library(rstan)

library(loo)

library(survival)

library(rms)

setwd("C:/R files BHMRA")

D = read.table("DS\_11\_8.txt ",header=T)

N=1000

attach(D)

# status = competing risks 1,2

cens1=ifelse(status==2,1,0)

cens2=ifelse(status==1,1,0)

#

**# Comparison K-M, Weibull, log-logistic**

#

D$Surv <- with(D, Surv(time, cens1))

## Kaplan-Meier estimator

km.null <- survfit(Surv(time,cens1) ~ 1,data=D)

# Parametric estimation, Weibull distribution

weibull.null <- survreg(data = D, Surv ~ 1, dist = "weibull")

**# Figure 11.9 Colour**

plot(km.null,conf.int=F, main="Figure 11.9 Survival Curve without Predictors",

xlab="Days in Neutropenia",lwd=3)

lines(x = predict(weibull.null, type = "quantile", p = seq(0.01, 0.99, by=.01))[1,],

y = rev(seq(0.01, 0.99, by = 0.01)),col = "red",lwd=3)

# Parametric estimation with log-logistic distribution

loglogistic.null <- survreg(data = D, Surv ~ 1, dist = "loglogistic")

lines(x = predict(loglogistic.null, type = "quantile", p = seq(0.01, 0.99, by=.01))[1,],

y = rev(seq(0.01, 0.99, by = 0.01)), col = "blue",lwd=3)

# Add legends

legend(x = "topright", legend = c("Kaplan-Meier", "Weibull", "Log-logistic"),

lwd=3, bty = "n", col = c("black", "red", "blue"))

**# schedule of specified timepoints for cumulative hazard plot**

timeprof=seq(0,25,1)

**# K=2 competing events,p=3 regressors**

D=list(N=1000,N2=2000,T=26,timeprof=timeprof,time=time,cens1=cens1,cens2=cens2,sex=sex,

allo=allo,p=3,K=2)

**#**

**# Weibull PH**

**#**

weibCR.stan ="

data {

int<lower=1> N; // number of cases

int<lower=1> N2; // number of cases

int<lower=1> T; // number of time points for CH profiles

int<lower=1> K; // number of competing causes of exit

vector[N] time; // observed or censored times

vector[T] timeprof; // time points for CH profiles

int<lower=0,upper=1> cens1[N]; // right censoring, cause 1

int<lower=0,upper=1> cens2[N]; // right censoring, cause 2

int<lower=0> p; // total regression parameters, including intercept

int<lower=0> allo[N];

int<lower=0> sex[N];

}

parameters { vector[p] beta1;

vector[p] beta2;

real<lower=0> shape[K]; // shape parameters

}

transformed parameters {

real eta1[N];

real nu1[N];

real eta2[N];

real nu2[N];

real S1allo[T]; // survival functions

real S1auto[T];

real S2allo[T];

real S2auto[T];

real CH1allo[T]; // cumulative hazards

real CH1auto[T];

real CH2allo[T];

real CH2auto[T];

for (t in 1:T) {

S1allo[t] = exp(-exp(beta1[1]+beta1[2])\*timeprof[t]^ shape[1]);

S1auto[t] = exp(-exp(beta1[1])\*timeprof[t]^ shape[1]);

S2allo[t] = exp(-exp(beta2[1]+beta2[2])\*timeprof[t]^shape[2]);

S2auto[t] = exp(-exp(beta2[1])\*timeprof[t]^ shape[2]);

CH1allo[t] = -log(S1allo[t]);

CH1auto[t] = -log(S1auto[t]);

CH2allo[t] = -log(S2allo[t]);

CH2auto[t] = -log(S2auto[t]);}

for (i in 1:N) {eta1[i]=beta1[1]+beta1[2]\*allo[i]+beta1[3]\*sex[i];

nu1[i] = exp(-eta1[i] / shape[1]);

eta2[i]=beta2[1]+beta2[2]\*allo[i]+beta2[3]\*sex[i];

nu2[i] = exp(-eta2[i] / shape[2]); }

}

model { target += gamma\_lpdf(shape | 0.01, 0.01);

for (i in 1:N) {

if (cens1[i] == 0) { target += weibull\_lpdf(time[i] | shape[1], nu1[i]); }

else if (cens1[i] == 1) { target += weibull\_lccdf(time[i] | shape[1], nu1[i]); }

if (cens2[i] == 0) { target += weibull\_lpdf(time[i] | shape[2], nu2[i]); }

else if (cens2[i] == 1) { target += weibull\_lccdf(time[i] | shape[2], nu2[i]); }

} }

generated quantities{real log\_lik[N2];

**// expanded log-likelihood vector over K=2 causes**

for (i in 1:N) {

if (cens1[i] == 0) { log\_lik[i]= weibull\_lpdf(time[i] | shape[1], nu1[i]); }

else if (cens1[i] == 1) { log\_lik[i]= weibull\_lccdf(time[i] | shape[1], nu1[i]); }

if (cens2[i] == 0) { log\_lik[i+N]= weibull\_lpdf(time[i] | shape[2], nu2[i]); }

else if (cens2[i] == 1) { log\_lik[i+N]= weibull\_lccdf(time[i] | shape[2], nu2[i]); }

}}

"

**# Estimation**

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

fitCR <- sampling(sm,data =D,iter = 1500,warmup=250,chains = 2,seed= 12345)

summary(fitCR, pars = c("beta1", "beta2","shape"), probs = c(0.025,0.5, 0.975))$summary

**# Fit**

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

**# Cumulative Hazards Plots**

summaryCH1=summary(fitCR, pars = c("CH1allo","CH1auto"), probs = c(0.025,0.5, 0.975))$summary

summaryCH2=summary(fitCR, pars = c("CH2allo","CH2auto"), probs = c(0.025,0.5, 0.975))$summary

CH1=summaryCH1[,1]

CH2=summaryCH2[,1]

CH1allo=CH1[1:26]

CH1auto=CH1[27:52]

CH2allo=CH2[1:26]

CH2auto=CH2[27:52]

**# Cumulative Hazard Plots for Bloodstream Infection**

plot(timeprof, CH1allo, type="o", col="blue", pch="o", lty=1, ylim=c(min(CH1allo),max(CH1auto)), ylab="Cumulative Cause-Specific Hazards",xlab="Days", main="Figure 11.10 Cumulative Hazard by Transplant Source")

points(timeprof, CH1auto, col="red", pch="\*")

lines(timeprof, CH1auto, col="red",lty=2)

legend(2.5,0.6,legend=c("allogeneic","autologous"), col=c("blue","red"),pch=c("o","\*"),lty=c(1,2))

**# Cumulative Hazard Plots for End of Neutropenia**

plot(timeprof, CH2allo, type="o", col="blue", pch="o", lty=1, ylim=c(min(CH2allo),max(CH2auto)), ylab="Cumulative Cause-Specific Hazards",xlab="Days")

points(timeprof, CH2auto, col="red", pch="\*")

lines(timeprof, CH2auto, col="red",lty=2)

legend(2.5,3,legend=c("allogeneic","autologous"), col=c("blue","red"),pch=c("o","\*"),lty=c(1,2))

**#**

**# Cox Regression, Define at risk and event indicators for two causes**

**#**

d1=ifelse(status==1,1,0)

t.d1=subset(time,status==1)

# unique event times

t.d1.unique=unique(t.d1)

NT1=length(t.d1.unique)

t1\_unique=c(sort(t.d1.unique),max(time)+1)

# define at risk and counting process increments

Y1=dN1=matrix(,N,NT1)

for (i in 1:N) { for (j in 1:NT1) {Y1[i,j] =ifelse(time[i]>=t1\_unique[j],1,0)}}

for (i in 1:N) { for (j in 1:NT1) {dN1[i, j] =Y1[i, j] \* (t1\_unique[j + 1] > time[i]) \* d1[i]}}

d2=ifelse(status==2,1,0)

t.d2=subset(time,status==2)

# unique event times

t.d2.unique=unique(t.d2)

NT2=length(t.d2.unique)

t2\_unique=c(sort(t.d2.unique),max(time)+1)

# define at risk and counting process increments

Y2=dN2=matrix(,N,NT2)

for (i in 1:N) { for (j in 1:NT2) {Y2[i,j] =ifelse(time[i]>=t2\_unique[j],1,0)}}

for (i in 1:N) { for (j in 1:NT2) {dN2[i, j] =Y2[i, j] \* (t2\_unique[j + 1] > time[i]) \* d2[i]}}

D=list(N=N,NT1=NT1,t1\_unique=t1\_unique,Y1=Y1,dN1=dN1,

NT2=NT2,t2\_unique=t2\_unique,Y2=Y2,dN2=dN2,allo=allo,sex=sex)

CR.stan ="

data {

int<lower=0> N;

int<lower=0> NT1; // number of unique time points

int<lower=0> NT2;

int<lower=0> Y1[N,NT1];

int<lower=0> dN1[N,NT1];

int<lower=0> Y2[N,NT2];

int<lower=0> dN2[N,NT2];

real <lower=0> t1\_unique[NT1 + 1];

real <lower=0> t2\_unique[NT2 + 1];

int<lower=0> allo[N];

int<lower=0> sex[N];

}

transformed data {

real c;

real r;

c = 0.001;

r = 0.1;

}

parameters {

real beta1[2];

real beta2[2];

real<lower=0> dL1[NT1];

real<lower=0> dL2[NT2];

}

model {

real dt1[NT1];

real dt2[NT2];

beta1 ~ normal(0, 10);

beta2 ~ normal(0, 10);

// gamma increments prior

for (j in 1:NT1) {dt1[j] = t1\_unique[j+1] - t1\_unique[j];

dL1[j] ~ gamma(r \* dt1[j] \* c, c);

for (i in 1:N) { if (Y1[i, j] != 0)

target += poisson\_lpmf(dN1[i, j]|

Y1[i, j]\*exp(beta1[1]\*allo[i]+beta1[2]\*sex[i]) \* dL1[j]); } }

for (j in 1:NT2) {dt2[j] = t2\_unique[j+1] - t2\_unique[j];

dL2[j] ~ gamma(r \* dt2[j] \* c, c);

for (i in 1:N) { if (Y2[i, j] != 0)

target += poisson\_lpmf(dN2[i, j]|

Y2[i, j]\*exp(beta2[1]\*allo[i]+beta2[2]\*sex[i]) \* dL2[j]); } }

}

generated quantities {

real S1\_allo[NT1];

real S1\_auto[NT1];

real CH1\_allo[NT1];

real CH1\_auto[NT1];

real S2\_allo[NT2];

real S2\_auto[NT2];

real CH2\_allo[NT2];

real CH2\_auto[NT2];

for (j in 1:NT1) { // Survival by transfusion type

real s1;

s1 = 0;

for (i in 1:j)

s1 = s1 + dL1[i];

S1\_allo[j] = pow(exp(-s1), exp(beta1[1]));

S1\_auto[j] = exp(-s1);

CH1\_allo[j] = -log(S1\_allo[j]);

CH1\_auto[j] = -log(S1\_auto[j]); }

for (j in 1:NT2) { // Survival by transfusion type

real s2;

s2 = 0;

for (i in 1:j)

s2 = s2 + dL2[i];

S2\_allo[j] = pow(exp(-s2), exp(beta2[1]));

S2\_auto[j] = exp(-s2);

CH2\_allo[j] = -log(S2\_allo[j]);

CH2\_auto[j] = -log(S2\_auto[j]); }

}

"

**# Compilation and Estimation**

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

fitCR = sampling(sm,data =D,iter = 1500,warmup=250,chains = 2,seed= 12345)

summaryCR1=summary(fitCR,pars =c("beta1","beta2"),probs = c(0.025,0.5, 0.975))$summary

summaryCR2=summary(fitCR,pars =c("CH1\_allo","CH1\_auto"),probs = c(0.025,0.5, 0.975))$summary

CH1allo=summaryCR2[1:NT1,1]

na=NT1+1; nb=2\*NT1

CH1auto=summaryCR2[na:nb,1]

**# Cumulative Hazard Plots for Bloodstream Infection**

t1=t1\_unique[1:NT1]

plot(t1, CH1allo, type="o", col="blue", pch="o", lty=1, ylim=c(min(CH1allo),max(CH1auto)), xlim=c(0,40),ylab="Cumulative Cause-Specific Hazards",xlab="Days", main="Cox Regression. Cumulative Hazard by Transplant Source")

points(t1, CH1auto, col="red", pch="\*")

lines(t1, CH1auto, col="red",lty=2)

legend(2.5,0.6,legend=c("allogeneic","autologous"), col=c("blue","red"),pch=c("o","\*"),lty=c(1,2))

summaryCR3=summary(fitCR,pars =c("CH2\_allo","CH2\_auto"),probs = c(0.025,0.5, 0.975))$summary

CH2allo=summaryCR3[1:NT2,1]

na=NT2+1; nb=2\*NT2

CH2auto=summaryCR3[na:nb,1]

t2=t2\_unique[1:NT2]

**# Cumulative Hazard Plots for End of Neutropenia**

plot(t2, CH2allo, type="o", col="blue", pch="o", lty=1, ylim=c(min(CH2allo),max(CH2auto)),

xlim=c(0,40),ylab="Cumulative Cause-Specific Hazards",lwd=3,

xlab="Days", main="Figure 11.11. Cox Regression. Cumulative Hazard, End of Neutropenia, by Transplant Source",cex.main=0.8)

points(t2, CH2auto, col="red", pch="\*")

lines(t2, CH2auto, col="red",lty=2,lwd=3)

legend(2.5,8,legend=c("allogeneic","autologous"), col=c("blue","red"),pch=c("o","\*"),lty=c(1,2),lwd=3)

#

# CLASSICAL CAUSE-SPECIFIC COX REGRESSION

#

d1 <-ifelse(status==1,1,0)

coxph(Surv(time,d1)~allo+sex)

d2 <-ifelse(status==2,1,0)

coxph(Surv(time,d2)~allo+sex)