

**Appendix 1 Source code**

```

library(ldbounds)
library(dplyr)
library(LearnBayes)

N_total <- 210
n1=n2=n3=70; group <- c(70, 70, 70)
block_number <- 3
time <- c(1/3, 2/3, 1)
bounds <- bounds(time, iuse = c(1, 1), alpha = c(0.025, 0.025))
supbound <- bounds$upper.bounds
futbound <- bounds$lower.bounds
px1 <- 0.5; betavec <- c(0, 2, 0); gammavec <- c(0, 0, 0)
nsample = 10000; nburn = 5000
rndmethod="cara1"
alternative="greater"
correct=FALSE

x1all <- rbinom(N_total, 1, px1)
x2all <- rbinom(N_total, 1, 0.5)
x1 <- x1all[1:n1]
x1.2 <- x1all[(n1+1):(n1+n2)]
x1.3 <- x1all[(n1+n2+1):N_total]
x2 <- x2all[1:n1]
x2.2 <- x2all[(n1+1):(n1+n2)]
x2.3 <- x2all[(n1+n2+1):N_total]

bound_index=1
G1 <- rbinom(n1, 1, 0.5)
onevec <- rep(1, n1)
xmat1 <- cbind(onevec, x1, x2)
pn <- dim(xmat1)[2]
p1 <- pnorm(xmat1%%betavec + G1*xmat1%%gammavec)
y1 <- rbinom(n1, 1, p1)
xdata1 <- cbind(onevec, x1, x2, G1, G1*x1, G1*x2)
fit <- glm(y1 ~ xdata1-1, family = binomial(link = probit))
mle_theta <- fit$coefficients
priorb = list(beta=rep(0, length(mle_theta)),
              P=diag(rep(2, length(mle_theta))))
res1 = bayes.probit(y1, xdata1, nsample, priorb)
resbetag <- res1$beta[-(1:nburn),]
resbeta <- res1$beta[-(1:nburn),1:pn]
resbetahat <- apply(resbetag, 2, mean)
ndiff1 <- length(which(G1==1)) - length(which(G1==0))
nf1 <- length(y1)-sum(y1)
data_total = data <- data.frame()
data <- data.frame(treatment=G1, outcome=y1)

```

```

data_total <- rbind(data_total, data)
data_total$treatment <- as.factor(data_total$treatment)
data_total$outcome <- as.factor(data_total$outcome)
data_total <- data_total %>% mutate(time = fac-tor(rep(1:bound_index,group[1:bound_index])))
ctrl_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment == 0])))
trt_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment == 1])))
if (all(data_total$time == 1) | N_total/block_number < 2) {
  if (((ctrl_prop - trt_prop >= 0) & alternative == "less") | ((trt_prop - ctrl_prop >= 0) & alternative == "greater")) {
    p.val1 <- chisq.test(data_total$treatment, data_total$outcome,correct = cor-rect)$p.value/2
    test1 <- sqrt(as.numeric(chisq.test(data_total$treatment,data_total$outcome, correct = correct)$statistic))
  }
  else {
    p.val1 <- 1
    test1 <- 0
  }
}else {
  p.val1 <- mantelhaen.test(table(data_total), alternative = alternative,correct = cor-rect)$p.val
  test1 <- sqrt(as.numeric(mantelhaen.test(table(data_total),
    alternative = alternative, correct = cor-rect)$statistic))
}
if (test1 > supbound[bound_index]) {
  ind <- bound_index
  ndiff <- ndiff1
  nf <- nf1
  ind.power <- 1
  next
}else if (test1 < futbound[bound_index]) {
  ind <- bound_index
  ndiff <- ndiff1
  nf <- nf1
  ind.power <- 0
  next
}else{
  bound_index <- 2
  onevec.2 <- rep(1, n2)
  newxdata2 <- cbind(onevec.2, x1.2, x2.2, onevec.2, x1.2, x2.2)
  newxdatas2 <- newxdata2[,1:pn]

  G2 <- c()
  if(rndmethod=="cara1"){
    exppart2 <- pnorm(newxdata2%*%t(resbetag)) - pnorm(newxdatas2%*%t(resbeta))
    postp <- c()
    for (l in 1:n2){
      postp[l] <- length(which(exppart2[l,]>0))/length(exppart2[l,])
    }
    pip2 <- sqrt(postp)/(sqrt(postp)+sqrt(1-postp))
    for(l in 1:n2){
      G2[l] <- sample(1:2, 1, prob=c(pip2[l], 1-pip2[l]))
    }
  }
}

```

```

}
G2[which(G2==2)] <- 0
}
if(rndmethod=="cara2"){
  for(l in 1:n2){
    p0z.x <- c(1, x1.2[l], x2.2[l], 1, x1.2[l], x2.2[l])
    p1z.x <- c(1, x1.2[l], x2.2[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat)
    p1z <- pnorm(p1z.x%%resbetahat)
    pip2 <- sqrt(p0z)/(sqrt(p0z)+sqrt(p1z))
    G2[l] <- sample(1:2, 1, prob=c(pip2, 1-pip2))
  }
  G2[which(G2==2)] <- 0
}
if(rndmethod=="cara3"){
  for(l in 1:n2){
    p0z.x <- c(1, x1.2[l], x2.2[l], 1, x1.2[l], x2.2[l])
    p1z.x <- c(1, x1.2[l], x2.2[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat)
    p1z <- pnorm(p1z.x%%resbetahat)
    pp1 <- p0z/(1-p0z)
    pp2 <- p1z/(1-p1z)
    pip2 <- pp1/(pp1+pp2)
    G2[l] <- sample(1:2, 1, prob=c(pip2, 1-pip2))
  }
  G2[which(G2==2)] <- 0
}
if(rndmethod=="cara4"){
  for(l in 1:n2){
    p0z.x <- c(1, x1.2[l], x2.2[l], 1, x1.2[l], x2.2[l])
    p1z.x <- c(1, x1.2[l], x2.2[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat)
    p1z <- pnorm(p1z.x%%resbetahat)
    pp1 <- p0z*(1-p0z)
    pp2 <- p1z*(1-p1z)
    pip2 <- sqrt(pp2)/(sqrt(pp2)+sqrt(pp1))
    G2[l] <- sample(1:2, 1, prob=c(pip2, 1-pip2))
  }
  G2[which(G2==2)] <- 0
}

xmat2 <- cbind(onevec.2, x1.2, x2.2)
p1.2 <- pnorm(xmat2%%betavec + G2*xmat2%%gammavec)
y2 <- rbinom(n2, 1, p1.2)
xdata2 <- cbind(onevec.2, x1.2, x2.2, G2, G2*x1.2, G2*x2.2)
xdata2 <- rbind(xdata1, xdata2)
y2a <- c(y1, y2)
G2a <- c(G1, G2)

```

```

fit2 <- glm(y2a ~ xdata2-1, family = binomial(link = probit))
mle_theta2 <- fit2$coefficients
priorb = list(beta=rep(0, length(mle_theta2)),
              P=diag(rep(2, length(mle_theta2))))
res2 = bayes.probit(y2a, xdata2, nsample, priorb)
resbetag2 <- res2$beta[-(1:nburn),]
resbeta2 <- res2$beta[-(1:nburn),1:pn]
resbetahat2 <- apply(resbetag2, 2, mean)
ndiff2 <- length(which(G2==1)) - length(which(G2==0))
nf2 <- length(y2) - sum(y2)
data <- data.frame(treatment=G2, outcome=y2)
data_total <- rbind(data_total[,1:2], data)
data_total$treatment <- as.factor(data_total$treatment)
data_total$outcome <- as.factor(data_total$outcome)
data_total <- data_total %>% mutate(time = fac-tor(rep(1:bound_index,group[1:bound_index])))
ctrl_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment ==0])))
trt_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment ==1])))
if (all(data_total$time == 1) | N_total/block_number < 2) {
  if (((ctrl_prop - trt_prop >= 0) & alternative == "less") | ((trt_prop - ctrl_prop >= 0) & alternative == "greater")) {
    p.val2 <- chisq.test(data_total$treatment, data_total$outcome,
                        correct = correct)$p.value/2
    test2 <- sqrt(as.numeric(chisq.test(data_total$treatment,
                                      data_total$outcome, correct = correct)$statistic))
  }
  else {
    p.val2 <- 1
    test2 <- 0
  }
} else {
  p.val2 <- mantelhaen.test(table(data_total), alternative = alternative,
                             correct = correct)$p.val
  test2 <- sqrt(as.numeric(mantelhaen.test(table(data_total),
                                             alternative = alternative, correct = correct)$statistic))
}
if (test2 > supbound[bound_index]){
  ind <- bound_index
  ndiff <- ndiff1+ndiff2
  nf <- nf1+nf2
  ind.power <- 1
  next
} else if (test2 < futbound[bound_index]){
  ind <- bound_index
  ndiff <- ndiff1+ndiff2
  nf <- nf1+nf2
  ind.power <- 0
  next
} else{
  bound_index <- 3
}

```

```

onevec.3 <- rep(1, n3)
newxdata3 <- cbind(onevec.3, x1.3, x2.3, onevec.3, x1.3, x2.3)
newxdatas3 <- newxdata3[,1:pn]

G3 <- c()
if(rndmethod=="cara1"){
  exppart3 <- pnorm(newxdata3%%t(resbetag2)) - pnorm(newxdatas3%%t(resbeta2))
  postp <- c()
  for (l in 1:n3){
    postp[l] <- length(which(exppart3[l,>0])/length(exppart3[l,]))
  }
  pip3 <- sqrt(postp)/(sqrt(postp)+sqrt(1-postp))
  for(l in 1:n3){
    G3[l] <- sample(1:2, 1, prob=c(pip3[l], 1-pip3[l]))
  }
  G3[which(G3==2)] <- 0
}
if(rndmethod=="cara2"){
  for(l in 1:n3){
    p0z.x <- c(1, x1.3[l], x2.3[l], 1, x1.3[l], x2.3[l])
    p1z.x <- c(1, x1.3[l], x2.3[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat2)
    p1z <- pnorm(p1z.x%%resbetahat2)
    pip3 <- sqrt(p0z)/(sqrt(p0z)+sqrt(p1z))
    G3[l] <- sample(1:2, 1, prob=c(pip3, 1-pip3))
  }
  G3[which(G3==2)] <- 0
}
if(rndmethod=="cara3"){
  for(l in 1:n3){
    p0z.x <- c(1, x1.3[l], x2.3[l], 1, x1.3[l], x2.3[l])
    p1z.x <- c(1, x1.3[l], x2.3[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat2)
    p1z <- pnorm(p1z.x%%resbetahat2)
    pp1 <- p0z/(1-p0z)
    pp2 <- p1z/(1-p1z)
    pip3 <- pp1/(pp1+pp2)
    G3[l] <- sample(1:2, 1, prob=c(pip3, 1-pip3))
  }
  G3[which(G3==2)] <- 0
}
if(rndmethod=="cara4"){
  for(l in 1:n3){
    p0z.x <- c(1, x1.3[l], x2.3[l], 1, x1.3[l], x2.3[l])
    p1z.x <- c(1, x1.3[l], x2.3[l], 0, 0, 0)
    p0z <- pnorm(p0z.x%%resbetahat2)
    p1z <- pnorm(p1z.x%%resbetahat2)
    pp1 <- p0z*(1-p0z)

```

```

pp2 <- p1z*(1-p1z)
pip3 <- sqrt(pp2)/(sqrt(pp2)+sqrt(pp1))
G3[l] <- sample(1:2, 1, prob=c(pip3, 1-pip3))
}
G3[which(G3==2)] <- 0
}

xmat3 <- cbind(onevec.3, x1.3, x2.3)
p1.3 <- pnorm(xmat3%%betavec + G3*xmat3%%gammavec)
y3 <- rbinom(n3, 1, p1.3)
xdata33 <- cbind(onevec.3, x1.3, x2.3, G3, G3*x1.3, G3*x2.3)
xdata33ab <- cbind(onevec.3, x1.3, x2.3, onevec.3, x1.3, x2.3)
xdata3 <- rbind(xdata1, xdata22, xdata33)
y3a <- c(y1, y2, y3)
G3a <- c(G1, G2, G3)
fit3 <- glm(y3a ~ xdata3-1, family = binomial(link = probit))
mle_theta3 <- fit3$coefficients
priorb = list(beta=rep(0, length(mle_theta3 )),
              P=diag(rep(2, length(mle_theta3 ))))
res3 = bayes.probit(y3a, xdata3, nsample, priorb)
resbetag3 <- res3$beta[-(1:nburn),]
resbeta3 <- res3$beta[-(1:nburn),1:pn]
ndiff3 <- length(which(G3==1)) - length(which(G3==0))
nf3 <- length(y3)-sum(y3)
data <- data.frame(treatment=G3, outcome=y3)
data_total <- rbind(data_total[,1:2], data)
data_total$treatment <- as.factor(data_total$treatment)
data_total$outcome <- as.factor(data_total$outcome)
data_total <- data_total %>% mutate(time = fac-tor(rep(1:bound_index,group[1:bound_index])))
ctrl_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment ==0])))
trt_prop <- mean(as.numeric(as.character(data_total$outcome[data_total$treatment ==1])))
if (all(data_total$time == 1) | N_total/block_number < 2) {
  if (((ctrl_prop - trt_prop >= 0) & alternative == "less") | ((trt_prop - ctrl_prop >= 0) & alternative == "greater")) {
    p.val3 <- chisq.test(data_total$treatment, data_total$outcome,
                        correct = correct)$p.value/2
    test3 <- sqrt(as.numeric(chisq.test(data_total$treatment,data_total$outcome, correct = correct)$statistic))
  }
  else {
    p.val3 <- 1
    test3 <- 0
  }
}else {
  p.val3 <- mantelhaen.test(table(data_total), alternative = alternative,
                             correct = correct)$p.val
  test3 <- sqrt(as.numeric(mantelhaen.test(table(data_total),
                                           alternative = alternative,correct = correct)$statistic))
}
if (test3 > supbound[bound_index]){

```

```
ind <- 3
ndiff <- ndiff1+ndiff2+ndiff3
nf <- nf1+nf2+nf3
ind.power <- 1
}else{
ind <- 3
ndiff <- ndiff1+ndiff2+ndiff3
nf <- nf1+nf2+nf3
ind.power <- 0
}
}
}
earlyst1 <- sum(ind==1)+sum(ind==2)
earlyst2 <- sum(ind==1)+sum(ind==2) + sum(ind==3)
earlyst <- earlyst1/earlyst2
power <- sum(ind.power, na.rm=T)/ earlyst2
nf
ndiff
```