library(R2OpenBUGS);

library(coda); library(loo); library(mcmcplots); library(brms)

setwd("C:/R files BHMRA")

# 134 classes, 28 schools, n[] are numbers of classes within each school

attach("DS\_8\_3.Rdata")

DBRMS <- read.table("Example\_8\_3\_TVSFP\_brms.txt",header=T)

# BRMS using rstan

# School and class varying intercepts

BRMS <- brm(thk ~ prethk+cc+tv+cctv+(1|sch)+(1|cls), data=DBRMS, family =cumulative, chains = 2)

summary(BRMS)

**# R2OpenBUGS Ordinal Logistic**

**model1 <- function() {**for (h in 1:N) { thk[h] ~ dcat(pi[h,])

for (j in 1:H-1) {logit(gam[h,j]) <- kap[j] - mu[h] }

pi[h,1] <- gam[h,1]; pi[h,H] <- 1-gam[h,H-1]

for (r in 2:H-1) { pi[h,r] <- gam[h,r] - gam[h,r-1] }

eta[h] <- beta[1]\*(prethk[h]-mean(prethk[]))

+beta[2]\*(cc[h]-mean(cc[]))

+beta[3]\*(tv[h]-mean(tv[]))

+beta[4]\*(cctv[h]-mean(cctv[]))

mu[h] <- eta[h]+u3[sch[h]]+u2[sch[h],cls[h]]}

for (i in 1:m) {u3[i] ~dnorm(0,tau[3]);

for (j in 1:n[i]) {u2[i,j] ~dnorm(0,tau[2])}}

for (j in 2:3) {sig[j] ~ dunif(0,1000);

sig2[j] <- sig[j]\*sig[j]; tau[j] <- 1/sig2[j]}

for (j in 1:4) {beta[j] ~ dnorm(0,0.001)}

kap[1] ~ dnorm(0, 1) %\_% I(,kap[2]);

kap[2] ~ dnorm(0, 1) %\_% I(kap[1],kap[3]);

kap[3] ~ dnorm(0, 1) %\_% I(kap[2],)}

**# Initial Values**

init1 <- list(beta=c(0,0,0,0),kap=c(0.5,1.5,2.5),sig=c(NA,0.5,0.5))

init2 <- list(beta=c(0,0,0,0),kap=c(1,2,3),sig=c(NA,1,1))

inits <- list(init1,init2)

n.iters=2500; n.burnin =500; n.chains=2

**# MODEL 1 HYPERPARAMETERS**

pars <- list("beta","sig","kap")

M1 <- bugs(DS\_8\_3,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = T,bugs.seed=10)

M1.coda <- read.bugs(M1)

summary(M1.coda)

gelman.diag(M1.coda,multivariate=F)

**# R2OpenBUGS Augmented Data**

**model2 <- function() {**for (h in 1:N) { for (j in 1:H-1) {

z[h,j] <- step(j-thk[h])

A[h,j] <- -10\*equals(z[h,j],0)

B[h,j] <- 10\*equals(z[h,j],1)

zstar[h,j] ~ dlogis(nu[h,j],1) %\_% I(A[h,j],B[h,j])

nu[h,j] <- kap[j] - mu[h]}

# regression term

mu[h] <- eta[h] +u3[sch[h]]+u2[sch[h],cls[h]]

# compare predicted and actual ordinal categories

pgrp1[h] <- step(kap[1]-zstar.mx[h])\*equals(thk[h],1)

pgrp2[h] <- step(zstar.mx[h]-kap[1])\*step(kap[2]-zstar.mx[h])\*equals(thk[h],2)

pgrp3[h] <- step(zstar.mx[h]-kap[2])\*step(kap[3]-zstar.mx[h])\*equals(thk[h],3)

pgrp4[h] <- step(zstar.mx[h]-kap[3])\*equals(thk[h],4)

# testmx=1 for a correct classifications

testmx[h] <- pgrp1[h]+ pgrp2[h]+ pgrp3[h]+ pgrp4[h]

# mixed predictive replicates

zstar.mx[h] ~ dlogis(mu.mx[h],1)

eta[h] <- beta[1]\*(prethk[h]-mean(prethk[]))+beta[2]\*(cc[h]-mean(cc[]))

+beta[3]\*(tv[h]-mean(tv[]))+beta[4]\*(cctv[h]-mean(cctv[]))

mu.mx[h] <- eta[h]+u3.mx[sch[h]]+u2.mx[sch[h],cls[h]]

# predicted probabilities

logit(q[h,1]) <- kap[1]-mu[h]

logit(q[h,2]) <- kap[2]-mu[h]

logit(q[h,3]) <- kap[3]-mu[h]

p[h,4] <- 1-q[h,3]

p[h,3] <- q[h,3]-q[h,2]

p[h,2] <- q[h,2]-q[h,1]

p[h,1] <- q[h,1]

LL[h] <- equals(thk[h],1)\*log(p[h,1])+ equals(thk[h],2)\*log(p[h,2])+ equals(thk[h],3)\*log(p[h,3])+

equals(thk[h],4)\*log(p[h,4])}

# Priors

for (i in 1:m) {u3[i] ~dnorm(0,tau[3])

u3.mx[i] ~dnorm(0,tau[3])

# j=1,..,n[i], index classes in school i

for (j in 1:n[i]) {u2[i,j] ~dnorm(0,tau[2])

u2.mx[i,j] ~dnorm(0,tau[2])}}

for (j in 2:3) {sig[j] ~ dunif(0,1000);

sig2[j] <- sig[j]\*sig[j]

tau[j] <- 1/sig2[j]}

for (j in 1:4) {beta[j] ~ dnorm(0,0.001)}

kap[1] ~ dnorm(0, 1) %\_% I(,kap[2]);

kap[2] ~ dnorm(0, 1) %\_% I(kap[1],kap[3]);

kap[3] ~ dnorm(0, 1) %\_% I(kap[2],)}

**# Initial Values**

init1 <- list(beta=c(0,0,0,0),kap=c(0.5,1.5,2.5),sig=c(NA,0.5,0.5))

init2 <- list(beta=c(0,0,0,0),kap=c(1,2,3),sig=c(NA,1,1))

inits <- list(init1,init2)

n.iters=2500; n.burnin =500; n.chains=2

**# Model 2 Hyperparameter Estimates**

pars <- list("beta","sig","kap")

M2 <- bugs(DS\_8\_3,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = T,bugs.seed=10)

M2.coda <- read.bugs(M2); gelman.diag(M2.coda,multivariate=F); summary(M2.coda)

**# Model 2 Fit Measures: Waic and CV Concordance**

pars <- list("testmx", "LL")

M2.L <- bugs(DS\_8\_3,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,

codaPkg = F,bugs.seed=10)

WAIC=waic(M2.L$sims.list$LL)

names(WAIC)

waic.pw=WAIC$pointwise[,3]

PPD=apply(M2.L$sims.list$testmx,2,mean)

**# subjects with worst fit**

sapply(sort(waic.pw, index.return=T), `[`, 1591:1600)

**# subjects with lowest cross-validatory concordance**

sapply(sort(PPD, index.return=T), `[`, 1:10)