setwd("C:/R files BHMRA")

require(jagsUI); require(normtest); require(runjags)

require(rube) ; options(scipen=999)

library(loo)

Sys.setenv(BUGSDIR="c:\\users\\p congdon\\documents\\WINBUGS14")

attach("DS\_4\_2.Rdata")

**# Normal- Normal model (jagsUI)**

cat(" model { ynew ~ dnorm(b.new,inv.s20);

b.new ~ dnorm(mu,1/tau2);

# treatment benefit, replicate data

OR.new <- exp(ynew)

for (j in 1:J) { y[j] ~ dnorm(b[j],inv.s2[j]);

# second stage

b[j] ~ dnorm(mu,1/tau2)

# mixed replicates

yrep[j] ~ dnorm(brep[j],inv.s2[j]);

brep[j] ~ dnorm(mu,1/tau2)

exceed[j] <- step(yrep[j]-y[j])

# log-likelihood

log(L[j]) <- -0.5\*log(6.28\*s2[j])-0.5\*pow(y[j]-b[j],2)/s2[j]

LL[j] <- log(L[j])

# marginal density log-likelihood

log(MDL[j]) <- -0.5\*log(6.28\*(s2[j]+tau2))-0.5\*pow(y[j]-b[j],2)/(s2[j]+tau2)

MDLL[j] <- log(MDL[j])}

# uniform shrinkage prior

s2.0 <- 90/sum(inv.s2[1:J])

inv.s20 <- 1/s2.0

tau2.S <- (s2.0-s2.0\*w)/w

w ~ dunif(0,1);

# half-Cauchy

tau.C ~ dt(0, 0.16, 1) T(0,)

tau2.C <- tau.C\*tau.C

# select shrinkage (tau2.S) or Cauchy (tau2.C) options for second stage variance

tau2 <- tau2.C

# prior on second stage mean

mu ~ dnorm(0,0.01);

# treatment benefit, observed data

OR.NRT <- exp(mu)

TMDLL <- sum(MDLL)}

", file="model1.jag")

**# initial values and estimation**

inits <- function(){list(b=rnorm(90,0,0.1),mu=rnorm(1,0,0.1))}

pars <- c("OR.NRT","tau2","b","OR.new","TMDLL","exceed","LL")

R1 <- autojags(DS\_4\_2, inits, pars, model.file="model1.jag",2, n.adapt=100, iter.increment=1000, n.burnin=500,Rhat.limit=1.1, max.iter=5000,seed=1234,codaOnly=c("LL"))

R1$summary

**# Extract MCMC Samples, posterior mean bj**

b.mn <- apply(as.matrix(R1$sims.list$b),2,mean)

jb.norm.test(b.mn, nrepl=2000)

shapiro.test(b.mn)

**# Measuring Fit**

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

WAIC=waic(as.matrix(R1$sims.list$LL))

waic=WAIC$pointwise[,3]

**# largest WAIC components (least well fitted)**

subj <- seq(1:90)

list.waic <- data.frame(waic,subj)

list.waic=list.waic[order(-list.waic$waic),]

head(list.waic,10)

**# Model 2, Student t second stage**

cat(" model { ynew ~ dnorm(b.new,1/s2.0)

b.new ~ dnorm(mu,1/tau2)

OR.new <- exp(ynew)

for (j in 1:J) { y[j] ~ dnorm(b[j],inv.s2[j])

b[j] ~ dnorm(mu, lam[j]/tau2)

# mixed replicates

yrep[j] ~ dnorm(brep[j],inv.s2[j]);

brep[j] ~ dnorm(mu,lam[j]/tau2)

exceed[j] <- step(yrep[j]-y[j])

# log-likelihood

log(L[j]) <- -0.5\*log(6.28\*s2[j])-0.5\*pow(y[j]-b[j],2)/s2[j]

LL[j] <- log(L[j])

# marginal density log-likelihood

log(MDL[j]) <- -0.5\*log(6.28\*(s2[j]+tau2/lam[j]))-0.5\*pow(y[j]-b[j],2)/(s2[j]+tau2/lam[j])

MDLL[j] <- log(MDL[j])

# Student t scale mixture terms

lam[j] ~ dgamma(nu/2,nu/2)

step.lam[j] <- step(1-lam[j])}

# uniform shrinkage prior

s2.0 <- J/sum(inv.s2[])

tau2 <- (s2.0-s2.0\*w)/w

w ~ dunif(0,1)

mu ~ dnorm(0,0.01);

# student t hyperparameters

nu ~ dexp(kap)

kap ~ dunif(0.01,0.5)

NRTpos <- step(mu)

OR.NRT <- exp(mu)

TMDLL <- sum(MDLL)}

", file="model2.jag")

**# initial values and estimation**

inits = function(){list(b=rnorm(90,0,0.1),mu=rnorm(1,0,0.1),nu=rexp(1,0.1))}

pars = c("OR.NRT","tau2","nu","OR.new","step.lam","TMDLL","exceed","LL")

R2 = autojags(DS\_4\_2, inits, pars, model.file="model2.jag",2, n.adapt=100, iter.increment=5000, n.burnin=500,Rhat.limit=1.1, max.iter=10000,seed=1234,codaOnly= c("LL"))

R2$summary

**# Fit**

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

WAIC=waic(as.matrix(R2$sims.list$LL))

waic=WAIC$pointwise[,3]

**# largest WAIC components (least well fitted)**

list.waic <- data.frame(waic,subj)

list.waic=list.waic[order(-list.waic$waic),]

head(list.waic,10)

**# MODEL 3, Skew t (rube)**

model3= "model { ynew ~ dnorm(b.new,s2.0.inv)

b.new ~ dnorm(mu,tau2.inv)

OR.new <- exp(ynew)

for (j in 1:J) { y[j] ~ dnorm(b[j],inv.s2[j])

b[j] ~ dnorm(mu, prec.b[j])

prec.b[j] <- f.scale[j]/tau2

# mixed replicates

yrep[j] ~ dnorm(brep[j],inv.s2[j]);

brep[j] ~ dnorm(mu,prec.b[j])

exceed[j] <- step(yrep[j]-y[j])

# Student t scale mixture terms

lam[j] ~ dgamma(nu.2,nu.2)

step.lam[j] <- step(1-lam[j])

# composite scale mixture

f.scale[j] <- lam[j]\*skew.scale[j]

# asymmetric skew scaling

skew.scale[j] <- step(y[j]-mu)/gam2+step(mu-y[j])\*gam2

# log-likelihood

log(L[j]) <- 0.5\*log(inv.s2[j]/6.28)-0.5\*inv.s2[j]\*pow(y[j]-b[j],2)

LL[j] <- log(L[j])

# marginal density log-likelihood

log(MDL[j]) <- -0.5\*log(6.28\*(s2[j]+tau2/f.scale[j]))-

0.5\*pow(y[j]-b[j],2)/(s2[j]+tau2/f.scale[j])

MDLL[j] <- log(MDL[j])}

s2.0 <- J/sum(inv.s2[])

s2.0.inv <- 1/s2.0

tau2 <- (s2.0-s2.0\*w)/w

tau2.inv <- 1/tau2

w ~ dunif(0,1)

mu ~ dnorm(0,0.01)

nu ~ dexp(kap)

nu.2 <- nu/2

# prior on gam2 as in Lee and Thompson (2008)

gam2 ~ dgamma(0.5,0.318)

step.gam2 <- step(gam2-1)

kap ~ dunif(0.01,0.5)

NRTpos <- step(mu)

OR.NRT <- exp(mu)

TMDLL <- sum(MDLL[])}

"

# initial values and estimation

inits = function(){list(b=rnorm(90,0,0.1),mu=rnorm(1,0,0.1),nu=rexp(1,0.1), gam2=rexp(1,1))}

C3 = rube(model3, DS\_4\_2, inits)

summary(C3)

pars = c("OR.NRT","tau2","nu","gam2","step.gam2","OR.new", "TMDLL", "exceed","step.lam")

R3 = rube(model3, DS\_4\_2, inits, pars, n.burn=500, n.thin=1, n.chains=2,n.iter=5000)

summary(R3,limit=90)

# Fit

pars = c("L","LL")

R3.LL = rube(model3, DS\_4\_2, inits, pars, n.burn=500, n.thin=1, n.chains=2,n.iter=5000)

c1=log(apply(R3.LL$sims.list$L,2,mean))

c2=apply(R3.LL$sims.list$LL,2,sd)^2

waic = -2\*(c1-c2)

WAIC = sum(waic)

# largest WAIC components (least well fitted)

list.waic = data.frame(waic,subj)

list.waic=list.waic[order(-list.waic$waic),]

head(list.waic,10)

**# Model 4 Discrete Mixture on Second Stage Variance**

model4= "model { ynew ~ dnorm(b.new,inv.s20);

b.new ~ dnorm(mu,inv.tau2[1]);

OR.new <- exp(ynew)

for (j in 1:J) { y[j] ~ dnorm(b[j],inv.s2[j]);

b[j] ~ dnorm(mu,inv.tau2[G[j]])

# mixed replicates

yrep[j] ~ dnorm(brep[j],inv.s2[j]);

brep[j] ~ dnorm(mu, inv.tau2[G[j]])

exceed[j] <- step(yrep[j]-y[j])

# latent groups

G[j] ~ dcat(p.G[1:2])

G2[j] <- equals(G[j],2)

# log-likelihood

log(L[j]) <- 0.5\*log(inv.s2[j]/6.28)-0.5\*inv.s2[j]\*pow(y[j]-b[j],2)

LL[j] <- log(L[j])

# marginal density log-likelihood

log(MDL[j]) <- -0.5\*log(6.28\*(s2[j]+tau2[G[j]]))-0.5\*pow(y[j]-b[j],2)/(s2[j]+tau2[G[j]])

MDLL[j] <- log(MDL[j])}

# prior discrete mixture probabilities

p.G[1] <- 0.95

p.G[2] <- 0.05

for (j in 1:2) {w.G[j] <- 0.5}

# uniform shrinkage prior

s2.0 <- 90/sum(inv.s2[1:J])

inv.s20 <- 1/s2.0

w ~ dunif(0,1)

tau2[1] <- (s2.0-s2.0\*w)/w

for (j in 1:2) {inv.tau2[j] <- 1/tau2[j]}

# second stage variance for outlier group

tau2[2] <- tau2[1]+deltau

deltau ~ dexp(10)

mu ~ dnorm(0,0.01);

OR.NRT <- exp(mu)

TMDLL <- sum(MDLL[]) }

"

**# initial values and estimation**

inits <- function(){list(b=rnorm(90,0,0.1),mu=rnorm(1,0,0.1))}

C4 = rube(model4, DS\_4\_2, inits)

summary(C4)

pars <- c("OR.NRT","tau2","OR.new","G2","TMDLL","exceed")

R4 = rube(model4, DS\_4\_2, inits, pars, n.burn=500, n.thin=1, n.chains=2,n.iter=5000)

summary(R4,limit=90)

**# Fit measures**

pars <- c("LL")

R4.LL = rube(model4, DS\_4\_2, inits, pars, n.burn=500, n.thin=1, n.chains=2,n.iter=5000)

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