setwd("C:/R files BHMRA")

library(rjags)

library(rstan)

require(normtest)

attach("DS\_10\_5.Rdata")

attach("DS\_10\_5\_STAN.Rdata")

**# RJAGS MODEL WISHART PRIOR**

cat("

model { for (i in 1:n) { b[i,1:2] ~ dmnorm(B[1:2],InvD[,])

for (t in 1:T) { mu[i,t] <- b[i,1]+b[i,2]\*Z[i,t] +beta[1]\*Tr[i]+ beta[2]\*Tr[i]\*Z[i,t]+beta[3]\*Gend[i]

for (j in 1:K) {d[i,t,j] <- equals(y[i,t],j)

dnew[i,t,j] <- equals(ynew[i,t],j)

log\_lik[i,t,j] <- d[i,t,j]\*log(p[i,t,j])

log\_lik\_new[i,t,j] <- dnew[i,t,j]\*log(p[i,t,j])}

for (j in 1:KM) {logit(nu[i,t,j]) <- kap[j] - mu[i,t]}

p[i,t,1] <- nu[i,t,1];

for (j in 2:KM) { p[i,t,j] <- nu[i,t,j] - nu[i,t,j-1] }

p[i,t,K] <- 1-nu[i,t,KM];

y[i,t] ~ dcat(p[i,t,])

ynew[i,t] ~ dcat(p[i,t,])}}

# thresholds for latent variable

kap[1] <- 0

for (k in 2:KM){ kap[k] <- kap[k-1]+del[k]

del[k] ~ dgamma(1,1)}

# random effects prec'n matrix

InvD[1:2,1:2] ~ dwish(Q.b[,],2)

D[1:2,1:2] <- inverse(InvD[,])

for (i in 1:2) {B[i] ~ dnorm(0,0.01)

sig[i] <- sqrt(D[i,i])

for (j in 1:2) {corr.b[i,j] <- D[i,j] / (sig[i]\*sig[j])}}

Q.b[1,1] <- 1

Q.b[2,2] <- 1

Q.b[2,1] <- 0

Q.b[1,2] <- 0

# posterior predictive check for total likelihoods

PPC <- step(sum(log\_lik\_new[,,])-sum(log\_lik[,,]))

# covariate effects

for (j in 1:3) {beta[j] ~ dnorm(0,0.01)}}

", file="ordsympt.jag")

**# Initial Values and Estimation**

B0 = matrix(0,324,2)

D0.1 = diag(10,2)

D0.2 = diag(20,2);

init1 = list(B=c(0,0),beta=c(0,0,0),del=c(NA,1,1,1,1,1), InvD=D0.1,b=B0)

init2 = list(B=c(0,0),beta=c(0,0,0),del=c(NA,2,2,2,2,2), InvD=D0.2,b=B0)

inits = list(init1,init2)

pars = c("kap","sig","corr.b[1,2]","beta","B","PPC")

M1 = jags.model(inits=inits,data=DS\_10\_5,n.chains=2, n.adapt=2000,file="ordsympt.jag")

update(M1,5000)

S1= coda.samples(M1,pars,n.iter=5000)

summary(S1)

**# Normality of random effects**

S1.b <- coda.samples(M1, c("b"),n.iter=500)

b.mn=apply(as.matrix(S1.b),2,mean)

qqnorm(b.mn[1:324])

qqnorm(b.mn[325:648])

ajb.norm.test(b.mn[1:324], nrepl=1000)

ajb.norm.test(b.mn[325:648], nrepl=1000)

**# STAN MODEL LKJ PRIOR**

**# LKJ Prior on Cholesky Factor of Correlation Matrix**

ordsympt.stan <- "

data {

int<lower=2> K;

int<lower=0> N;

int<lower=0> n;

int<lower=1> D;

int<lower=1,upper=K> y[N];

vector[N] time;

int<lower=1,upper=n> subj[N];

vector[n] trt;

vector[n] gend;

}

parameters {

vector[D] beta;

ordered[K-1] kap;

vector[2] b[n];

vector[2] mu\_b;

cholesky\_factor\_corr[2] Lcorr;

vector<lower=0>[2] sigb;

}

model {

mu\_b[1] ~ normal(0,0.001);

mu\_b[2] ~ normal(0, 100);

sigb ~ cauchy(0, 5);

Lcorr ~ lkj\_corr\_cholesky(1.5);

b ~ multi\_normal\_cholesky(mu\_b, diag\_pre\_multiply(sigb, Lcorr));

for (k in 1:N)

y[k] ~ ordered\_logistic(b[subj[k],1] + b[subj[k],2]\*time[k]

+beta[1]\*trt[subj[k]] + beta[2]\*time[k]\*trt[subj[k]]+beta[3]\*gend[subj[k]], kap);

}

generated quantities {

matrix[2,2] Omega;

matrix[2,2] Sigma\_b;

matrix[2,2] R;

Omega = multiply\_lower\_tri\_self\_transpose(Lcorr);

Sigma\_b = quad\_form\_diag(Omega, sigb);

for (i in 1:2) for (j in 1:2) R[i,j] = Sigma\_b[i,j]/ sqrt(Sigma\_b[i,i]\*Sigma\_b[j,j]);

}

"

sm <- stan\_model(model\_code=ordsympt.stan)

fit <- sampling(sm, data =DS\_10\_5\_STAN, iter = 2000,warmup=500,chains = 2,seed=12345)

pars = c("kap","mu\_b","beta","sigb","R")

summary(fit, pars = pars, probs = c(0.025,0.50, 0.975))$summary