Supplementary Materials to "Assessment of Bayesian Expected Power via Bayesian Bootstrap"

Fang Liu*

Applied and Computational Mathematics and Statistics

University of Notre Dame, Notre Dame, IN 46556

*email: fang.liu.131@nd.edu

May 11, 2017

1 R codes for the Bayesian bootstrap procedure (BBS)

```
##############################################################################
# pilot: pilot data in data.frame format with subject ID = ID
# future: future study data
# n: sample size of pilot data
# N: sample size of planned future trial
# post.pi: posterior samples of the probabilities of the sample cases
# alpha: a vector of hyperparameters of dimensional n in the dirichlet prior
# (page 5 in the manuscript)
# m: number of posterior samples on pi
# t: sets of bootstrapped future data of size N per posterior sample pi
##############################################################################
library(MCMCpack)
for(i in 1:m) {
  post.pi <- rdirichlet(1, rep(1+alpha,n))
  for(k in 1:t) {
    future <- NULL
    count<- c(rmultinom(1,N,post.pi))
    cum.count<- 0;
    for(j in 1:n) {
      if(count[j] != 0) {
        for(l in 1:count[j]) {
          cum.count<- cum.count+1;
          drawY<- pilot[pilot$ID==j,]
          drawY$ID<- cum.count
          future<- rbind(future, drawY)
        }
      }
    }
  # code the planned analysis on data ``future''
  }
}
```

2   R codes for the double bootstrap procedure (BS2)

```
###############################################################################
# pilot: pilot data in data.frame format with subject ID = ID
# future: future study data
# n: sample size of pilot data
# N: sample size of planned future trial
# m: sets  of bootstrapped  pilot data of size n
# t: sets of bootstrapped future data of size N per bootstrapped sample data
###############################################################################
for (i in 1:m) {
  draw.ID<-  sample(1:n, n, replace=T)
  tmp   <- NULL
  for(j in 1:n) {
    drawY<- pilot[pilot$ID==draw.ID[j],]
    drawY$ID<- j
    tmp <- rbind(tmp , drawY)
  }
  for(k in 1:t) {
    draw.ID<- sample(1:n, N, replace=T)
    future<- NULL
    for(j in 1:N) {
      drawY<- tmp[tmp$ID==draw.ID[j],]
      drawY$ID<- j
      future <- rbind( future, drawY)
    }
    # code the planned analysis on data ``future''
  }
}
```

3   R codes for the power calculation based on multivariate t distribution with co-primary endpoints in equivalence studies

```
###############################################################################
# N: sample size
# df: degrees of freedom of the multivariate t distribution
# diff:  expected differences on the endpoints
# Sigma: variance-covariance matrix of the repeated measures;
#        note that the dimension of Sigma is 2 folds of the diff length
# lower & upper: lower and upper bounds of the equivalence hypotheses
#                that can be different across different endpoints
# alpha: Type I error rate
###############################################################################
library(mvtnorm)
pw.eq.mt<- function(N, df, diff, Sigma, lower=log(0.80),
                    upper=log(1.25), alpha=0.05)
{
  nend<-  length(diff)
  S<- matrix(0,nend,nend)
```

```
    cc<- matrix(0,nend, nrow(Sigma))
    for(i in 1:nend){
      j<- 2*(i-1)+1;
      cc[i,j:(j+1)]<-  c(1,-1)}
    S<- cc%*%Sigma%*%t(cc)/N
    crr0<-  cov2cor(S);
    l<- rep(0,nend)
    u<- rep(0,nend)
    if (length(lower)==1) lower<-  rep(lower,nend)
    if (length(upper)==1) upper<-  rep(upper,nend)

    crr<- diag(2*nend);
    for (i in 1:(nend-1)){
      k<- 2*(i-1)+1;
      crr[k:(k+1), k:(k+1)]<- 1
      for (j in (i+1):nend){
        l<- 2*(j-1)+1
        crr[k:(k+1), l:(l+1)]<- crr0[i,j]
        crr[l:(l+1), k:(k+1)]<- crr0[i,j]
      }
    }
    k<- 2*(nend-1)+1;
    crr[k:(k+1), k:(k+1)]<- 1
    D<- diag(nend)
    for (i in 1:nend) D[i,i]<- 1/sqrt(S[i,i])
    delta1<-  D%*%(diff-lower)
    delta2<-  D%*%(diff-upper)
    delta<- rep(0,2*nend)
    c<-  qt(alpha,df)
    Lb<-  rep(-c, 2*nend)
    Ub<-  rep( c, 2*nend)

    for(i in 1:nend){
      j<- 2*i-1
      delta[j]<- delta1[i]
      delta[j+1]<- delta2[i]
      Lb[j+1]<- -Inf
      Ub[j]<-  Inf
    }
    crr<- cov2cor(crr)
    pw<- pmvt(lower=Lb, upper=Ub, df=df, delta=delta, corr=crr);
    pw<- round(pw[1],4)
    return(pw)
}
# example (note that the dimension of Sigma is 2 folds of the diff length)
# pw.eq.mt(100, 5, c(0.1,-0.05), 0.16*diag(4))
```