setwd("C:/R files BHMRA")

data = read.table("DS\_11\_2.txt",header=T)

attach(data)

require(rube)

require(loo)

require(rstan)

Sys.setenv(BUGSDIR="c:\\users\\P Congdon\\documents\\WINBUGS14")

**# number of subjects**

N=137

**#**

**# Define data for models 1 and 2**

**#**

**# number of intervals**

J=20

**# quantile or equally spaced cutpoints on time scale**

a=numeric(J+1)

for (j in 1:J+1){a[j] <- quantile(t,(j-1)/J)

# a[j] <- max(t)\*(j-1)/J

}

**# index for likelihood terms**

idx=matrix(,N,J)

ndx=0

for (i in 1:N) {for (j in 1:J){ndx = ndx+1

idx[i,j]=ndx}}

**# response and offset**

y=del=matrix(,N,J)

for (i in 1:N) {for (j in 1:J){y[i,j] <- d[i]\*(t[i]>=a[j])\*(a[j+1] >= t[i]);

**# offset terms**

del[i,j] <- (min(t[i],a[j+1])-a[j])\*(t[i]>=a[j])}}

**#**

**# Model 1 RW1 prior on interval intercepts**

**#**

**# Data**

D=list(N=N,J=J,cell=cell,KS=KS,PT=PT,y=y,del=del,idx=idx)

model1= " model { **# car.normal inputs at boundary points**

w[1] <- 1; adj[1] <- 2; n[1] <- 1

w[(J-2)\*2 + 2] <- 1; adj[(J-2)\*2 + 2] <- J-1; n[J] <- 1

**# car.normal inputs at interior points**

for (t in 2:J-1) {w[2+(t-2)\*2] <- 1; adj[2+(t-2)\*2] <- t-1

w[3+(t-2)\*2] <- 1; adj[3+(t-2)\*2] <- t+1;

n[t] <- 2}

**# RW1 prior on interval intercepts**

alpha0[1:J] ~ car.normal(adj[], w[], n[], tau.alpha)

tau.alpha ~ dgamma(1,0.01)

sigma.alpha <- sqrt(1/tau.alpha)

beta0 ~ dnorm(0,0.001)

for (j in 1:J) {alpha[j] <- beta0+alpha0[j]}

for (i in 1:N) { KS.c[i] <- (KS[i]-mean(KS[]))/10

for (j in 1:J){**# Poisson likelihood with offset del**

y[i,j] ~ dpois(mu[i,j])

mu[i,j] <- del[i,j]\*theta[i,j]

log(theta[i,j]) <- beta0+alpha0[j]+beta.cell[cell[i]] +

beta.PT\*PT[i] + beta.KS\*KS.c[i] + beta.int\*KS.c[i]\*PT[i]

loglik[idx[i,j]] <- -mu[i,j]+y[i,j]\*log(mu[i,j])-logfact(y[i,j]) }}

# Priors:

beta.KS ~ dnorm(0.0, 0.01);

beta.int ~ dnorm(0.0, 0.001);

beta.PT ~ dnorm(0.0, 0.001);

beta.cell[1] <- 0;

for(k in 2:4) { beta.cell[k] ~ dnorm(0.0, 0.001)}}

"

**# Initial Values and Estimation**

inits1=list(beta.cell=c(NA,0,0,0),beta.PT=0, beta.KS=0,beta.int=0,beta0=-5,

sigma.alpha=0.1,alpha0=rep(0,J))

inits2=list(beta.cell=c(NA,0,0,0), beta.PT=0.1, beta.KS=-0.5,beta.int=-0.5,

beta0=-4.9,sigma.alpha=0.5,alpha0=rep(0,J))

inits=list(inits1,inits2)

pars= c("beta.KS","beta.int","sigma.alpha","beta.PT","alpha","loglik","beta.cell")

M1 = rube(model1, D, inits)

R1 = rube(model1, D, inits, pars, n.burn=1000, n.thin=1, n.chains=2,n.iter=20000)

summary(R1,limit=50)

**# Density plots**

p3(R1)

**# Posterior mean and 80% CRI for interval varying intercept**

A1=apply(as.matrix(R1$sims.list$alpha),2,quantile,0.10)

A3=apply(as.matrix(R1$sims.list$alpha),2,quantile,0.90)

intervals=seq(1,20)

par(bg = "grey90")

plot(intervals, R1$mean$alpha, type="o", col="black", pch="o", lty=1, ylim=c(min(A1),max(A3)),

xlab="Interval",font.axis=2,cex.lab=1.3,ylab="")

points(intervals, A1, col="black", pch="\*",cex=2)

lines(intervals, A1, col="black",lty=3)

points(intervals, A3, col="black",pch="+",cex=1.5)

lines(intervals, A3, col="black", lty=3)

title(ylab="Posterior Mean and 80% CRI", line=2, cex.lab=1.2)

**# Fit**

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

**#**

**# Model 2. RW1 prior on interval intercepts and time varying impact of Karnofsky score**

**#**

**# Data**

D=list(N=N,J=J,cell=cell,KS=KS,PT=PT,y=y,del=del,idx=idx,a=a)

model2= " model { # car.normal boundary points

w[1] <- 1; adj[1] <- 2; n[1] <- 1

w[(J-2)\*2 + 2] <- 1; adj[(J-2)\*2 + 2] <- J-1; n[J] <- 1

# car.normal interior points

for (t in 2:J-1) {w[2+(t-2)\*2] <- 1; adj[2+(t-2)\*2] <- t-1

w[3+(t-2)\*2] <- 1; adj[3+(t-2)\*2] <- t+1;

n[t] <- 2}

**# RW1 prior on interval intercepts**

alpha0[1:J] ~ car.normal(adj[], w[], n[], tau.alpha)

tau.alpha ~ dgamma(1,0.01)

for (j in 1:J) {alpha[j] <- beta0+alpha0[j]}

beta0 ~ dnorm(0,0.001)

**# RW1 prior on KS coefficient**

# alphaKS.0[1:J] ~ car.normal(adj[], w[], n[], tau.KS)

# tau.KS ~ dgamma(1,0.01)

# betaKS.0 ~ dnorm(0,0.001)

**# Prior Precision for prior on KS Coeffcient Adjusted for Interval Length**

beta.KS[1] ~ dnorm(0,0.1)

for (j in 2:J) {beta.KS[j] ~ dnorm(beta.KS[j-1],tau.beta.KS[j])

tau.beta.KS[j] <- tau.KS/(a[j]-a[j-1]) }

tau.KS ~ dgamma(1,0.0001)

**# scaled KS**

for (i in 1:N) { KS.c[i] <- (KS[i]-mean(KS[]))/10

for (j in 1:J){

**# Poisson likelihood with offset del**

y[i,j] ~ dpois(mu[i,j])

mu[i,j] <- del[i,j]\*theta[i,j]

log(theta[i,j]) <- beta0+alpha0[j]+beta.cell[cell[i]] +

beta.PT\*PT[i] + beta.KS[j]\*KS.c[i] + beta.int\*KS.c[i]\*PT[i]

loglik[idx[i,j]] <- -mu[i,j]+y[i,j]\*log(mu[i,j])-logfact(y[i,j])}}

# Priors:

beta.int ~ dnorm(0.0, 0.001);

beta.PT ~ dnorm(0.0, 0.001);

beta.cell[1] <- 0;

for(k in 2:4) { beta.cell[k] ~ dnorm(0.0, 0.001)}}

"

**# Initial Values and Estimation**

inits1=list(beta.cell=c(NA,0,0,0),beta.PT=0,beta0=-5, beta.KS=rep(0,J),

beta.int=0,tau.alpha=10,tau.KS=10,alpha0=rep(0,J))

inits2=list(beta.cell=c(NA,0,0,0), beta.PT=-0.1, beta.KS=rep(0,J),

beta.int=-0.5, beta0=-4.9,tau.alpha=20,tau.KS=20,alpha0=rep(0,J))

inits=list(inits1,inits2)

pars=c("beta.int","tau.alpha","tau.KS","beta.KS","beta.PT","loglik")

M2 = rube(model2, D, inits)

R2 = rube(model2, D, inits, pars, n.burn=1000, n.thin=1, n.chains=2,n.iter=20000)

summary(R2,limit=50)

**# interval varying effect of Karnofsky score**

par(bg = "light yellow")

plot(R2$mean$beta.KS, font.axis=2,xlab ="Interval",ylab = expression(paste("Posterior mean ", beta[KS])))

lines(R2$mean$beta.KS, y = NULL, type = "l")

**# Fit**

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

**#**

**# Model 3, Counting Process Version of Cox Model**

**#**

**# Define data**

**#**

obs\_t=t

t.d=subset(obs\_t,d==1)

**# unique event times**

t.d.unique=unique(t.d)

NT=length(t.d.unique)

t\_unique=c(sort(t.d.unique),max(obs\_t)+1)

**# define at risk and counting process increments**

Y=dN=matrix(,N,NT)

for (i in 1:N) { for (j in 1:NT) {Y[i,j] =ifelse(obs\_t[i]>=t\_unique[j],1,0)}}

for (i in 1:N) { for (j in 1:NT) { dN[i, j] =Y[i, j] \* (t\_unique[j + 1] > obs\_t[i]) \* d[i]}}

**# centred and scaled Karnosky score**

KS.c=(KS-mean(KS))/10

**# dataset**

Dstan=list(N=N,NT=NT,t\_unique=t\_unique,Y=Y,dN=dN,Z=PT,KS=KS.c)

CP.stan ="

data {

int<lower=0> N;

real KS[N];

int<lower=0> NT;

int<lower=0> Y[N,NT];

int<lower=0> dN[N,NT];

int<lower=0> t\_unique[NT + 1];

real PT[N];

}

transformed data {

real c;

real r;

c = 0.001;

r = 0.1;

}

parameters {

real beta[2];

real betaKS[NT];

real<lower=0.001> sigmaKS;

real<lower=0> dL0[NT];

}

model {

real dt[NT];

beta ~ normal(0, 10);

sigmaKS ~ uniform(0,1);

betaKS[1] ~ normal(0,1);

**// RW prior on KS coefficients**

for (j in 2:NT){betaKS[j] ~ normal(betaKS[j-1],sigmaKS);}

**// gamma increments prior**

for (j in 1:NT) {dt[j] = t\_unique[j+1] - t\_unique[j];

dL0[j] ~ gamma(r \* dt[j] \* c, c);

for (i in 1:N) { if (Y[i, j] != 0)

target += poisson\_lpmf(dN[i, j]|

Y[i, j]\*exp(betaKS[j]\*KS[i]+beta[1]\*PT[i]+beta[2]\*PT[i]\*KS[i]) \* dL0[j]); } }

}

generated quantities {

real S\_noPT[NT];

real S\_PT[NT];

for (j in 1:NT) { **// Survivor functions by prior therapy, Karnofsky score set at upper quartile**

real s;

s = 0;

for (i in 1:j)

s = s + dL0[i];

S\_PT[j] = pow(exp(-s), exp(betaKS[j]\*1.64+beta[1]+beta[2]\*1.64));

S\_noPT[j] = pow(exp(-s), exp(betaKS[j]\*1.64)); }}

"

**# Compilation and Estimation**

sm = stan\_model(model\_code=CP.stan)

fit = sampling(sm,data =Dstan,iter = 1500,warmup=250,chains = 2,seed= 12345)

print(fit)

**# plot of time-varying effect of KS score**

betaKS <- extract(fit,"betaKS",permute=F)

par(bg = "light yellow")

A1=apply(betaKS,3,quantile,0.10)

A2=apply(betaKS,3,mean)

A3=apply(betaKS,3,quantile,0.90)

plot(A2, type="o", col="black", pch="o", lty=1, ylim=c(min(A1),max(A3)), ylab=expression(paste("Posterior Mean and 80% CRI, ",beta[KS])),xlab="Interval", font.axis=2)

points(A1, col="black", pch="\*")

lines(A1, col="black",lty=2)

points(A3, col="black",pch="+")

lines(A3, col="black", lty=3)

**# plots of survival curves**

S\_PT <- extract(fit,"S\_PT",permute=F)

S\_noPT <- extract(fit,"S\_noPT",permute=F)

plot(apply(S\_PT,3,mean), type="o", col="blue", pch=21, lty=1,xlab ="Interval",ylab ="Posterior mean survival")

points(apply(S\_noPT,3,mean), col="red", pch=22,lty=2)

lines(apply(S\_noPT,3,mean), col="red", pch=22,lty=2)

legend("topright", c("PT","No PT"), cex=0.8, col=c("blue","red"), pch=21:22, lty=1:2)