Bayesian analaysis of the DTE - at an interim analysis

September 2022

We have some data $\mathbf{x} = (x_{c,1}, \ldots, x_{c,n}, x_{t,1}, \ldots, x_{t,m})$. We are tasked with finding the most likely values for the parameters λ_1, γ_1, λ_2, γ_2 and T if we assume the data come from the following survival functions

$$S_c(t) = \exp\{-(\lambda_2 t)^{\gamma_2}\}$$

$$S_t(t) = \exp\{-(\lambda_2 t)^{\gamma_2}\}\mathbb{1}_{t \leq T} + \exp\{-(\lambda_2 T)^{\gamma_2} - \lambda_1^{\gamma_1}(t^{\gamma_1} - T^{\gamma_1})\}\mathbb{1}_{t > T}$$

We have the relationship

$$f(t) = \frac{d}{dt}[1 - S(t)]$$

Therefore, the above survival probabilities can be manipulated to give the following densities

$$f_c(t) = \frac{\gamma_2(\lambda_2 t)^{\gamma_2}\exp\{-(\lambda_2 t)^{\gamma_2}\}}{t}$$

$$f_t(t) = \frac{\gamma_2(\lambda_2 t)^{\gamma_2}\exp\{-(\lambda_2 t)^{\gamma_2}\}}{t}\mathbb{1}_{t \leq T} + \gamma_1\lambda_1^{\gamma_1}t^{\gamma_1 - 1}\exp\{-\lambda_1^{\gamma_1}(t^{\gamma_1} - T^{\gamma_1}) - (T\lambda_2)^{\gamma_2}\}\mathbb{1}_{t > T}$$

If we let $\gamma_1 = \gamma_2$ the hazard ratio is

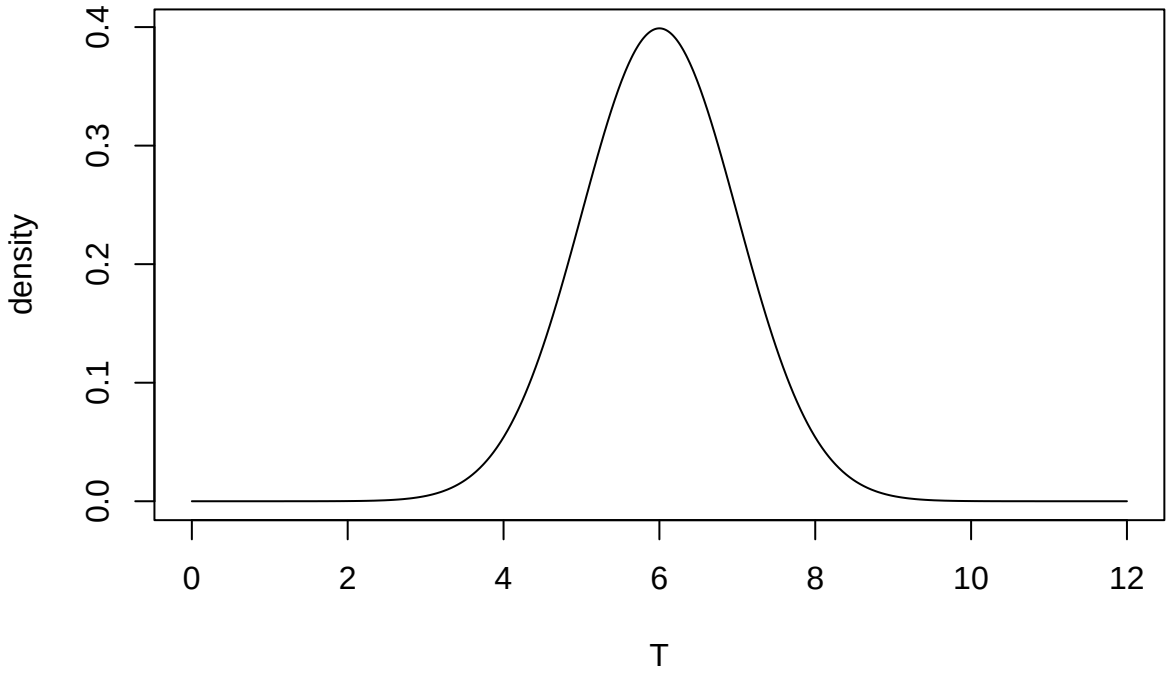
$$\text{HR} = \begin{cases} 1,, & t \leq T \\ (\frac{\lambda_1}{\lambda_2})^{\gamma_2}, & t > T \end{cases}$$

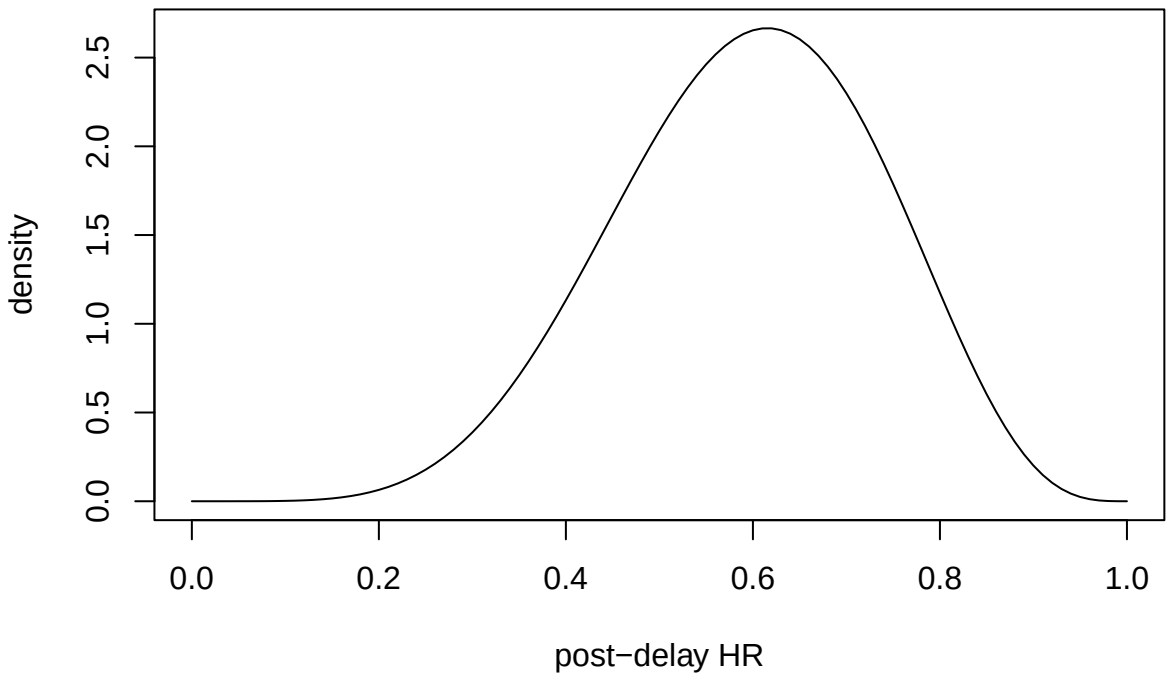
In consultation with experts, we have elicited the following distributions:

$$T \sim N(6, 1)$$

$$\text{post-delay HR} \sim Be(6.6, 4.5)$$

These distributions can be seen below

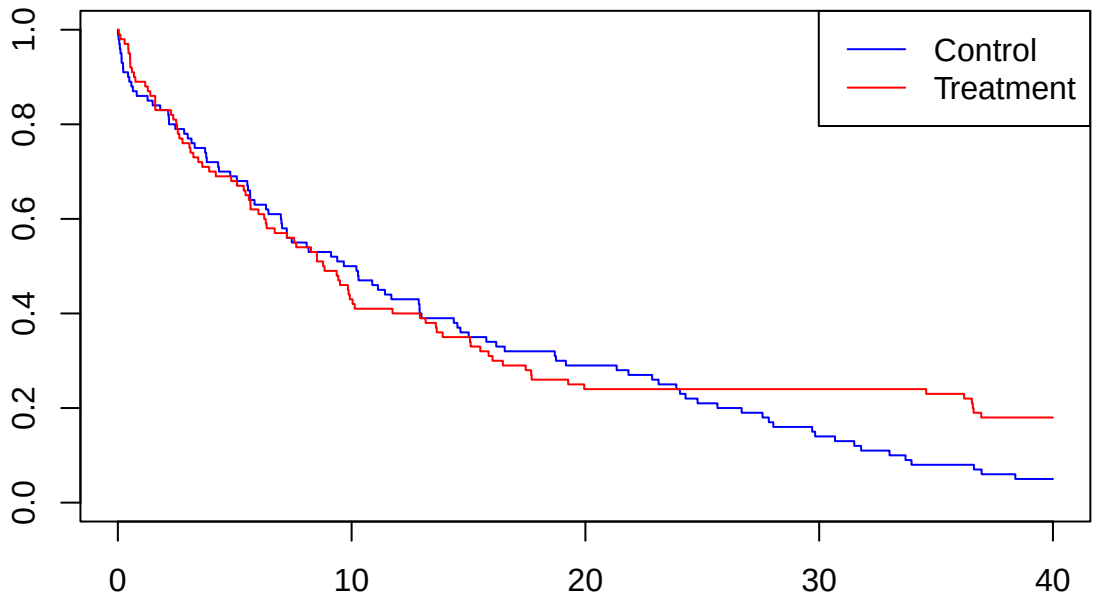

density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10
12
T



We can simulate some data according to some underlying parameters (in practice, we would not know these), combine the data with our prior distributions to obtain posterior distributions for T and post-delay HR.

```
#set.seed(53)
#Setting up the correct parameters - we would not know this in practice

lambda1 <- 0.04
lambda2 <- 0.08
gamma2 <- 0.8
gamma1 <- 0.8
bigT <- 20

#Sample sizes in each group
n1 <- 100
n2 <- 100

#When is the IA time?
IATime <- 40

#When is the final analysis time?
trialLength <- 60

#Simulating the control and treatment data - again, we would not normally know the underlying structure
#Control
controldata <- rweibull(n1, gamma2, 1/lambda2)
#Treatment
```

```r
CP <- exp(-(lambda2*bigT)^gamma2)[[1]]
u <- runif(n2)
suppressWarnings(treatmentdata <- ifelse(u>CP, (1/lambda2)*exp(1/gamma2*log(-log(u))), exp((1/gamma1)*lo

combinedData <- data.frame(time = c(controldata, treatmentdata), group = c(rep("Control", n1), rep("Trea

combinedData$event <- combinedData$time<IATime

combinedData$time[combinedData$time>IATime] <- IATime

controlkm <- survfit(Surv(time, event)~group, data = combinedData)

plot(controlkm, col=c("blue", "red"))

legend("topright", legend = c("Control", "Treatment"), col = c("blue", "red"), lty=1)
```




```r
n <- n1
m <- n1+n2

#JAGS code which calculates posterior distributions

modelstring="

data {
  for (j in 1:m){
```

```
    zeros[j] <- 0
  }
}

model {
  C <- 10000
  for (i in 1:n){
    zeros[i] ~ dpois(zeros.mean[i])
    zeros.mean[i] <-  -l[i] + C
    l[i] <- ifelse(datEvent[i]==1, log(gamma2)+gamma2*log(lambda2*datTimes[i])-(lambda2*datTimes[i])^ga
  }
  for (i in (n+1):m){
    zeros[i] ~ dpois(zeros.mean[i])
    zeros.mean[i] <-  -l[i] + C
    l[i] <- ifelse(datEvent[i]==1, ifelse(datTimes[i]<bigT, log(gamma2)+gamma2*log(lambda2*datTimes[i])
      ifelse(datTimes[i]<bigT, -(lambda2*datTimes[i])^gamma2, -(lambda2*bigT)^gamma2-lambda1^gamma2*(da
  }

    lambda2 ~ dbeta(1,1)T(0,)
    gamma2 ~ dbeta(1,1)T(0,)
    HR ~ dbeta(6.6, 4.5)T(0,1)
    bigT ~ dnorm(6, 0.01)T(0,)
    lambda1 <- lambda2*pow(HR, 1/gamma2)

    }
"

model = jags.model(textConnection(modelstring), data = list(datTimes = combinedData$time, datEvent = co
```

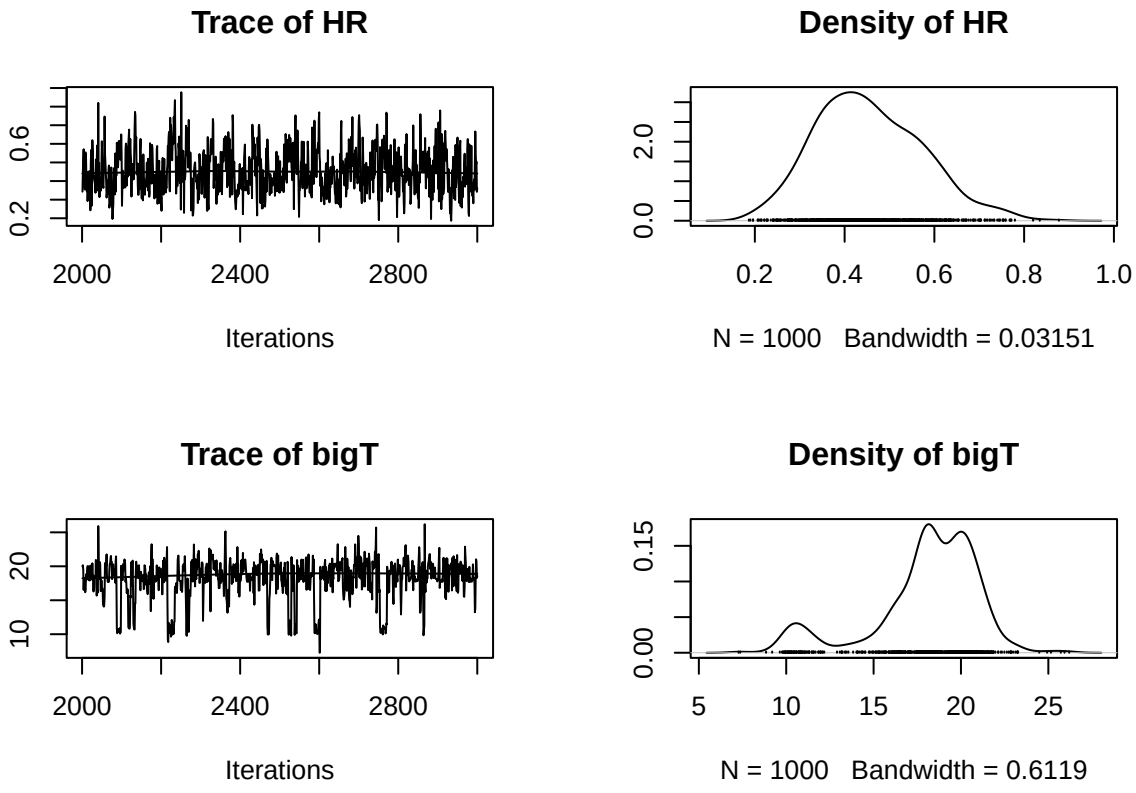
```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 200
##    Unobserved stochastic nodes: 4
##    Total graph size: 3681
##
## Initializing model
```

```
update(model, n.iter=1000)
output=coda.samples(model=model, variable.names=c("HR", "bigT"), n.iter = 1000)


summary(output)
```

```
##
## Iterations = 2001:3000
```

```
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##         Mean     SD Naive SE Time-series SE
## HR    0.4549 0.1183 0.003742       0.006577
## bigT 18.0171 3.0795 0.097382       0.250587
##
## 2. Quantiles for each variable:
##
##         2.5%     25%     50%     75%   97.5%
## HR    0.2491  0.3669  0.4426  0.5383  0.7122
## bigT 10.1676 17.0291 18.4216 20.1088 22.1167
```

```
plot(output)
```




```
#We now need to sample from these posterior distributions
#We sample future patient data

BPPvec <- rep(NA, 100)

for (i in 1:100){
```

```r
  #Calculating how many control events are remaining

noremainingcontrolevents <-  combinedData %>%
filter(group=="Control"& event==F) %>%
  count() %>%
   pull()

#What is the  survival probability at time = IAtime?

controlIAsurv <- exp(-(lambda2*IATime)^gamma2)

u <- runif(noremainingcontrolevents, 0, controlIAsurv)

#Simulating the remaining control data

remainingcontrolevents <- data.frame(time = (1/lambda2)*(-log(u))^(1/gamma2), group = rep("Control", no


#Now need to simulate the remaining treatment events
#Using the posteriors for T and HR

noremainingtreatmentevents <- combinedData %>%
filter(group=="Treatment"& event==F) %>%
  count() %>%
   pull()


sampledHR <- sample(output[[1]][,1], size =1)
sampledT <- sample(output[[1]][,2], size =1)

# print(sampledHR)
# print(sampledT)

sampledlambda1 <- lambda2*sampledHR^(1/gamma2)

#What is the  survival probability at time = IAtime?
treatmentIAsurv <- exp(-(lambda2*sampledT)^gamma2-sampledlambda1^gamma1*(IATime^gamma1-sampledT^gamma1)

u <- runif(noremainingtreatmentevents, 0, treatmentIAsurv)

#Sampling the remaining treatment events
remainingtreatmentevents <- data.frame(time = ((1/(lambda1^gamma1))*((lambda1*sampledT)^gamma1-log(u)-(

#Now we need to combine the new simulated data with the original data set

original <- combinedData[combinedData$event==T,]

finalDataSet <- rbind(original, remainingcontrolevents, remainingtreatmentevents)

finalDataSet$event <- finalDataSet$time<trialLength

finalDataSet$time[finalDataSet$time>IATime] <- trialLength
```

```
test <- survdiff(Surv(time, event)~group, data = finalDataSet)

BPPvec[i] <- test$chisq > qchisq(0.95, 1)
}

mean(BPPvec)
```

```
## [1] 0
```

There is a question of what to do at this interim analysis stage, we could use these posteriors - especially the one for HR - to look at how much more informed we are at this interim analysis stage than we were compared to before the trial. We could also sample future observations from these posterior distributions and then combine these simulated future observations with the observations seen at the interim analysis stage to calculate Bayesian predictive probabilities.