500, Rhat.limit=1.1, max.iter=10000, seed=1234,

codaOnly= c("LL","mu","resid"))

R1$summary

**# Effective sample sizes**

ess=ess(as.data.frame(R1$sims.list))

**# Fit**

loo(as.matrix(R1$sims.list$LL))

**# Diagnostic Plots,**

# Residuals against Fitted

mu.m=apply (as.matrix(R1$sims.list$mu),2,mean)

res.m=apply (as.matrix(R1$sims.list$resid),2,mean)

plot(mu.m,res.m, xlab="Fitted",ylab="Residuals", main="Figure 12.5.1 Residuals against Fitted, Homoscedastic Model")

u <- par("usr")

rect(u[1], u[3], u[2], u[4], col = "lightgrey", border = "black")

points(mu.m, res.m, pch = 16, col = "black")

# Residuals against Predictor FSM

plot(DS\_12\_3$emcred,res.m, xlab="FSM",ylab="Residuals", main="Figure 12.5.2 Residuals against Free School Meals")

u <- par("usr")

rect(u[1], u[3], u[2], u[4], col = "lightgrey", border = "black")

points(DS\_12\_3$emcred, res.m, pch = 16, col = "black")

**# MODEL 2 Heteroscedastic, Corner Constraint Smooth**

DS\_12\_3$K=9

DS\_12\_3$kap=c(11, 23, 40.7, 57, 67.5, 76, 84, 93, 99)

cat("model { for (i in 1:400) {

y[i] ~ dnorm(mu[i],inv.sig2[i])

resid[i] <- y[i]-mu[i]

LL[i] <- 0.5\*log(inv.sig2[i]/6.283)-0.5\*inv.sig2[i]\*resid[i]^2

mu[i] <- beta0+beta[1]\*(fsm[i]-mean(fsm[]))

+beta[2]\*(elp[i]-mean(elp[]))

+beta[3]\*(emcred[i]-mean(emcred[]))

# Heteroscedasticity Regression

log(sig2[i]) <- gam0 +sum(S[i,])

inv.sig2[i] <- 1/sig2[i]

log.variance[i] <- log(sig2[i])

for (k in 1:K) {S[i,k] <- c[k]\*step(fsm[i]-kap[k])\*pow(fsm[i]-kap[k],3)}}

for (j in 1:3) {beta[j] ~ dnorm(0,0.1)}

gam0 ~ dnorm(0,0.001)

beta0 ~ dnorm(0,0.000001)

cbar <- mean(c[])

c[1] <- 0

for (k in 2:K) {c[k] ~ dnorm(0,inv.phi.c)}

for (k in 1:K) {c.cent[k] <- c[k] - cbar}

inv.phi.c ~ dgamma(1,1)

phi.c.sqrt <- 1/sqrt(inv.phi.c)}

", file="model2.jag")

**# Initial values and estimation**

init1 <- list(gam0=2, beta0=600,beta=c(-3.2,-0.9,-1.8),inv.phi.c=7,c=c(NA,rep(0,8)))

init2 <- list(gam0=2.5,beta0=650,beta=c(-3,-0.8,-1.6),inv.phi.c=8, c=c(NA,rep(0,8)))

inits <- list(init1,init2)

pars <- c("beta","gam0","c.cent","phi.c.sqrt","LL","log.variance")

R2 = autojags(DS\_12\_3, inits, pars,model.file="model2.jag",2,iter.increment=2500, n.burnin=500, Rhat.limit=1.1, max.iter=20000, seed=1234,codaOnly= c("LL","log.variance"))

R2$summary

**# Effective sample sizes**

ess=ess(as.data.frame(R2$sims.list))

**# Fit**

loo(as.matrix(R2$sims.list$LL))

**# Log Variance against Predictor FSM**

logvar=apply (as.matrix(R2$sims.list$log.variance),2,mean)

plot(DS\_12\_3$fsm,logvar, xlab="FSM",ylab="Log Variance",main="Figure 12.5.3")

u <- par("usr")

rect(u[1], u[3], u[2], u[4], col = "lightgrey", border = "black")

points(DS\_12\_3$fsm,logvar, pch = 16, col = "black")

**# MODEL 3 Heteroscedastic, Centred Smooth**

cat("model { for (i in 1:400) {y[i] ~ dnorm(mu[i],inv.sig2[i])

resid[i] <- y[i]-mu[i]

LL[i] <- 0.5\*log(inv.sig2[i]/6.283)-0.5\*inv.sig2[i]\*resid[i]^2

mu[i] <- beta0+beta[1]\*(fsm[i]-mean(fsm[]))

+beta[2]\*(elp[i]-mean(elp[]))

+beta[3]\*(emcred[i]-mean(emcred[]))

# Heteroscedasticity Regression

log(sig2[i]) <- gam0 + tS[i]-mean(tS[])

tS[i] <- sum(S[i,])

inv.sig2[i] <- 1/sig2[i];

log.variance[i] <- log(sig2[i])

for (k in 1:K) {S[i,k] <- c[k]\*step(fsm[i]-kap[k])\*pow(fsm[i]-kap[k],3)}}

for (j in 1:3) {beta[j] ~ dnorm(0,0.1)}

gam0 ~ dnorm(0,0.001)

beta0 ~ dnorm(0,0.000001)

cbar <- mean(c[])

for (k in 1:K) {c[k] ~ dnorm(0,inv.phi.c)

c.cent[k] <- c[k] - cbar}

inv.phi.c ~ dgamma(1,1)

phi.c.sqrt <- 1/sqrt(inv.phi.c)}

", file="model3.jag")

**# Initial values and estimation**

init1 <- list(gam0=2, beta0=600,beta=c(-3.2,-0.9,-1.8),inv.phi.c=7,c=c(rep(0,9)))

init2 <- list(gam0=2.5,beta0=650,beta=c(-3,-0.8,-1.6),inv.phi.c=8, c=c(rep(0,9)))

inits <- list(init1,init2)

pars <- c("beta","gam0","c","phi.c.sqrt","LL","log.variance")

R3 = autojags(DS\_12\_3, inits, pars,model.file="model3.jag",2,iter.increment=2500, n.burnin=500, Rhat.limit=1.1, max.iter=20000, seed=1234,codaOnly= c("LL","log.variance"))

R3$summary

**# Effective sample sizes**

ess=ess(as.data.frame(R3$sims.list))

**# Fit**

loo(as.matrix(R3$sims.list$LL))

**# Log Variance against Predictor FSM**

logvar=apply (as.matrix(R3$sims.list$log.variance),2,mean)

plot(DS\_12\_3$fsm,logvar, xlab="FSM",ylab="Log Variance")

u <- par("usr")

rect(u[1], u[3], u[2], u[4], col = "lightgrey", border = "red")

points(DS\_12\_3$fsm,logvar, pch = 16, col = "red")