options(scipen=999)

library(R2OpenBUGS)

library(loo)

library(rstan)

library(jagsUI)

library(runjags)

library(coda)

library(MCMCvis)

**# Regression Data**

library(heavy)

data(ereturns)

x=as.vector(ereturns[[4]])

y=as.vector(ereturns[[3]])

**# number of data points**

n=60

x.new=0.13

**# Data for BUGS and JAGS**

D=list(y=y,x=x,n=n,x.new=x.new)

**#**

**# BUGS code (run using R2OpenBUGS)**

**#**

model <- function() { for (i in 1:n) {y[i] ~dnorm(mu[i],tau)

mu[i] <- beta[1] + beta[2]\*(x[i]-mean(x[]))

# log-likelihood

LL[i] <- -0.92+0.5\*log(tau)-0.5\*tau\*pow(y[i]-mu[i],2)

# replicates at observed x[i]

yrep[i] ~dnorm(mu[i],tau)

# check replicate against actual observation

check[i] <- step(yrep[i]-y[i])}

# priors

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

# calculate precision

tau <- 1/(sigma\*sigma)

sigma ~ dunif(0,100)

# prediction at new x value

mu.new <- beta[1]+beta[2]\*(x.new-mean(x[]))

y.new ~dnorm(mu.new,tau)}

# Estimation

inits1 = list(beta=rep(0,2), sigma=1)

inits2 = list(beta=rep(0,2), sigma=2)

inits=list(inits1,inits2)

pars = c("beta","sigma","check","y.new","LL")

n.iters=10000; n.burnin =500; n.chains=2

**# Using coda (codaPkg=T)**

R = bugs(D,inits,pars,n.iters,model,n.chains, n.burnin,debug=T,codaPkg = T,bugs.seed=10)

R.out=read.bugs(R)

# posterior summary and densities

MCMCsummary(R.out)

# densityplot(R.out)

**# Not using coda (codaPkg=F)**

R = bugs(D,inits,pars,n.iters,model,n.chains, n.burnin,debug=T,codaPkg = F,bugs.seed=10)

# Posterior Summary

R$summary

# Fit

LOO=loo(R$sims.list$LL)

LOO.PW.BUGS=LOO$pointwise[,3]

order(LOO.PW.BUGS)

#

**# JAGS code (using jagsUI)**

#

cat("

model {for (i in 1:n) {y[i] ~dnorm(mu[i], 1/sigma^2)

mu[i] = beta[1] + beta[2]\*(x[i]-mean(x[]))

# log-likelihood

LL[i] = logdensity.norm(y[i],mu[i],1/sigma^2)

# replicates at observed x[i]

yrep[i] ~dnorm(mu[i],1/sigma^2)

# check replicate against actual observation

check[i] = step(yrep[i]-y[i])}

# priors

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

sigma ~ dunif(0,100)

# prediction at new x value

mu.new = beta[1]+beta[2]\*(x.new-mean(x[]))

y.new ~dnorm(mu.new, sigma^2 )}

", file="model.jag")

**# Estimation**

inits <- function(){list(sigma=runif(0,5), beta=rnorm(2,0,0.1))}

pars = c("beta","sigma","check","y.new","LL")

R = autojags(D, inits, pars,model.file="model.jag",2,iter.increment=1000, n.burnin=100,Rhat.limit=1.025, max.iter=5000, seed=1234, codaOnly= c('LL'))

**# Posterior Summary**

R$summary

**# Posterior Densities**

# plot(R)

**# Fit**

LOO=loo(as.matrix(R$sims.list$LL))

LOO.PW.JAGS=LOO$pointwise[,3]

order(LOO.PW.JAGS)

#

**# JAGS code (using runjags)**

#

model ="model {for (i in 1:n) {y[i] ~dnorm(mu[i], 1/sigma^2)

mu[i] = beta[1] + beta[2]\*(x[i]-mean(x[]))

# log-likelihood

LL[i] = logdensity.norm(y[i],mu[i],1/sigma^2)

# replicates at observed x[i]

yrep[i] ~dnorm(mu[i],1/sigma^2)

# check replicate against actual observation

check[i] = step(yrep[i]-y[i])}

# priors

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

sigma ~ dunif(0,100)

# prediction at new x value

mu.new = beta[1]+beta[2]\*(x.new-mean(x[]))

y.new ~dnorm(mu.new, sigma^2 )} "

inits <- function(){list(sigma=runif(0,5), beta=rnorm(2,0,0.1))}

pars = c("beta","sigma","check","y.new","LL")

R = autorun.jags(model,data=D,startburnin=500,startsample=4000, inits=inits,

monitor=pars ,n.chains=2)

add.summary(R)

**# MCMC output for log-likelihoods**

LLsamps=as.matrix(as.mcmc.list(R, vars = "LL"))

LOO=loo(LLsamps)

LOO.PW.JAGS=LOO$pointwise[,3]

order(LOO.PW.JAGS)

#

**# rstan, Data and Code**

#

K=2

X=matrix(,60,K)

X[,1]=1

X[,2]=x-mean(x)

x\_new=x.new-mean(x)

D=list(y=y,X=X,n=60,K=2,x\_new=x\_new)

model="

data {

int n; // number of observations

real y[n]; // response

real x\_new; // new predictor value

int K; // number of predictors

matrix[n,K] X; // predictor matrix

}

parameters {

vector[K] beta; // regression coefficients

real <lower=0> sigma; // residual standard deviation

}

transformed parameters {

vector[n] eta; // linear regression term

eta = X\*beta;

}

model {

sigma ~ uniform(0,100);

beta ~ normal(0,31.6);

y ~ normal(eta,sigma);

}

generated quantities { real LL[n];

real y\_rep[n];

real y\_new;

real check[n];

for (i in 1:n) {LL[i]= normal\_lpdf(y[i] |eta[i],sigma); }

for (i in 1:n) {y\_rep[i] =normal\_rng(eta[i],sigma);}

for (i in 1:n) {check[i] =step(y\_rep[i]-y[i]);}

y\_new = normal\_rng(beta[1]+beta[2]\*x\_new,sigma); // prediction at new x value

}

"

**# Estimation**

fit=stan(model\_code = model,data=D, iter = 2000,warmup = 250,chains=2)

# Posterior Summary

print(fit,digits=3)

# plot of posterior densities

# stan\_dens(fit)

# Fit

LLsamps <- as.matrix(fit,pars="LL")

LOO=loo(LLsamps)

LOO.PW.STAN= LOO$pointwise[,3]

order(LOO.PW.STAN)

**# Alternative Stan Code using target +**

model="

data {

int n; // number of observations

real y[n]; // response

real x\_new; //new predictor value

int K; // number of predictors

matrix[n,K] X; // predictor matrix

}

parameters {

vector[K] beta; // regression coefficients

real <lower=0> sigma; // residual standard deviation

}

transformed parameters {

vector[n] eta; // linear regression term

eta = X\*beta;

}

model {

target += uniform\_lpdf(sigma|0,100);

target += normal\_lpdf(beta | 0, 31.6);

target += normal\_lpdf(y | eta, sigma);

}

generated quantities { real LL[n];

real y\_rep[n];

real y\_new;

real check[n];

for (i in 1:n) {LL[i]= normal\_lpdf(y[i] |eta[i],sigma); }

for (i in 1:n) {y\_rep[i] =normal\_rng(eta[i],sigma);}

for (i in 1:n) {check[i] =step(y\_rep[i]-y[i]);}

y\_new = normal\_rng(beta[1]+beta[2]\*x\_new,sigma); // prediction at new x value

}

"

**# Estimation**

fit=stan(model\_code = model,data=D, iter = 2000,warmup = 250,chains=2)

# Posterior Summary

print(fit,digits=3)