setwd("C:/R files BHMRA")

library(R2OpenBUGS)

library(loo)

options(scipen=999)

attach("DS\_10\_3.Rdata")

**# Independent Errors Model**

model1 <- function() { for (i in 1:5400) { y[id[i],time[i]] <- Y[i]}

for (i in 1:n) {# firm-specific AR1 estimates

rh1[i] <- sum(p1[i,2:T])/sum(d2[i,1:T])

rh1.step[i] <- step(rh1[i])

DW[i] <- sum(d1[i,2:T])/sum(d2[i,1:T])

DW.r[i] <- sum(d1.r[i,2:T])/sum(d2.r[i,1:T])

b[i,1:2] ~ dmnorm(B[1:2],D.inv[1:2,1:2])

b.r[i,1:2] ~ dmnorm(B[1:2],D.inv[1:2,1:2])

# RIAS model

for (t in 1:T) {mu[i,t] <- b[i,1]+b[i,2]\*x[t]

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

log\_lik[i,t] <- 0.5\*log(tau/6.28)-0.5\*tau\*pow(y[i,t]-mu[i,t],2)

# full data posterior replicates and residuals

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

e[i,t] <- y[i,t]-mu[i,t];

e.r[i,t] <- y.r[i,t]-mu[i,t]

# mixed predictive replicates

mu.r[i,t] <- b.r[i,1]+b.r[i,2]\*x[t]

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

test.mx[i,t] <- step(y.mx.r[i,t]-y[i,t])}

for (t in 3:T){p2[i,t] <- e[i,t]\*e[i,t-2]}

for (t in 1:T){d2[i,t] <- pow(e[i,t],2)

d2.r[i,t] <- pow(e.r[i,t],2)}

for (t in 2:T){p1[i,t] <- e[i,t]\*e[i,t-1]

d1[i,t] <- pow(e[i,t]-e[i,t-1],2)

d1.r[i,t] <- pow(e.r[i,t]-e.r[i,t-1],2) }}

# mean error autocorrelation (AR1)

rh1.m <- mean(rh1[])

**# posterior predictive check**

DW.m <- mean(DW[])

DW.r.m <- mean(DW.r[])

PPC <- step(DW.m-DW.r.m)

**#priors**

tau ~ dgamma(1,0.001)

D.inv[1:2,1:2] ~ dwish(Wsc[,],2)

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

for (j in 1:2) {sigb[j] <- sqrt(D[j,j])

B[j] ~ dnorm(0,0.001)

for (k in 1:2) {Wsc[j,k] <- equals(j,k)}}}

**# Initial values and Estimation**

inits <- list(list(B=c(0,0),tau=1, D.inv=structure(.Data=c(1,0,0,1),.Dim=c(2,2))),

list(B=c(-0.5,0.7),tau=0.01, D.inv=structure(.Data=c(4,4,4,15),.Dim=c(2,2))))

n.iters=5000; n.burnin =500; n.chains=2

pars <- c("B","rh1.step","rh1.m","PPC","sigb")

M1 =bugs(DS\_10\_3,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F,bugs.seed=10)

M1$summary

**# estimate mixed predictive probabilities**

n.iters=5000; n.burnin =4000; n.chains=2

pars = c("test.mx")

M1.mx = bugs(DS\_10\_3,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg =F,

bugs.seed=10)

mx.means=apply(M1.mx$sims.list$test.mx,c(2,3),mean)

sum(mx.means<0.05)+sum(mx.means>0.95)

**# LOO-IC**

n.iters=5000; n.burnin =4000; n.chains=2

pars = c("log\_lik")

M1.LL = bugs(DS\_10\_3,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

loo(matrix(as.array(M1.LL$sims.list$log\_lik),2000,90\*60))

**# AR1 Errors Model**

model2 <- function() { for (i in 1:5400) { y[id[i],time[i]] <- Y[i]}

for (i in 1:n) { b[i,1:2] ~ dmnorm(B[1:2],D.inv[1:2,1:2])

# firm-specific AR1 estimates

rh1[i] <- sum(p1[i,2:T])/sum(d2[i,1:T])

rh1.step[i] <- step(rh1[i])

for (t in 1:T) {mu[i,t] <- b[i,1]+b[i,2]\*x[t]

d2[i,t] <- pow(e[i,t],2)}

y[i,1] ~ dnorm(mu[i,1],tau1)

e[i,1] <- y[i,1]-mu[i,1]

log\_lik[i,1] <- 0.5\*log(tau1/6.28)-0.5\*tau1\*pow(y[i,1]-mu[i,1],2)

for (t in 2:T) {y[i,t] ~ dnorm(nu[i,t],tau)

e[i,t] <- y[i,t]-nu[i,t]

nu[i,t] <- rho\*y[i,t-1]+mu[i,t]-rho\*mu[i,t-1]

log\_lik[i,t] <- 0.5\*log(tau/6.28)-0.5\*tau\*pow(y[i,t]-nu[i,t],2)

p1[i,t] <- e[i,t]\*e[i,t-1] }}

# variance component estimates

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

for (j in 1:2) { sigb[j] <- sqrt(D[j,j]);

for (k in 1:2) { R[j,k] <- D[j,k]/sqrt(D[j,j]\*D[k,k])}}

# mean error autocorrelation (AR1)

rh1.m <- mean(rh1[])

# Priors

for (j in 1:2) {B[j] ~ dnorm(0,0.001)

for (k in 1:2) {Wsc[j,k] <- equals(j,k)}}

D.inv[1:2,1:2] ~ dwish(Wsc[,],2)

tau ~ dgamma(1,0.001)

rho ~ dunif(-1,1)

# inverse of error variance in period 1

tau1<- (1-rho\*rho)\*tau}

**# Initial values and Estimation**

inits <- list(list(B=c(0,0),tau=1, rho=0,D.inv=structure(.Data=c(1,0,0,1),.Dim=c(2,2))),

list(B=c(-0.5,0.7),tau=0.01, rho=0.1,D.inv=structure(.Data=c(4,4,4,15),.Dim=c(2,2))))

n.iters=5000; n.burnin =500; n.chains=2

pars = c("B","sigb","rho","rh1.step","rh1.m")

M2 = bugs(DS\_10\_3,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

M2$summary

**# LOO-IC**

n.iters=5000; n.burnin =4000; n.chains=2

pars = c("log\_lik")

M2.LL = bugs(DS\_10\_3,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

loo(matrix(as.array(M2.LL$sims.list$log\_lik),2000,90\*60))

**# Varying AR1 Errors Model**

model3 <- function() { for (i in 1:5400) { y[id[i],time[i]] <- Y[i]}

for (i in 1:n) { delta[i] ~ dbeta(delta.a,delta.b)

rho[i] <- 2\*delta[i]-1

# inverse of error variance in period 1

tau1[i] <- (1-rho[i]\*rho[i])\*tau

b[i,1:2] ~ dmnorm(B[1:2],D.inv[1:2,1:2])

# firm-specific AR1 estimates

rh1[i] <- sum(p1[i,2:T])/sum(d2[i,1:T])

rh1.step[i] <- step(rh1[i])

for (t in 1:T) {mu[i,t] <- b[i,1]+b[i,2]\*x[t]

d2[i,t] <- pow(e[i,t],2)}

y[i,1] ~ dnorm(mu[i,1],tau1[i])

e[i,1] <- y[i,1]-mu[i,1]

log\_lik[i,1] <- 0.5\*log(tau1[i]/6.28)-0.5\*tau1[i]\*pow(y[i,1]-mu[i,1],2)

for (t in 2:T) {y[i,t] ~ dnorm(nu[i,t],tau)

e[i,t] <- y[i,t]-nu[i,t]

nu[i,t] <- rho[i]\*y[i,t-1]+mu[i,t]-rho[i]\*mu[i,t-1]

log\_lik[i,t] <- 0.5\*log(tau/6.28)-0.5\*tau\*pow(y[i,t]-nu[i,t],2)

p1[i,t] <- e[i,t]\*e[i,t-1] }}

# variance component estimates

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

for (j in 1:2) { sigb[j] <- sqrt(D[j,j]);

for (k in 1:2) { R[j,k] <- D[j,k]/sqrt(D[j,j]\*D[k,k])}}

# mean error autocorrelation (AR1)

rho.m <- mean(rho [])

# Priors

for (j in 1:2) {B[j] ~ dnorm(0,0.001)

for (k in 1:2) {Wsc[j,k] <- equals(j,k)}}

D.inv[1:2,1:2] ~ dwish(Wsc[,],2)

tau ~ dgamma(1,0.001)

delta.a ~ dexp(1)

delta.b ~ dexp(1) }

**# Initial values and Estimation**

inits <- list(list(B=c(0,0),delta.a=2,delta.b=2,

tau=1,D.inv=structure(.Data=c(1,0,0,1),.Dim=c(2,2))),

list(B=c(-0.5,0.7),tau=0.01,delta.a=1,delta.b=1,

D.inv=structure(.Data=c(4,4,4,15),.Dim=c(2,2))))

n.iters=5000; n.burnin =500; n.chains=2

pars = c("B","sigb","rho","rho.m","rh1.step")

M3 = bugs(DS\_10\_3,inits,pars,n.iters,model3,n.chains, n.burnin,debug=T,codaPkg = F,bugs.seed=10)

**# LOO-IC**

n.iters=5000; n.burnin =4000; n.chains=2

pars = c("log\_lik")

M3.LL = bugs(DS\_10\_3,inits,pars,n.iters,model3,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

loo(matrix(as.array(M3.LL$sims.list$log\_lik),2000,90\*60))