setwd("C:/R files BHMRA")

library(R2OpenBUGS); library(mcmcplots); library(loo); library(rstan)

options(scipen=999)

attach("DS\_10\_2.Rdata")

**# Regression for mean and covariances**

model1 <- function() {for (i in 1:n) { for (t in 1:T) {

y[i,t] ~ dnorm(nu[i,t],inv.d[i,t]);

log\_lik[i,t] <- -0.5\*log(d[i,t]\*6.28)-0.5\*inv.d[i,t]\*pow(y[i,t]-mu[i,t],2)

log(d[i,t]) <- gam[1]+gam[2]\*t+gam[3]\*t\*t;

inv.d[i,t] <- 1/d[i,t]

mu[i,t] <- beta[1]+beta[2]\*t+beta[3]\*t\*t+beta[4]\*t\*t\*t

# predictions

ynew[i,t] ~ dnorm(nu[i,t],inv.d[i,t])

predexc[i,t] <- step(ynew[i,t]-y[i,t])}

# initial condition

nu[i,1] <- mu[i,1]

for (t in 2:T) {nu[i,t] <- mu[i,t]+sum(ph.prod[i,t,1:t-1])

for (j in 1:t-1) {ph[i,t,j] <- lam[1]+lam[2]\*(t-j)+lam[3]\*(t-j)\*(t-j)

ph.prod[i,t,j] <- ph[i,t,j]\*(y[i,j]-mu[i,j])}}}

# priors

for (j in 1:3) {lam[j] ~ dnorm(0,0.001); gam[j] ~ dnorm(0,0.001)}

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

# global fit

TLL <- sum(log\_lik[,]); BIC <- -2\*TLL+npar\*n\*T}

# estimate hyperparameters

inits <- list(list(beta=c(94,0.8,-0.02,0),gam=c(0.3,-0.2,0),lam=c(0.4,-0.1,0)),

list(beta=c(95,1.1,0,0),gam=c(1.3,0,0),lam=c(0.4,-0.05,0)))

n.iters=10000; n.burnin =1000; n.chains=2

pars <- c("beta","gam","lam","predexc","TLL")

M1 <- bugs(DS\_10\_2,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

n.iters=10000; n.burnin =1000; n.chains=2

pars <- c("log\_lik")

M1LL <- bugs(DS\_10\_2,inits,pars,n.iters,model1,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

loo(matrix(as.array(M1LL$sims.list$log\_lik),18000,6\*24))

**# Wishart Prior**

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

mu[i,t] <- b[i,1]+b[i,2]\*t+b[i,3]\*t\*t

# predictions

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

predexc[i,t] <- step(ynew[i,t]-y[i,t])}

# random effects

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

tau ~ dgamma(1,0.001); sig2 <- 1/tau

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);

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

TLL <- sum(log\_lik[,])}

**# Estimate hyperparameters**

ini1 <- list(B=c(0,0,0),inv.D=structure(.Data=c(100,0,0, 0,100,0, 0,0,100),.Dim=c(3,3)))

ini2 <- list(B=c(50,0,0),inv.D=structure(.Data=c(1000,0,0, 0,1000,0, 0,0,1000),.Dim=c(3,3)))

inits = list(ini1,ini2)

n.iters=10000; n.burnin =1000; n.chains=2

pars <- c("B","TLL")

M2 <- bugs(DS\_10\_2,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = F,bugs.seed=10)

M2$summary

**# obtain LOO-IC**

pars <- c("log\_lik")

n.iters=10000; n.burnin =1000; n.chains=2

M2LL <- bugs(DS\_10\_2,inits,pars,n.iters,model2,n.chains, n.burnin,debug=T,codaPkg = F, bugs.seed=10)

loo(matrix(as.array(M2LL$sims.list$log\_lik),18000,6\*24))

X=matrix(,24,3)

X[,1]=1

X[,2]=seq(1:24)

X[,3]=seq(1:24)^2

DS\_10\_2$X=X

DS\_10\_2$N=DS\_10\_2$n

# Model 3

model3 = "

functions { matrix cov\_matrix\_ar1(real ar, real sigma, int nrows) {

matrix[nrows, nrows] mat;

vector[nrows - 1] gamma;

mat = diag\_matrix(rep\_vector(1, nrows));

for (i in 2:nrows) { gamma[i - 1] = pow(ar, i - 1);

for (j in 1:(i - 1)) { mat[i, j] = gamma[i - j];

mat[j, i] = gamma[i - j]; } }

return sigma^2 / (1 - ar^2) \* mat; } }

data { int<lower=1> T; // total number of times

int<lower=1> N; // subjects

matrix[N,T] y; // response variable

matrix[T,3] X;

}

transformed data { vector[T] se2 = rep\_vector(0, T); }

parameters {

vector[3] beta;

real<lower=0> sigma; // residual SD

real <lower=-1,upper=1> phi; }

model { matrix[T, T] res\_cov\_matrix;

matrix[T, T] Sigma;

vector[T] mu = X\*beta;

res\_cov\_matrix = cov\_matrix\_ar1(phi, sigma, T);

Sigma = res\_cov\_matrix + diag\_matrix(se2);

Sigma = cholesky\_decompose(Sigma);

// priors including all constants

beta ~ student\_t(3,0,10);

sigma ~ cauchy(0,5);

for (i in 1:N) {y[i,] ~ multi\_normal\_cholesky(mu,Sigma); } }

"

fit3 = stan(model\_code=model3, data= DS\_10\_2, iter=2000, chains=2, seed="1234")

summary(fit3, pars = c("beta","sigma","phi"), probs = c(0.025,0.5, 0.975))$summary

# Model 4 (Alternative Coding of Model 3)

model4 = "

data {

int<lower=1> T; // total number of times

int<lower=1> N; // subjects

matrix[N,T] y;

matrix[T,3] X;

}

parameters {

vector[3] beta;

real<lower=0> sigma;

real<lower=-1,upper=1> phi;

}

transformed parameters {

matrix[N,T] mu;

matrix[N,T] epsilon;

for (i in 1:N) {for (t in 1:T) { mu[i,t] = X[t,]\*beta;}}

for (i in 1:N) {epsilon[i,1] = y[i,1] - mu[i,1];

for (t in 2:T) {epsilon[i,t] = (y[i,t] - mu[i,t]);

mu[i,t] = mu[i,t] + phi\*epsilon[i,t-1]; } }}

model {

phi ~ uniform(-1,1);

beta ~ normal(0,100);

sigma ~ cauchy(0,5);

for (i in 1:N) {for (t in 1:T) {y[i,t] ~ normal(mu[i,t], sigma); }}}

generated quantities {

matrix[N,T] log\_lik;

for (i in 1:N) {for (t in 1:T) {log\_lik[i,t] =normal\_lpdf(y[i,t]|mu[i,t], sigma); }}}

"

fit4 = stan(model\_code=model4, data= DS\_10\_2, iter=2000, chains=2, seed="1234")

summary(fit4, pars = c("beta","sigma","phi"), probs = c(0.025,0.5, 0.975))$summary

LLsamps <- as.matrix(fit4,pars="log\_lik")

loo(as.array(LLsamps))