HCM-SCD-CEA.R

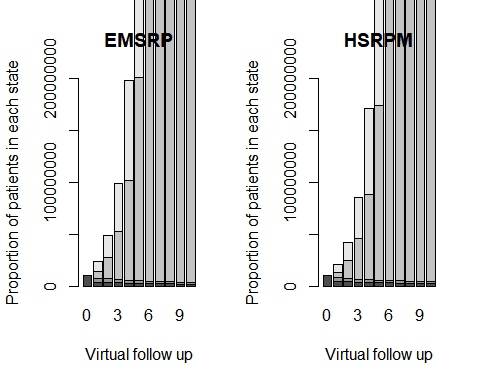
ADDAH

2019-12-27

# Bayesian Models for Cost-Effectiveness Analysis  
# Loads packages  
  
 S <- 4 # Number of health states  
 # s = 1 Healthy health state  
 # s = 2 Stroke HCM-Related Health state  
 # s = 3 SCD(Sudden Cardiac Death) Health state  
 # s = 4 DAC (Death All Causes) Health state  
 # EXISTING METHOD OF SCD-RISK-PREDICTION (EMSRP):Current Method of SCD Risk Prediction  
 # HCM - SCD RISK PREDICTION MODEL(HSRPM)  
   
 J <-10 # Number of years of follow up  
   
 # Now load the observered data on transitions among the states for the two treatments  
 Model.file="HCMmodel.txt" # Specifies file with the model defining observed data  
 inits1=source("hcminits1.txt")$value # Loads the initial values for the first chain  
 inits2=source("hcminits2.txt")$value # Loads the initial values of the second chain  
 inits=list(inits1,inits2) # Combines into a list files with inital values  
 dataBugs=source("HCMdata.txt")$value # Loads observed data  
   
 # Now run the MCMC  
 library(R2OpenBUGS)  
 params=c("lambda.0","lambda.1") # Defines parameters to save  
 n.iter <- 10000  
 n.burnin<- 5000  
 n.thin <- floor((n.iter- n.burnin)/5000)  
 n.chain=2  
 debug=FALSE  
 mm1 <- bugs(data=dataBugs,inits=inits,parameters.to.save = params,  
 model.file = Model.file,n.chains=n.chain,n.iter=n.iter,n.burnin=n.burnin,n.thin=n.thin,DIC = TRUE)  
 print(mm1,digits = 3)

## Inference for Bugs model at "HCMmodel.txt",   
## Current: 2 chains, each with 10000 iterations (first 5000 discarded)  
## Cumulative: n.sims = 10000 iterations saved  
## mean sd 2.5% 25% 50% 75% 97.5%  
## lambda.0[1,1] 3.18e-01 0.024 2.72e-01 3.01e-01 3.17e-01 3.34e-01 3.66e-01  
## lambda.0[1,2] 4.10e-01 0.050 3.14e-01 3.76e-01 4.10e-01 4.44e-01 5.06e-01  
## lambda.0[1,3] 7.06e-01 0.047 6.09e-01 6.75e-01 7.08e-01 7.38e-01 7.91e-01  
## lambda.0[1,4] 9.31e-01 0.030 8.62e-01 9.14e-01 9.35e-01 9.53e-01 9.77e-01  
## lambda.0[2,2] 6.46e-01 0.022 6.02e-01 6.31e-01 6.46e-01 6.60e-01 6.89e-01  
## lambda.0[2,3] 4.59e-01 0.052 3.57e-01 4.24e-01 4.59e-01 4.94e-01 5.58e-01  
## lambda.0[2,4] 6.04e-01 0.053 4.96e-01 5.69e-01 6.05e-01 6.41e-01 7.03e-01  
## lambda.1[1,1] 4.67e-01 0.026 4.14e-01 4.49e-01 4.66e-01 4.84e-01 5.18e-01  
## lambda.1[1,2] 3.23e-01 0.050 2.26e-01 2.89e-01 3.22e-01 3.56e-01 4.23e-01  
## lambda.1[1,3] 5.51e-01 0.051 4.49e-01 5.17e-01 5.52e-01 5.86e-01 6.50e-01  
## lambda.1[1,4] 7.26e-01 0.049 6.24e-01 6.95e-01 7.30e-01 7.60e-01 8.17e-01  
## lambda.1[2,2] 6.42e-01 0.022 5.99e-01 6.27e-01 6.42e-01 6.56e-01 6.85e-01  
## lambda.1[2,3] 4.65e-01 0.053 3.60e-01 4.30e-01 4.65e-01 5.01e-01 5.68e-01  
## lambda.1[2,4] 6.10e-01 0.051 5.06e-01 5.76e-01 6.11e-01 6.44e-01 7.05e-01  
## deviance 3.94e+12 0.000 3.94e+12 3.94e+12 3.94e+12 3.94e+12 3.94e+12  
## Rhat n.eff  
## lambda.0[1,1] 1.001 3700  
## lambda.0[1,2] 1.001 7000  
## lambda.0[1,3] 1.001 10000  
## lambda.0[1,4] 1.003 4600  
## lambda.0[2,2] 1.001 10000  
## lambda.0[2,3] 1.001 10000  
## lambda.0[2,4] 1.001 10000  
## lambda.1[1,1] 1.001 10000  
## lambda.1[1,2] 1.001 10000  
## lambda.1[1,3] 1.001 8100  
## lambda.1[1,4] 1.001 10000  
## lambda.1[2,2] 1.001 10000  
## lambda.1[2,3] 1.001 10000  
## lambda.1[2,4] 1.001 10000  
## deviance 1.000 1  
##   
## For each parameter, n.eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
##   
## DIC info (using the rule, pD = Dbar-Dhat)  
## pD = 1.225 and DIC = 3.94e+12  
## DIC is an estimate of expected predictive error (lower deviance is better).

attach.bugs(mm1)  
   
 # Now we run the Markov model from R  
 start <- c(1000,0,0,0) # Analysis for virtual cohort of 1000, individuals  
 # NB All patients enter the model from the first state "Healthy"  
   
 #Determine the Markov transitions  
 m.0 <- m.1 <- array(NA,c(n.sims,S,(J+1)))  
 for(s in 1:S){  
 m.0[,s,1] <- start[s]  
 m.1[,s,1] <- start[s]  
 }  
   
 #NB  
 # BUGS only outputs matrices for lambda.0 and lambda.1 with simulations for the "random" part  
 # ie only the first 2 rows, as the last two are deterministically defined as c(0,0,1,1)  
 # because once a patient is in SCD,and DAC, they can't move away. So there is the need to   
 # reconstruct a full matrix with S rows and S columns for each MCMC simulations. This is done by   
 # defining new arrays lamda0 and lamda1 and then stacking up the simulated values for the first (S-2)  
 # rows saved in lambda.0[i,,] and lambda.1[i,,] for MCMC simulations i with a row vector  
 # containing (S-2) 0s and then two 1's, ie c(0,0, 1,1)  
  
 lamda0=lamda1=array(NA, c(n.sims,S,S))  
 for (i in 1:n.sims) {  
 lamda0[i,,]=rbind(rbind(lambda.0[i,,],c(0,0,1,1)),c(0,0,1,1))  
 lamda1[i,,]=rbind(rbind(lambda.1[i,,],c(0,0,1,1)),c(0,0,1,1))  
 for (j in 2: (J+1)) {  
 for (s in 1:S) {  
 # Now use lamda0,and lamda1, for the matrix multiplication  
 m.0[i,s,j] <- sum(m.0[i,,j-1]\*lamda0[i,,s])  
 m.1[i,s,j] <- sum(m.1[i,,j-1]\*lamda1[i,,s])  
 }  
   
 }  
   
 }  
 # Now we draw barplot of the number of people in each state at each time point during follow up  
   
 par(mfrow=c(1,2))  
 barplot(apply(m.0,c(2,3),sum),names.arg=seq(0,10),space=.2,xlab="Virtual follow up",  
 ylab="Proportion of patients in each state",main="EMSRP",ylim = c(0,200000000))  
   
 barplot(apply(m.1,c(2,3),sum),names.arg=seq(0,10),space=.2,xlab="Virtual follow up",  
 ylab="Proportion of patients in each state",main="HSRPM",ylim = c(0,200000000))

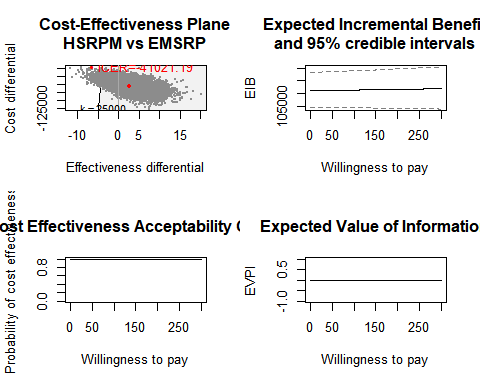
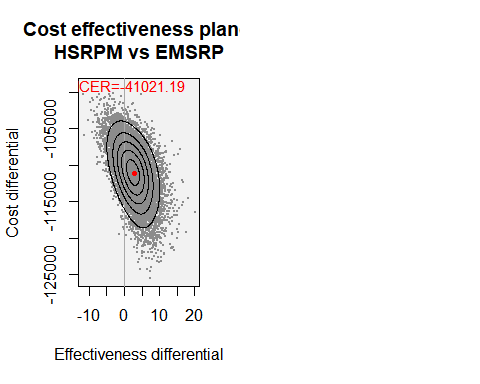


# Run the economic analysis  
 #Now we define the benefits  
 utility.score.0<-utility.score.1<-array(NA,c(n.sims,2,J)) # Defines measures of accumulated untility  
 # under each treatment  
 dec.rate<-0.35 # Defines utility decreament rate to apply when a non-fatal HCM event occurs at j >0  
 utility.score.0[,1,]<-rep(0.51,J) # Utility for occupying the state "Healthy" under treatment t=0  
 utility.score.1[,1,]<-rep(0.51,J) # Utility for occupyingthe state "Healty" under treatment t= 1  
 utility.score.0[,2,]<-rep(dec.rate\*utility.score.0[1,1,1], J) # Utility for occupying state "Stroke-HCM Related" under treatment t=0  
 utility.score.1[,2,]<- rep(dec.rate\*utility.score.1[1,1,1],J) # Utility for occupying state "Stroke-HCM Related" under treatment t=1  
 # for (i in 1:n.sims) {  
 # for (j in 2:J) {  
 # utility.score.0[i,2,j]<-utility.score.0[i,2,j-1]\*(1- dec.rate)  
 # utility.score.1[i,2,j]<-utility.score.1[i,2,j-1]\*(1- dec.rate)  
 # }  
   
 #}  
 # We now compute QALY's accumulated under each treatment for each year of follow up  
 Qal.0<-Qal.1<-matrix(NA,n.sims,J)  
 for (i in 1:n.sims) {  
 for (j in 1:J) {  
 Qal.0[i,j]<-(m.0[i,1,j]%\*%utility.score.0[i,1,j]   
 + m.0[i,2,j]%\*%utility.score.0[i,2,j])/m.0[1,1,1]  
 Qal.1[i,j]<-(m.1[i,1,j]%\*%utility.score.1[i,1,j]   
 + m.1[i,2,j]%\*%utility.score.1[i,2,j])/m.1[1,1,1]  
   
 }  
   
 }  
# Now sum values across all time points, and create matrix effectiveness  
 eff<-array(NA,c(n.sims,2,J))  
 eff[,1,]<-apply(Qal.0, 1,sum)  
 eff[,2,]<-apply(Qal.1, 1,sum)  
   
 # We define the annual cost for each non-fatal health state under each treatment  
 unit.cost.0 <-c(2.34,3.43)  
 unit.cost.1 <-c(2.23,3.43) #  
   
 #Create a holding cost variable to track yearly (j>0)accumulated cost under each treatment  
 cost.0<-cost.1<-matrix(NA,n.sims,J)  
 for (i in 1: n.sims) {  
 for (j in 2:(J+1)) {  
 cost.0[i,j-1]<-(m.0[i,S,j]+ m.0[i,(S-1),j])\*(unit.cost.0%\*%m.0[i,1:(S-2),j])/sum(m.0[i,1:(S-2),j])  
 + unit.cost.0%\*%m.0[i,1:(S-2),j]  
 cost.1[i,j-1]<-(m.0[i,S,j]+ m.0[i,(S-1),j])\*(unit.cost.1%\*%m.0[i,1:(S-2),j])/sum(m.0[i,1:(S-2),j])  
 + unit.cost.1%\*%m.0[i,1:(S-2),j]  
   
 }  
   
 }  
   
   
# We now derive a general formulation to apply discount to cost and benefits  
 rate.b <- 0.035 # discount rate for benefits (3.5%)  
 rate.c <- 0.035 # discount rate for costs (3.5%)  
 # Defines the discount factors  
 disc.b <- numeric(); disc.c <- numeric()  
 disc.b[1] <- 1; disc.c[1] <- 1  
 for (j in 2:J) {  
 disc.b[j] <- (1+rate.b)^(j-1)  
 disc.c[j] <- (1+rate.c)^(j-1)  
 }  
 disc.cost.0 <- disc.eff.0 <- disc.cost.1 <- disc.eff.1 <- matrix(NA,n.sims,J)  
 for (j in 1:J) {  
 disc.cost.0[,j] <- cost.0[,j]/disc.c[j]  
 disc.cost.1[,j] <- cost.1[,j]/disc.c[j]  
 disc.eff.0[,j] <- eff[,1,j]/disc.b[j]  
 disc.eff.1[,j] <- eff[,2,j]/disc.b[j]  
 }  
   
 # Now sum the values across all time points and create matrix of costs  
 c <- matrix(NA,n.sims,2)  
 c[,1] <- apply(disc.cost.0,1,sum)  
 c[,2] <- apply(disc.cost.1,1,sum)  
   
 # Sum all discounted valees of effectiveness and create a matrix of discounted effectiveness  
 e <- matrix(NA,n.sims,2)  
 e[,1] <- apply(disc.eff.0,1,sum)  
 e[,2] <- apply(disc.eff.1,1,sum)  
   
   
 # Cost-effectiveness analysis  
 library(BCEA)  
 ints <- c("EMSRP","HSRPM")  
 m <- bcea(e,c,ref=2,interventions=ints,Kmax=300)  
   
 contour2(m,300)

## The first available comparison will be selected. To plot multiple comparisons together please use the ggplot2 version. Please see ?contour2 for additional details.

## Loading required namespace: MASS

plot(m)



summary(m)

## NB: k (wtp) is defined in the interval [0 - 300]  
##   
## Cost-effectiveness analysis summary   
##   
## Reference intervention: HSRPM  
## Comparator intervention: EMSRP  
##   
## HSRPM dominates for all k in [0 - 300]   
##   
##   
## Analysis for willingness to pay parameter k = 300  
##   
## Expected utility  
## EMSRP -5519619  
## HSRPM -5407710  
##   
## EIB CEAC ICER  
## HSRPM vs EMSRP 111909 1 -41021  
##   
## Optimal intervention (max expected utility) for k=300: HSRPM  
##   
## EVPI -0.0000000002406