setwd("C:/R files BHMRA")

library(R2OpenBUGS); library(coda); library(mcmcplots); library(rstan); library(loo); options(scipen=999)

attach("DS\_10\_1.RData")

**# Wishart prior on precision**

model1 <- function() {for (i in 1:500) {

for (t in 1:5) {y[i,t] ~ dnorm(mu[i,t],tau)

mu[i,t] <- b[i,1]+b[i,2]\*(t-3)+b[i,3]\*(x[i,t]-mean(x[,]))}

b[i,1:3] ~ dmnorm(B[1:3],inv.D[,])}

# intercept adjusted for centering

B1 <- B[1] - B[2] \* 3 - B[3]\*(-0.0067)

# residual precision

tau ~ dgamma(1,0.001)

sig2 <- 1/tau

# Scale matrix for Wishart prior

for (k in 1:3) {Q[j,k] <- 0.1\*equals(j,k)

R[j,k] <- D[j,k]/(sigb[j]\*sigb[k])}}

inv.D[1:3,1:3] ~ dwish(Q[,],3)

D[1:3,1:3] <- inverse(inv.D[1:3,1:3])

for (j in 1:3) {sigb[j] <- sqrt(D[j,j])

B[j] ~ dnorm(0,0.01)}

**# Model 1 Hyperparameters**

inits <- list(list(inv.D=structure(.Data=c(1,0,0, 0,10,0, 0,0,10),.Dim=c(3,3)), B=c(5,0.5,0.1),tau=0.5),

list(inv.D=structure(.Data=c(2,0,0, 0,20,0, 0,0,20),.Dim=c(3,3)),

B=c(5,0,0),tau=1))

n.iters=5000; n.burnin =1000; n.chains=2

pars <- c("sigb","D","B","B1","sig2","R")

M1 <- bugs(DS\_10\_1,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

M1$summary

**# Random effects selection, approach 1**

data <- list(y=DS\_10\_1$y, x=DS\_10\_1$x,Q=3,agam=c(5.7,1.1,0.35),

bgam=c(5.1,4.7,0.95))

model2 <- function() { for (i in 1:500) { for (t in 1:5) {

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

z1[i,t] <-1

z2[i,t] <- (t-3)

z3[i,t] <- x[i,t] -mean(x[,])

mu[i,t] <- B[1]+B[2]\*(t-3)+B[3]\*(x[i,t] -mean(x[,]))

+zeta[i,1]\*(z1[i,t]\*G[1,1]+z2[i,t]\*G[2,1]+z3[i,t]\*G[3,1])

+zeta[i,2]\*(z2[i,t]\*G[2,2]+z3[i,t]\*G[3,2])

+zeta[i,3]\*(z3[i,t]\*G[3,3])}

for (k in 1:3) {zeta[i,k] ~ dnorm(0,1)}}

# intercept adjusted for centering

B1 <- B[1] - B[2] \* 3 - B[3]\*(-0.0067)

G[1,2] <- 0; G[1,3] <- 0; G[2,3] <- 0

for (p in 1:Q) { sigb[p] <- sqrt(D[p,p])

# realised covariance terms

for (q in 1:Q) {D[p,q] <- sum(G.s[p,q,]);

for (r in 1:Q) {G.s[p,q,r] <- G[p,r]\*G[q,r]}}}

tau ~ dgamma(1,0.001)

sig2 <- 1/tau

# Covariance Selection Priors

for (k in 1:Q) {B[k] ~ dnorm(0,0.01)

c[k,k] ~ dgamma(agam[k],bgam[k]);

G[k,k] <- c[k,k]\*gam[k,k];

gam[k,k] ~ dbern(pi[k,k])

pi[k,k] ~ dbeta(1,1)

for (m in 1:k-1){c[k,m] ~ dnorm(0,1)

G[k,m] <- c[k,m]\*gam[k,m]

gam[k,m] ~ dbern(pi[k,m])

pi[k,m] ~ dbeta(1,1)}}}

# Model 2 Hyperparameters

ini1 <- list(B=c(5,0.5,0.1),tau=0.5,

c = structure(.Data = c(1, NA, NA, 0,0.1,NA, 0,0,0.1),.Dim = c(3,3)),

gam = structure(.Data = c(1, NA, NA, 0,1,NA, 0,0,1),.Dim = c(3,3)),

pi = structure(.Data = c(0.5,NA,NA, 0.5,0.5,NA, 0.5,0.5,0.5),.Dim = c(3,3)))

ini2 <- list(B=c(5,0,0),tau=5,

c = structure(.Data = c(1, NA, NA, 0,0.2,NA, 0,0,0.2),.Dim = c(3,3)),

gam = structure(.Data = c(1, NA, NA, 0,1,NA, 0,0,1),.Dim = c(3,3)),

pi = structure(.Data = c(0.5,NA,NA, 0.5,0.5,NA, 0.5,0.5,0.5),.Dim = c(3,3)))

n.iters=5000; n.burnin =1000; n.chains=2

inits=list(ini1,ini2)

pars <- c("sigb","D","B1","B","sig2","gam")

M2 <- bugs(data,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = F)

M2$summary

**# Random effects selection, Approach 2**

data = list(y=DS\_10\_1$y, x=DS\_10\_1$x,Q=3,agam=c(5.7,1.1,0.35),bgam=c(5.1,4.7,0.95))

model3 <- function() { for (i in 1:500) { for (t in 1:5) {

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

z1[i,t] <-1

z2[i,t] <- t-3

z3[i,t] <- x[i,t]- mean(x[,])

mu[i,t] <- B[1]+B[2]\*(t-3)+B[3]\*(x[i,t]- mean(x[,]))

+zeta[i,1]\*(z1[i,t]\*G[1,1]+z2[i,t]\*G[2,1]+z3[i,t]\*G[3,1])

+zeta[i,2]\*(z2[i,t]\*G[2,2]+z3[i,t]\*G[3,2])

+zeta[i,3]\*(z3[i,t]\*G[3,3])}

for (k in 1:3) {zeta[i,k] ~ dnorm(0,1)}}

for (p in 1:Q) { sigb[p] <- sqrt(D[p,p])

# realised covariance

for (q in 1:Q) {D[p,q] <- sum(G.s[p,q,])

for (r in 1:Q) {G.s[p,q,r] <- G[p,r]\*G[q,r]}}}

# intercept adjusted for centering

B1 <- B[1] - B[2] \* 3 - B[3]\*(-0.0067)

tau ~ dgamma(1,0.001)

sig2 <- 1/tau

# gam.w ~ dbeta(1,1)

gam.w <- 0.5

for (k in 1:Q) {B[k] ~ dnorm(0,0.01)

lam[k]~dgamma(agam[k],bgam[k]);

gamma[k,k] ~ dbern(gam.w)

omega[k,k] <- 1

G[k,k] <- lam[k]\*gamma[k,k]

for (m in k+1:Q){omega[k,m]<- 0

gamma[k,m]<- 0

G[k,m] <- 0}

for (m in 1:k-1) {omega[k,m] ~ dnorm(0,2)

gamma[k,m] ~ dbern(gam.w)

# realised covariance terms

G[k,m] <- lam[k]\*omega[k,m]\*gamma[k,k]\*gamma[m,m]\*gamma[k,m]}}}

**# Model 3 Hyperparameters**

ini1 = list(B=c(5,0.5,0.1),tau=0.5,lam=c(1,1,1),

omega = structure(.Data = c(NA,NA,NA, 0,NA,NA, 0,0,NA),.Dim = c(3,3)),

gamma = structure(.Data = c(1,NA,NA, 1,1,NA, 1,1,1),.Dim = c(3,3)))

ini2 = list(B=c(5,0,0),tau=1,lam=c(2,2,2),

omega = structure(.Data = c(NA,NA,NA, 0,NA,NA, 0,0,NA),.Dim = c(3,3)),

gamma = structure(.Data = c(1,NA,NA, 1,1,NA, 1,1,1),.Dim = c(3,3)))

n.iters=5000; n.burnin =1000; n.chains=2

inits=list(ini1,ini2)

pars = c("sigb","D","B1","B","sig2","gamma")

M3 = bugs(data,inits,pars,n.iters,model3,n.chains, n.burnin,debug=T,codaPkg = F)

M3$summary

sigbsamps=M3$sims.list$sigb

h2=hist(sigbsamps[,2])

d2=density(sigbsamps[,2])

plot(d2)

h3=hist(sigbsamps[,3])

d3=density(sigbsamps[,3])

plot(d3)

**#**

**# LKJ Prior On Lower Cholesky Factor of Correlation Matrix**

**#**

data <- list(y=DS\_10\_1$y, x=DS\_10\_1$x,N=500,T=5)

growthLKJ.stan <- "

data {

int<lower=0> N;

int<lower=0> T;

real x[N,T];

real y[N,T];

}

parameters {

vector[3] b[N];

vector[3] mu\_b;

real<lower=0> sigmasq\_y;

cholesky\_factor\_corr[3] Lcorr;

vector<lower=0>[3] sigma;

}

transformed parameters {

real<lower=0> sigma\_y;

sigma\_y = sqrt(sigmasq\_y);

}

model {

sigmasq\_y ~ inv\_gamma(0.001, 0.001);

mu\_b ~ normal(0, 100);

sigma ~ cauchy(0, 5);

sigma\_b ~ cauchy(0, 5);

Lcorr ~ lkj\_corr\_cholesky(1.5);

b ~ multi\_normal\_cholesky(mu\_b,

diag\_pre\_multiply(sigma\_b, Lcorr));

for (n in 1:N) for (t in 1:T) // centered x[]

y[n,t] ~ normal(b[n,1] + b[n,2]\*(t-3) + b[n,3] \*x[n,t], sqrt(sigmasq\_y));

}

generated quantities {

matrix[3,3] Omega;

matrix[3,3] Sigma\_b;

matrix[3,3] R;

matrix[N,T] log\_lik;

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

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

for (i in 1:3) for (j in 1:3) R[i,j] = Sigma\_b[i,j]/ sqrt(Sigma\_b[i,i]\*Sigma\_b[j,j]);

for (n in 1:N) for (t in 1:T) log\_lik[n,t] = -(y[n,t]-b[n,1]-b[n,2]\*(t-3)-

b[n,3]\*x[n,t])^2/(2\*sigmasq\_y)-0.5\*log(sigmasq\_y);

}

"

smLKJ <- stan\_model(model\_code=growthLKJ.stan)

fitLKJ <- sampling(smLKJ,data=data, iter = 5000,chains = 2,seed= 12345)

summary(fitLKJ, pars = c("mu\_b","R","sigma","Sigma\_b"),probs = c(0.025, 0.975))$summary

# LOO and WAIC

loo(as.matrix(fitLKJ,pars="log\_lik"))

waic(as.matrix(fitLKJ,pars="log\_lik"))

**#**

**# LKJ Prior On Lower Cholesky Factor of Correlation Matrix, Dirichlet Variance Partition**

**#**

phi0=c(1,1,1);

data <- list(y=DS\_10\_1$y, x=DS\_10\_1$x,N=500,T=5,phi0)

growthLKJpart.stan <- "

data {

int<lower=0> N;

int<lower=0> T;

real x[N,T];

real y[N,T];

vector<lower=0>[3] phi0;

}

parameters {

vector[3] b[N];

vector[3] mu\_b;

real<lower=0> sigmasq\_y;

real<lower=0> sigma2\_t;

cholesky\_factor\_corr[3] Lcorr;

simplex[3] phi;

simplex[3] phi\_w;

}

transformed parameters {

real<lower=0> sigma\_y;

vector<lower=0>[3] sigma;

sigma\_y = sqrt(sigmasq\_y);

for (k in 1:3) sigma[k] =sqrt(phi[k]\*sigma2\_t);

}

model {

sigmasq\_y ~ inv\_gamma(0.001, 0.001);

mu\_b ~ normal(0, 100);

phi ~ dirichlet(phi\_w);

phi\_w ~ dirichlet(phi0);

sigma2\_t ~ cauchy(0,5);

Lcorr ~ lkj\_corr\_cholesky(1);

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

for (n in 1:N) for (t in 1:T)

y[n,t] ~ normal(b[n,1]+b[n,2]\*(t-3)+b[n,3]\*x[n,t], sqrt(sigmasq\_y));

}

generated quantities {

matrix[3,3] Omega;

matrix[3,3] Sigma\_b;

matrix[3,3] R;

matrix[N,T] log\_lik;

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

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

for (i in 1:3) for (j in 1:3) R[i,j] = Sigma\_b[i,j]/ sqrt(Sigma\_b[i,i]\*Sigma\_b[j,j]);

for (n in 1:N) for (t in 1:T) log\_lik[n,t] = -(y[n,t]-b[n,1]-b[n,2]\*(t-3)-

b[n,3]\*x[n,t])^2/(2\*sigmasq\_y)-0.5\*log(sigmasq\_y);

}

"

smLKJpart <- stan\_model(model\_code=growthLKJpart.stan)

fitLKJpart <- sampling(smLKJpart, data =data, iter = 5000,warmup=1000,chains = 2,seed= 12345)

summary(fitLKJpart, pars = c("mu\_b","R","sigma","Sigma\_b"),

probs = c(0.025, 0.975))$summary

# LOO and WAIC

loo(as.matrix(fitLKJpart,pars="log\_lik"))

waic(as.matrix(fitLKJpart,pars="log\_lik"))

#

# LKJ PRIOR ON CORRELATION MATRIX

#

growthLKJcor.stan <- "

data {

int<lower=0> N;

int<lower=0> T;

real x[N,T];

real y[N,T];

}

parameters {

vector[3] beta[N];

vector[3] mu\_beta;

real<lower=0> sigmasq\_y;

corr\_matrix[3] Omega;

vector<lower=0>[3] sigma;

}

transformed parameters {

real<lower=0> sigma\_y;

cov\_matrix[3] Sigma\_beta;

vector[3] sig\_beta;

sigma\_y = sqrt(sigmasq\_y);

Sigma\_beta = quad\_form\_diag(Omega, sigma);

for (j in 1:3) sig\_beta[j] = sqrt(Sigma\_beta[j,j]);

}

model {

sigmasq\_y ~ inv\_gamma(0.001, 0.001);

mu\_beta ~ normal(0, 100);

sigma ~ cauchy(0, 5);

Omega ~ lkj\_corr(1); // LKJ prior on correlation matrix

for (n in 1:N)

beta[n] ~ multi\_normal(mu\_beta, Sigma\_beta);

for (n in 1:N)

for (t in 1:T)

# centered x[]

y[n,t] ~ normal(beta[n, 1] + beta[n,2]\*(t-3) + beta[n, 3] \* x[n,t], sqrt(sigmasq\_y));

}

"

smLKJcor <- stan\_model(model\_code=growthLKJcor.stan)

fitLKJcor <- sampling(smLKJcor, data =data, iter = 5000, warmup=1000, chains = 2,seed= 12345)

summary(fitLKJcor, pars = c("sig\_beta","Omega"),probs = c(0.025, 0.975))$summary

# LOO and WAIC

loo(as.matrix(fitLKJcor,pars="log\_lik"))

waic(as.matrix(fitLKJcor,pars="log\_lik"))