library(rstan)

library(loo)

setwd("C:/R files BHMRA")

attach("DS\_7\_4.Rdata")

attach("DS\_7\_4\_stan.Rdata")

**# Poisson-gamma**

attendPG.stan <- "

data { int N;

int K;

int y[N];

matrix[N,K] X; }

parameters { vector[K] beta;

real <lower=0> omega[N];

real<lower=0> alpha;}

transformed parameters { vector[N] mu;

real phi;

phi =1/alpha;

mu = exp(X\*beta);

}

model { alpha ~ gamma(1, 0.01);

beta[1] ~ normal(0,100);

omega ~ gamma(alpha,alpha);

for(i in 2:K) beta[i] ~ normal(0,1);

for (i in 1:N) { target += poisson\_lpmf(y[i] | mu[i]\*omega[i]); }

}

generated quantities{real log\_lik[N];

real omega\_new[N];

int y\_new[N];

real dv[N];

real <lower=0, upper=1> exc[N];

for (i in 1:N) { log\_lik[i]= poisson\_lpmf(y[i] | mu[i]\*omega[i]);

omega\_new[i] = gamma\_rng(alpha,alpha);

y\_new[i] = poisson\_rng(mu[i]\*omega\_new[i]);

dv[i] = y[i]\*log((y[i]+0.001)/(mu[i]\*omega[i]+0.001))-(y[i]- mu[i]\*omega[i]);

exc[i] = (y\_new[i] > y[i])+0.5\*(y\_new[i] ==y[i]);}

}

"

**# Estimation**

sm <- stan\_model(model\_code=attendPG.stan)

fitPG <- sampling(sm, data = DS\_7\_4\_stan, iter = 1500, warmup=250, chains = 2, seed=12345)

summary(fitPG, pars = c("beta","alpha"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# posterior mean deviances**

dv.mn <- apply(as.matrix(fitPG,pars="dv"),2,mean)

**# Scaled Deviance**

2\*sum(dv.mn)

**# posterior mean exceedance probabilities**

exc.samps <- as.matrix(fitPG,pars="exc")

exc.mn <- apply(exc.samps,2,mean)

sum((exc.mn>0.95)+(exc.mn <0.05))

**# Fit**

loo(as.matrix(fitPG,pars="log\_lik"))

**# extreme omega**

omega.mn <- apply(as.matrix(fitPG,pars="omega"),2,mean)

range(omega.mn)

**# Direct NB2**

attends.stan <- "

data { int N; int K; int y[N]; matrix[N,K] X; }

parameters { vector[K] beta;

real<lower=0> alpha; }

transformed parameters { vector[N] mu;

mu = exp(X\*beta); }

model { alpha ~ gamma(1, 0.01);

beta[1] ~ normal(0,100);

for(i in 2:K) beta[i] ~ normal(0,1);

y ~ neg\_binomial\_2(mu,alpha);}

generated quantities{real log\_lik[N];

for (i in 1:N) { log\_lik[i]= neg\_binomial\_lpmf(y[i] | mu[i],alpha); }

}

"

sm <- stan\_model(model\_code=attends.stan)

fitNB <- sampling(sm, data = DS\_7\_4\_stan, iter = 1500, warmup=250,chains = 2, seed=12345)

summary(fitNB, pars = c("beta","alpha"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

loo(as.matrix(fitNB,pars="log\_lik"))

lkd.mn <- apply(as.matrix(fitNB,pars="log\_lik"),2,mean)

sum(lkd.mn)

**# Dispersion Regression**

attendDR.stan <- "

data { int N; int K; int y[N]; matrix[N,K] X; }

parameters { vector[K] beta;

vector[K] delta; }

transformed parameters { vector[N] mu;

vector[N] alpha;

mu = exp(X\*beta);

alpha = exp(X\*delta);}

model { beta ~ normal(0,10);

delta ~ normal(0,10);

y ~ neg\_binomial\_2(mu,alpha);}

generated quantities{real log\_lik[N];

for (i in 1:N) { log\_lik[i]= neg\_binomial\_lpmf(y[i] | mu[i],alpha[i]); }

}

"

sm <- stan\_model(model\_code=attendDR.stan)

fitDR <- sampling(sm, data = DS\_7\_4\_stan, iter = 1500, warmup=250,chains = 2, seed=12345)

summary(fitDR, pars = c("beta","delta"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# Fit

loo(as.matrix(fitDR,pars="log\_lik"))

**# Poisson-gamma NBP**

attendNBP.stan <- "

data { int N;

int K;

int y[N];

matrix[N,K] X; }

parameters { vector[K] beta;

real <lower=0> omega[N];

real<lower=0> alpha;

real<lower=0> P;}

transformed parameters { vector[N] mu;

vector[N] xi;

mu = exp(X\*beta);

for (i in 1:N) { xi[i]=alpha\*mu[i]^(2-P);}

}

model { alpha ~ gamma(1, 0.01);

P ~ exponential(1);

beta ~ normal(0,10);

for (i in 1:N) { omega[i] ~ gamma(xi[i],xi[i]);

target += poisson\_lpmf(y[i] | mu[i]\*omega[i]); }

}

generated quantities{real log\_lik[N];

for (i in 1:N) { log\_lik[i]= poisson\_lpmf(y[i] | mu[i]\*omega[i]); }

}

"

sm <- stan\_model(model\_code=attendNBP.stan)

fitNBP <- sampling(sm, data = DS\_7\_4\_stan, iter = 1500, warmup=250, chains = 2, seed=12345)

summary(fitNBP, pars = c("beta","alpha","P"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# fit

loo(as.matrix(fitNBP,pars="log\_lik"))

lkd.mn <- apply(as.matrix(fitNBP,pars="log\_lik"),2,mean)

sum(lkd.mn)