require(jagsUI)

require(MCMCvis)

require(loo)

D <- list(T=100,Y=c(1120,1160,963,1210,1160,1160,813,1230,1370,1140,995,935,1110,994,1020,

960,1180,799,958,1140,1100,1210,1150,1250,1260,1220,1030,1100,774,840,874,694,940,833,701,

916,692,1020,1050,969,831,726,456,824,702,1120,1100,832,764,821,768,845,864,862,698,845,

744,796,1040,759,781,865,845,944,984,897,822,1010,771,676,649,846,812,742,801,1040,860,874,

848,890,744,749,838,1050,918,986,797,923,975,815,1020,906,901,1170,912,746,919,718,714,740))

attach(D)

y <- numeric(100)

for (t in 1:100) {y[t] <- Y[t]-Y[1]}

D=list(T=100,y=y)

# Classical Time Point Estimate (at most one changepoint)

require(changepoint)

yts <- ts(y, start=c(1871), end=c(1970), frequency=1)

cptm <- cpt.mean(yts,method='BinSeg',Q=1)

#

**# AR(2) Model**

#

**cat("**model { for (t in 1:T) { LL[t] <- -0.92+0.5\*log(tau)-0.5\*tau\*(y[t]-mu[t])\*(y[t]-mu[t])

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

for (t in 3:T) {mu[t] <- phi0+phi[1]\*y[t-1]+phi[2]\*y[t-2]}

mu[2] <- phi0+phi[1]\*y[1]+phi[2]\*y0

mu[1] <- phi0+phi[1]\*y0+phi[2]\*ym1

# initial conditions

pre.mu0 <- phi0+eps[1]

pre.mu1 <- phi0+eps[2]

y0 ~ dt(pre.mu0,tau,2)

ym1 ~ dt(pre.mu1,tau,2)

# other priors

phi0 ~ dnorm(0,0.001)

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

eps[j] ~ dnorm(0,1)}

tau ~ dgamma(1,0.001) }

", file="model1.jag")

**# Initial Values and Estimation**

inits1 <- list(phi0=0,phi=c(0,0),tau=1,y0=0,ym1=0)

inits2 <- list(phi0=0,phi=c(0.5,0.5),tau=0.1,y0=0,ym1=0)

inits=list(inits1,inits2)

pars <- c("mu","phi","LL")

R1 <- autojags(D, inits, pars,model.file="model1.jag",2,iter.increment=5000, n.burnin=500,Rhat.limit=1.1, max.iter=100000, seed=1234,codaOnly=c("LL"))

R1$summary

loo(as.matrix(R1$sims.list$LL))

#

**# AR(2) with shift year**

#

**cat("**model { for (t in 1:T) {y[t] ~ dnorm(mu[t],tau)

LL[t] <- -0.92+0.5\*log(tau)-0.5\*tau\*(y[t]-mu[t])\*(y[t]-mu[t])}

for (t in 3:T) { mu[t] <- phi0[1]+phi0[2]\*step(t-kappa) +d[1]\*phi[1]\*y[t-1] +d[2]\*phi[2]\*y[t-2]}

mu[2] <- phi0[1]+d[1]\*phi[1]\*y[1]+d[2]\*phi[2]\*y0

mu[1] <- phi0[1]+d[1]\*phi[1]\*y0+d[2]\*phi[2]\*ym1

# initial conditions

pre.mu0 <- phi0[1]+eps[1]

pre.mu1 <- phi0[1]+eps[2]

y0 ~ dt(pre.mu0,tau,2)

ym1 ~ dt(pre.mu1,tau,2)

# prior on shift year

kappa ~ dunif(3,97)

for (j in 1:2) {

# alternative intercepts

phi0[j] ~ dnorm(0,0.001)

eps[j] ~ dnorm(0,1)

d[j] ~ dbern(pi.d);

phi[j] ~ dnorm(0,1)}

# prior on retention probability

pi.d ~ dbeta(1,1)

tau ~ dgamma(1,0.001)}

", file="model2.jag")

**# Initial Values and Estimation**

inits1 <- list(phi0=c(0,0),phi=c(0,0),tau=1,y0=0,ym1=0,eps=c(0,0))

inits2 <- list(phi0=c(0,0),phi=c(0.5,0.5),tau=0.1,y0=0,ym1=0,eps=c(0,0))

inits=list(inits1,inits2)

pars <- c("mu","phi","d","kappa","LL","phi0")

R2 <- autojags(D, inits, pars,model.file="model2.jag",2,iter.increment=25000, n.burnin=500,Rhat.limit=1.1, max.iter=500000, seed=1234,codaOnly=c("LL"))

R2$summary

loo(as.matrix(R2$sims.list$LL))

#

**# AR(1) with shift year**

#

**cat("**model { for (t in 1:T) {y[t] ~ dnorm(mu[t],tau)

LL[t] <- -0.92+0.5\*log(tau)-0.5\*tau\*(y[t]-mu[t])\*(y[t]-mu[t])}

for (t in 2:T) { mu[t] <- phi0[1]+phi0[2]\*step(t-kappa) +phi1\*y[t-1]}

mu[1] <- phi0[1]+phi1\*y0

# initial condition

pre.mu0 <- phi0[1]+eps1

y0 ~ dt(pre.mu0,tau,2)

eps1 ~ dnorm(0,1)

# prior on shift year

kappa ~ dunif(3,97)

for (j in 1:2) {# alternative intercepts

phi0[j] ~ dnorm(0,0.001) }

phi1 ~ dnorm(0,1)

tau ~ dgamma(1,0.001)}

", file="model3.jag")

**# Initial Values and Estimation**

inits1 <- list(phi0=c(0,0),phi1=0,tau=1,y0=0,eps1=0)

inits2 <- list(phi0=c(0,0),phi1=0.5,tau=0.1,y0=0,eps1=0)

inits=list(inits1,inits2)

pars <- c("mu","phi1","kappa","LL","phi0")

R3 <- autojags(D, inits, pars,model.file="model3.jag",2,iter.increment=5000, n.burnin=500,Rhat.limit=1.1, max.iter=500000, seed=1234)

R3$summary

loo(as.matrix(R3$sims.list$LL))

#

**# AR(2) with SETAR shift**

#

**cat("**model { for (t in 1:T) {y[t] ~ dnorm(mu[t],tau)

LL[t] <- -0.92+0.5\*log(tau)-0.5\*tau\*(y[t]-mu[t])\*(y[t]-mu[t])}

for (t in 3:T) { mu[t] <- phi0[1]+phi0[2]\*step(y[t-1]-kappa.y) +phi[1]\*y[t-1] +phi[2]\*y[t-2]}

mu[2] <- phi0[1]+phi[1]\*y[1]+phi[2]\*y0

mu[1] <- phi0[1]+phi[1]\*y0+phi[2]\*ym1

# initial conditions

pre.mu0 <- phi0[1]+eps[1]

pre.mu1 <- phi0[1]+eps[2]

y0 ~ dt(pre.mu0,tau,2)

ym1 ~ dt(pre.mu1,tau,2)

# prior on shift in discharge value (differenced data)

kappa.y ~ dunif(-700,300)

for (j in 1:2) {phi0[j] ~ dnorm(0,0.001)

eps[j] ~ dnorm(0,1)

phi[j] ~ dnorm(0,1)}

tau ~ dgamma(1,0.001)}

", file="model4.jag")

**# Initial Values and Estimation**

inits1 <- list(phi0=c(0,0),phi=c(0,0),tau=1,y0=0,ym1=0,kappa.y=-415)

inits2 <- list(phi0=c(0,0),phi=c(0.5,0.5),tau=0.1,y0=0,ym1=0,kappa.y=-420)

inits=list(inits1,inits2)

pars <- c("mu","phi","d","kappa.y","LL","phi0")

R4 <- autojags(D, inits, pars,model.file="model4.jag",2,iter.increment=10000, n.burnin=500,Rhat.limit=1.05, max.iter=100000, seed=1234)

R4$summary

loo(as.matrix(R4$sims.list$LL))

**# prior posterior overlap for SETAR shift**

PR <- runif(20000, -700, 300)

MCMCtrace(R4, params = 'kappa.y', priors = PR, pdf = FALSE)