library(rstan)

library(loo)

setwd("C:/R files BHMRA")

data = read.table("DS\_11\_5.txt",header=T)

attach(data)

Data=list(n=91,age=age,d=d,e=e)

makeham.stan ="

data { int<lower=1> n; // number of ages

int<lower=0> d[n];

vector[n] e; // offset

int<lower=0> age[n];

}

parameters {

vector[n] log\_alpha;

vector[n] log\_beta;

vector[n] log\_delta;

real<lower=0.001> sdRW[3]; // RW1 effects, st devs

}

transformed parameters {

real alpha[n];

real beta[n];

real delta[n];

real mu[n];

real q[n];

for (i in 1:n) {alpha[i]=exp(log\_alpha[i]);

beta[i]=exp(log\_beta[i]);

delta[i] = exp(log\_delta[i]);

mu[i] = alpha[i]+beta[i]\*delta[i]^age[i];

q[i] = 1-exp(-mu[i]);}

}

model { target += student\_t\_lpdf(sdRW| 4,0, 0.25);

target += normal\_lpdf(log\_alpha[1]| 0, 5);

target += student\_t\_lpdf(log\_beta[1]| 4,0, 5);

target += student\_t\_lpdf(log\_delta[1]| 4, 0, 5);

for (i in 2:n) {

target += normal\_lpdf(log\_alpha[i] | log\_alpha[i-1], sdRW[1]);

target += student\_t\_lpdf(log\_beta[i] | 4, log\_beta[i-1], sdRW[2]);

target += student\_t\_lpdf(log\_delta[i] | 4, log\_delta[i-1], sdRW[3]); }

for (i in 1:n) {

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

}

generated quantities{real log\_lik[n];

for (i in 1:n) { log\_lik[i]= poisson\_lpmf(d[i] | e[i]\*mu[i]); }

}

"

**# Compilation**

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

**# Estimation**

fit <- sampling(sm,data=Data,iter = 5000,warmup=250,chains = 2,seed= 12345)

summary <- summary(fit, pars = c("beta","alpha","delta","q","sdRW"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Plot of alpha**

plot(apply(matrix(extract(fit,"alpha",permute=F),2\*4750,91),2,mean),xlab="Age", ylab=expression(paste("Posterior Mean ",alpha)))

**# Fit**

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

**# Predictive Exceedance Checks**

mu.samps = (as.matrix(fit,pars="mu"))

dnew=check=matrix(,9500,91)

for (t in 1:9500) {for (i in 1:91) {dnew[t,i] = rpois(1,mu.samps[t,i]\*e[i])

check[t,i] = ifelse(dnew[t,i]>d[i],1,0)+ifelse(dnew[t,i]==d[i],0.5,0)}}

predexceed=apply(check,2,mean)

sum((predexceed>0.95)+(predexceed<0.05))

**# samples of q, Life Table and Actuarial Value Calculations**

qsamps = (as.matrix(fit,pars="q"))

L=D=N=a=WLAD=matrix(,9500,91)

for (t in 1:9500) {L[t,1] <- 10000

for (x in 2:91) {L[t,x] <- L[t,x-1]\*(1-qsamps[t,x-1])}

for (x in 1:91) {D[t,x] <- L[t,x]/(1.06^(x-1))}}

for (t in 1:9500) {for (x in 1:91) {N[t,x] <- sum(D[t,x:91])}}

for (t in 1:9500) {for (x in 1:91) { a[t,x] <- N[t,x]/D[t,x]

WLAD[t,x] <- 12\*(a[t,x]-11/24)}}

**# Whole Life Annuity Due, Age 60**

hist(WLAD[,61],prob=T,xlab="Life Annuity",main="Figure 11.7 Posterior Density, Monthly Whole Life Annuity",col="gray")

lines(density(WLAD[,61]))

**#**

**# Makehams with Constant Delta**

**#**

makehamFX.stan ="

data { int<lower=1> n; // number of ages

int<lower=0> d[n];

vector[n] e; // offset

int<lower=0> age[n];

}

parameters {

vector[n] log\_alpha;

vector[n] log\_beta;

real<lower=0.001> sdRW[2]; // RW1 st devns

real<lower=0> delta;

}

transformed parameters {

real alpha[n];

real beta[n];

real mu[n];

real q[n];

for (i in 1:n) {alpha[i]=exp(log\_alpha[i]);

beta[i]=exp(log\_beta[i]);

mu[i] = alpha[i]+beta[i]\*delta^age[i];

q[i] = 1-exp(-mu[i]);}

}

model { target += normal\_lpdf(sdRW| 0, 0.25);

target += gamma\_lpdf(delta| 1, 0.1);

target += normal\_lpdf(log\_alpha[1]| 0, 5);

target += normal\_lpdf(log\_beta[1]| 0, 5);

for (i in 2:n) {

target += normal\_lpdf(log\_alpha[i] | log\_alpha[i-1], sdRW[1]);

target += normal\_lpdf(log\_beta[i] | log\_beta[i-1], sdRW[2]);

}

for (i in 1:n) {target += poisson\_lpmf(d[i]|e[i]\*mu[i]);}

}

generated quantities{real log\_lik[n];

for (i in 1:n) { log\_lik[i]= poisson\_lpmf(d[i] | e[i]\*mu[i]); }

}

"

# Compilation

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

# Estimation

fitFX <- sampling(sm,data=Data,iter=5000,warmup=250,chains = 2,seed= 12345)

summary(fitFX, pars = c("beta","alpha","delta","q","sdRW"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

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

**# Predictive Exceedance Checks**

mu.samps = as.matrix(fitFX,pars="mu")

dnew=check=matrix(,9500,91)

for (t in 1:9500) {for (i in 1:91) {dnew[t,i] = rpois(1,mu.samps[t,i]\*e[i])

check[t,i] = ifelse(dnew[t,i]>d[i],1,0)+ifelse(dnew[t,i]==d[i],0.5,0)}}

predexceed=apply(check,2,mean)

sum((predexceed>0.95)+(predexceed<0.05))