Appendix B-1

```
# R code to calculate the approximate group sizes for testing non-inferiority/
# superiority of a linear contrast for a designated power
##########################################################
# 1. You need to specify the following values in the input part: significance level,
#    Type II error, the deviation from H0 to Ha, contrast coefficients,
#    and the corresponding unit cost, and variance for each group.
# 2. Based on different purposes, you can have different group size allocation ratios.
#    You can choose gamma=1 to minimize total cost (MC), gamma=2 to minimize total
#    sample size (MS) or gamma=3 for equal group size (ES).
# 3. We use the data from Clarke et al. (2005) and the sample size calculation is
#    demonstrated in the Section of illustrative example in this paper.
##########################################################
# --------------- procedure --------------------
                                    # begin of the function of SizeInfe
SizeInfe = function(alpha, beta, Delta, lc, cost, variance, gamma)
{
# Step 1: Calculate the group size allocation ratio and the initial group sizes
                         # Equ.11 of allocation ratio for minimal total cost
 gam1 = abs(lc/lc[1])*(variance/variance[1])^0.5*(cost[1]/cost)^0.5
 gam2 = abs(lc/lc[1])*(variance/variance[1])^0.5      # for minimal sample size
 gam3 = c(rep(1,length(lc)))                          # for equal group size
 if (gamma == 1) {gam = gam1}
   else if (gamma == 2) {gam = gam2}
     else  {gam = gam3}
 zalpha = qnorm(1-alpha)              # quantile of the standard normal distribution
 zbeta = qnorm(1-beta)
 sum0 = lc^2 %*%(variance/gam)
 zn1 = sum0*(zalpha+zbeta)^2/Delta^2   # initial first group size in Equ. 2
 zn = zn1 * gam                        # initial sizes for all groups
# Step 2: Calculate the degrees of freedom and find the quantile of the t distribution
df = function(lc, variance, ns)        # find degrees of freedom in Equ. 14
 {
   u1 = lc^2 * variance / ns
   sum1 = sum(u1)
   sum2 = sum(u1^2/(ns+1))
   df = sum1^2 / sum2 - 2
 }
 df1 = df(lc,variance,zn)
```

```
 talpha = qt(1-alpha, df1)             # quantile of the t distribution with df1
 tbeta = qt(1-beta, df1)
# Step 3: Calculate the approximate group sizes
 tn1 = sum0*(talpha+tbeta)^2/Delta^2    # approximate first group size in Equ. 15
 tn = tn1 * gam                         # approximate group sizes for all groups
 fn = ceiling(tn)                       # final group sizes after rounding up
 Ton = sum(fn)                          # the resulting total size
 Toc = fn %*% cost                      # the resulting total cost
 list(gamma=gam, sum0=sum0, zalpha=zalpha, zbeta=zbeta, size.ini=zn, df=df1,
     talpha=talpha, tbeta=tbeta, size.t=tn, size.fixpower=fn, total.size=Ton,
     total.cost=Toc)
}                                       # end of the function of SizeInfe
# -------------- input -------------------------
alpha = 0.05            # significance level
beta = 0.2              # Type II error rate
Delta = 2.2             # big Delta, the deviation from H0 to Ha
# In the following, you can change the number of elements in the parentheses based on
# the number of groups you are investigating.
lc = c(1/2,1/2,-1)       # coefficients of contrast
cost = c(20,50,100)      # unit cost for each group
variance = c(79.21,57.76,77.44)    # variance for each group
gamma = 1                # gamma = 1 for minimal total cost(MC)
                         # gamma = 2 for minimal total sample size(MS)
                         # gamma = 3 for equal group size(ES)
# ------------ output ---------------------------
SizeInfe(alpha, beta, Delta, lc, cost, variance, gamma)
```

Appendix B-2

```
# R code to calculate the approximate group sizes for testing equivalence of a linear
# contrast for a designated power
########################################################
# 1. You need to specify the following values in the input part: significance level,
#    Type II error, equivalence margins (upper & lower), linear contrast for the
#    alternative hypothesis, contrast coefficients, and the corresponding unit cost,
#    and variance for each group.
# 2. Based on different purposes, you can have different group size allocation ratios.
#    You can choose gamma=1 to minimize total cost (MC), gamma=2 to minimize total
#    sample size (MS) or gamma=3 for equal group size (ES).
# 3. We use the data from Van Wier et al.(2012) and the sample size calculation is
#    demonstrated in the Section of illustrative example in this paper.
########################################################
#--------------------- procedure ---------------------------------------
                                # begin of the function of SizeEqui
SizeEqui = function(alpha, beta, delta1, delta2, delta, lc, cost, variance, gamma)
{
# Step 1: Calculate the group size allocation ratios and the initial group sizes
                            # Equ. 11 of allocation ratio for minimal total cost
  gam1 = abs(lc/lc[1])*(variance/variance[1])^0.5*(cost[1]/cost)^0.5
  gam2 = abs(lc/lc[1])*(variance/variance[1])^0.5        # for minimal sample size
  gam3 = c(rep(1,length(lc)))                            # for equal group size
 if (gamma == 1) {gam = gam1}
   else if (gamma == 2) {gam = gam2}
     else  {gam = gam3}
 zalpha = qnorm(1-alpha)           # quantile of the standard normal distribution
 if (delta == (delta1+delta2)/2) {power = 1-beta/2       # given power in Equ. 8
      }  else  {power = 1-beta }                          # given power in Equ. 9
 zbeta = qnorm(power)
 sum0 =  lc^2 %*%(variance/gam)
 eta = min(delta - delta1, delta2 - delta)
 zn1 = sum0*(zalpha+zbeta)^2/eta^2   # initial first group size in Equ. 8 or 9
 zn = zn1 * gam                      # initial sizes for all groups
# Step 2: Calculate the degrees of freedom and find the quantile of the t distribution
df = function(lc, variance, ns)     # find degrees of freedom in Equ. 14
  {
   u1 = lc^2 * variance / ns
   sum1 = sum(u1)
```

```
  sum2 = sum(u1^2/(ns+1))
  df = sum1^2 / sum2  - 2
  }
 df1 = df(lc, variance, zn)
  talpha = qt(1-alpha, df1)       # quantile of the t distribution with df1
  tbeta = qt(power, df1)
# Step 3: Calculate the approximate group sizes
  tn1 = sum0*(talpha+tbeta)^2/eta^2     # approximate first group size in Equ.15
  tn = tn1 * gam; tn0 = tn              # approximate groups sizes for all groups
# Step 4: Check the power
tpower = function(ns)                    # power function of t test
  {
  v = sum0/ns[1]
  t1 = (delta-delta1)/v^0.5
  t2 = (delta2-delta)/v^0.5
  df = df(lc, variance, ns)
  talpha = qt(1-alpha, df)
  tpw = pt(-talpha, df, ncp=-t2)- pt(talpha, df, ncp=t1)  # Equ.16 for noncentral t
  }
  tpw = tpower(tn); tpw0 = tpw                       # tentative power
# Run iteration if the tentative power is not achieved the designated power
 while(tpw < 1-beta)
  {
  tn[1] = tn[1] + 0.2                  # increment value 0.2
  tn = tn[1] * gam
  tpw = tpower(tn)
  }
 fn = ceiling(tn)                      # final group sizes after rounding up
 Ton = sum(fn)                         # the resulting total size
 Toc = fn %*% cost                     # the resulting total cost
 list(gamma=gam, sum0=sum0, zalpha=zalpha, zbeta=zbeta, size.ini=zn, df=df1,
      talpha=talpha, tbeta=tbeta, size.t=tn0, power.ini=tpw0, power.final = tpw,
      size.fixpower=fn, total.size=Ton, total.cost=Toc)
}                                    # end of the function of SizeEqui
# -------------- input ------------------------------------------
alpha = 0.05                   # significance level
beta = 0.2                     # Type II error rate
delta1 = -2                    # lower margin
delta2 = 2                     # upper margin
```

```
delta = 0.2                     # linear contrast for the alternative hypothesis
# In the following, you can change the number of elements in the parentheses based on
# the number of groups you are investigating.
lc = c(-1,1/2,1/2)              # coefficients of contrast
cost = c(20,69,65)              # unit cost for each group
variance = c(48.79,38.10,32.81) # variance for each group
gamma = 1                       # gamma = 1 for minimal total cost (MC)
                                # gamma = 2 for minimal total sample size (MS)
                                # gamma = 3 for equal group size (ES)
# ------------ output --------------------------------------------
SizeEqui(alpha, beta, delta1, delta2, delta, lc, cost, variance, gamma)
```

Appendix B-3

```
# R code to calculate other non-fixed group sizes when some group sizes are fixed,
# for testing equivalence of a linear contrast for a designated power
######################################################
# 1. You need to specify the following values in the input part: significance level,
#    Type II error, equivalence margins (upper & lower) and linear contrast for the
#    alternative hypothesis. You also need to specify the vectors of contrast
#    coefficients of fixed group(s) and non-fixed group(s), the corresponding
#    variances and unit costs, respectively. (see Input statement)
# 2. Based on different purposes, you can have different group size allocation ratios.
#    You can choose gamma=1 to minimize total cost (MC), gamma=2 to minimize total
#    sample size (MS) or gamma=3 for equal non-fixed group size (ES).
# 3. We use the data from Van Wier et al.(2012) and the sample size calculation is
#    demonstrated in the Section of illustrative example in this paper.
######################################################
#------------------ procedure ----------------------------
                                     # begin of the function of SizeEquiFix
SizeEquiFix = function(alpha, beta, delta1, delta2, delta, lc, cost, variance, gamma)
{
# Step 1: Calculate the allocation ratios and the initial non-fixed group sizes
                            # Equ. 11 of allocation ratio for minimal total cost
  gam1 = abs(la/lc[1])*(vara / vara[1])^0.5 *(costa[1]/costa)^0.5
  gam2 = abs(la/lc[1])*(vara / vara[1])^0.5    # for minimal sample size
  gam3 = c(rep(1,length(la)))                  # for equal non-fixed group size
if (gamma == 1) {gam = gam1}
  else if (gamma == 2) {gam = gam2}
    else  {gam = gam3}
  zalpha = qnorm(1-alpha)           # quantile of the standard normal distribution
 if (delta == (delta1+delta2)/2) {power = 1-beta/2     # given power in Equ. 8
      } else {power = 1-beta }                         # given power in Equ. 9
 zbeta = qnorm(power)
 suma = la^2 %*%(vara/gam)
 eta = min(delta - delta1, delta2 - delta)
 zdd = eta^2 - (zalpha+zbeta)^2*(lb^2 %*% (varb/nb))   # zdd needs to be positive
 if (zdd > 0.5)                      # The arbitrary value 0.5 is to set the
                                     # threshold to prevent from running too long.
 zn1 = suma*(zalpha+zbeta)^2/zdd     # initial first non-fixed group size in Equ. 13
 zna = zn1 * gam                     # initial other non-fixed group sizes
 zn = c(zna, nb)                     # initial sizes for all groups
```

```
# Step 2: Calculate the degrees of freedom and find the quantile of the t distribution
df = function(lc, variance, ns)     # find degrees of freedom in Equ. 14
 {
   u1 = lc^2 * variance / ns
   sum1 = sum(u1)
   sum2 = sum(u1^2/(ns+1))
   df = sum1^2 / sum2 - 2
 }
 df1 = df(lc, variance, zn)
   talpha = qt(1-alpha, df1)        # quantile of the t distribution with df1
   tbeta = qt(power, df1)
# Step 3: Calculate the approximate group sizes
   tdd = eta^2 - (talpha+tbeta)^2*(lb^2 %*% (varb/nb))
   if (tdd > 0.5)               # approximate first non-fixed group size in Equ. 15
   tn1 = suma*(talpha+tbeta)^2/tdd
   tna = tn1 * gam              # approximate other non-fixed group sizes
   tn = c(tna, nb); tn0 = tn    # approximate sizes for all groups
# Step 4: Check the power
tpower = function(ns)           # power function of t test
 {
   v = suma/ns[1] + lb^2 %*% varb/nb
   t1 = (delta-delta1)/v^0.5
   t2 = (delta2-delta)/v^0.5
   df = df(lc, variance, ns)
   talpha = qt(1-alpha, df)
   tpw = pt(-talpha ,df, ncp=-t2)- pt(talpha ,df, ncp=t1)   # Equ. 16 for noncentral t
 }
   tpw = tpower(tn); tpw0 = tpw                             # tentative power
# Run iteration if the tentative power is not achieved the designated power
  while(tpw < 1-beta)
  {
   tn[1] = tn[1] + 0.2                         # increment value 0.2
   tn = c(tn[1]*gam, nb)
   tpw = tpower(tn)
  }
 fn = ceiling(tn)                              # final group sizes after rounding up
 Ton = sum(fn)                                 # the resulting total size
 Toc = fn %*% cost                             # the resulting total cost
 list(gamma=gam, suma=suma, zalpha=zalpha, zbeta=zbeta, size.ini=zn, df=df1,
```

```
     talpha=talpha,tbeta=tbeta, size.t=tn0, power.ini=tpw0, power.final = tpw,
     size.fixpower=fn, total.size=Ton, total.cost=Toc)
}                                          # end of the function of SizeEquiFix
#--------------- input --------------------------
alpha = 0.05           # significance level
beta = 0.2             # Type II error rate
delta1 = -2            # lower margin
delta2 = 2             # upper margin
delta = 0.2            # linear contrast for the alternative hypothesis
# In the following, we use b to represent the fixed group, a to represent other
# non-fixed groups. For example, if we fix the first group size, nb = c(200), then
# the corresponding coefficients of contrast for the fixed group lb = c(-1), and for
# other non-fixed groups la = c(1/2,1/2). You can also change the nummber of elements
# in the parentheses (vector) based on the number of groups you are investigating.
# Note that if the fixed group size is not sufficient, the program will print the
# error message "tn1 not found" and will be terminated.
nb = c(200)              # fixed group size n1 = 200
la = c(1/2,1/2)          # coefficients for other non-fixed groups
lb = c(-1)               # coefficient for the fixed group
lc = c(la,lb)            # coefficients for all groups
costa = c(69,65)         # unit costs for other non-fixed groups
costb = c(20)            # unit cost for the fixed group
cost = c(costa, costb)   # unit costs for all groups
vara = c(38.10,32.81)    # variances for other non-fixed groups
varb = c(48.79)          # variance for the fixed group
variance = c(vara, varb)
gamma =  1               # gamma = 1 for minimal total cost
                         # gamma = 2 for minimal sample size
                         # gamma = 3 for equal group size
#------------- output ---------------------------
SizeEquiFix(alpha, beta, delta1, delta2, delta, lc, cost, variance, gamma)
```