Heterogeneity Assessment Basket Trial

Kee-Young Shin

```
library(parallel)

# create function for one trial
trial = function(it, h, n1.goal, K, n2.0.goal, rS, rC, n2.1.goal, theta_0,
alphaS, alphaC, gamma, theta_a){
  prob = c(rep(theta_a,h), rep(theta_0,K-h))
  # subjects in first stage
  placement = sample(1:K, n1.goal, replace = T)
  n1 = c(length(placement[placement==1]), length(placement[placement==2]),
         length(placement[placement==3]), length(placement[placement==4]),
         length(placement[placement==5]))

  # generate number of responses
  yes_s1 = rbinom(K,n1,prob)
  no_s1 = n1 - yes_s1

  # test of homogeneity
  tab = matrix(c(yes_s1,no_s1),nrow = K,byrow = F)
  p_val = ifelse(sum(yes_s1)==0, fisher.test(tab)$p.value,
fisher.test(tab,hybrid = T,simulate.p.value=T)$p.value) # don't simulate pval
if marginal of all yes = 0
  toh = as.numeric(p_val <= gamma)

  yes_s2 = rep(NA,K); dec = rep(0,K); K_star = {}; stage2_t2 = 0; stage2_t1 =
rep(0,K); n2_0 = rep(NA,K); n2_1 = rep(NA,K)

  if (toh == 1){
    # determine which baskets to keep
    K_star = which(yes_s1 >= rS) # keep baskets with min desirable RR
    stage2_t1[K_star] = 1 # keep track of baskets that go on
    if (length(K_star) > 0){
      n2_1[K_star] = floor(n2.1.goal/K)

      # stage 2
      yes_s2[K_star] = rbinom(length(K_star),n2_1[K_star],prob[K_star])
      no_s2 = n2_1 - yes_s2

      # decision to reject in each basket
      for (i in 1:length(K_star)){
        dec[K_star[i]] =
          as.numeric(binom.test(yes_s1[K_star[i]]+yes_s2[K_star[i]],
                                n1[K_star[i]]+n2_1[K_star[i]],
                                theta_0,alternative = "greater")$p.value <=
```

```
                    (alphaS/length(K_star)) ) }
    }
  } else if (toh==0) {

      # keeping track if baskets go on to stage 2
      stage2_t2 = ifelse(sum(yes_s1) >= rC,1,0)

      if (stage2_t2 == 1){
        placement_t2 = sample(1:K, n2.0.goal, replace = T)
        n2_0 = c(length(placement_t2[placement_t2==1]),
                 length(placement_t2[placement_t2==2]),
                 length(placement_t2[placement_t2==3]),
                 length(placement_t2[placement_t2==4]),
                 length(placement_t2[placement_t2==5]))

      # generate Stage 2 responses
      yes_s2 = rbinom(K,n2_0,prob)
      no_s2 = n2_0 - yes_s2

      # decision to reject one sample
      dec = rep(ifelse(as.numeric(binom.test(sum(yes_s1+yes_s2),
                                             sum(n1+n2_0),
                                             theta_0,
                                             alternative = "greater")$p.value
                                  <= (alphaC)),1,0),K)
      }
    }
  # end trial
  return(c(dec, yes_s1, yes_s2, length(K_star), toh,
           stage2_t1,stage2_t2,n1,n2_0))
}

gamma = seq(0.1,0.9,by = 0.05)
alphaS = seq(0.01,0.1,by = 0.01)
alphaC = seq(0.01,0.05, by = 0.01)

### search grid for 3 alphas: error rates ###
full.set.index = cbind(rep(1:length(gamma),each = length(alphaS) *
length(alphaC)), rep(rep(1:length(alphaS),each = length(alphaC)),times =
length(gamma)), rep(1:length(alphaC),times = length(gamma)*length(alphaS)))
for (it in 1:length(gamma)){
  Full.set =
cbind(gamma[full.set.index[,1]],alphaS[full.set.index[,2]],alphaC[full.set.in
dex[,3]])
}

# simulations function
sim_heterogeneity = function(h, sim, n1.goal, K, n2.0.goal, rS, rC,
n2.1.goal, theta_0, theta_a, alphaS, alphaC, gamma){
```

```r
  set.seed(31)
  result = sapply(1:sim, trial, h = h, K = K, n1.goal = n1.goal, n2.1.goal =
n2.1.goal, n2.0.goal = n2.0.goal, theta_0 = theta_0, theta_a = theta_a, rS =
rS, rC = rC, gamma = gamma, alphaS = alphaS, alphaC = alphaC)

  # Marginal Rejection Probability
  mrp = rowSums(result[(1:(K)),],na.rm = T)/sim

  # FWER
  fwer = ifelse(h < K - 1, sum(as.numeric(colSums(result[((1+h):(K)),],na.rm
= T) >= 1))/sim, ifelse( h == K-1, sum(result[((1+h):(K)),],na.rm = T)/sim,
NA )) # only one basket under the null

  # number of trials each track
  toh1 = which(result[(3*K+2),] == 1) # Track 1
  toh0 = which(result[(3*K+2),] == 0) # Track 2
  results1 = result[,toh1]
  results0 = result[,toh0]
  track1 = length(toh1)
  track2 = length(toh0)

  # count of stage 2 for each track ###
  stage2_t1 = rowSums(result[(3*K+3):(4*K+2),],na.rm = T)
  stage2_t2 = sum(result[(4*K+3),],na.rm = T)

  # EN
  n1.obs = rowSums(result[(4*K+4):(5*K+3),])/sim
  n2.0.obs = rowSums(result[(5*K+4):(6*K+3),],na.rm = T)/stage2_t2
  temp = (n2.1.goal/K)*(stage2_t1/track1)
  EN = sum(n1.obs)+(track1/sim)*sum(temp)+(track2/sim)*
    (sum(n2.0.obs)*(stage2_t2/track2) )


  return(c(mrp, fwer, EN))
}

# create function with pooling for heterogeneous track
trial2 = function(it, h, n1.goal, K, n2.0.goal, rS, rC, n2.1.goal, theta_0,
alphaS, alphaC, gamma, theta_a){
  prob = c(rep(theta_a,h), rep(theta_0,K-h))
  # subjects in first stage
  placement = sample(1:K, n1.goal, replace = T)
  n1 = c(length(placement[placement==1]), length(placement[placement==2]),
         length(placement[placement==3]), length(placement[placement==4]),
         length(placement[placement==5]))

  # generate number of responses
  yes_s1 = rbinom(K,n1,prob)
  no_s1 = n1 - yes_s1
```

```r
  # test of homogeneity
  tab = matrix(c(yes_s1,no_s1),nrow = K,byrow = F)
  p_val = ifelse(sum(yes_s1)==0, fisher.test(tab)$p.value,
fisher.test(tab,hybrid = T,simulate.p.value=T)$p.value) # don't simulate pval
if marginal of all yes = 0
  toh = as.numeric(p_val <= gamma)

  yes_s2 = rep(NA,K); dec = rep(0,K); K_star = {}; stage2_t2 = 0; stage2_t1 =
rep(0,K); n2_0 = rep(NA,K); n2_1 = rep(NA,K)

  if (toh == 1){
    # determine which baskets to keep
    K_star = which(yes_s1 >= rS) # keep baskets with min desirable RR
    stage2_t1[K_star] = 1 # keep track of baskets that go on
    if (length(K_star) > 0){
      n2_1[K_star] = floor(n2.1.goal/K)

      # stage 2
      yes_s2[K_star] = rbinom(length(K_star),n2_1[K_star],prob[K_star])
      no_s2 = n2_1 - yes_s2


      if (length(K_star)>3){
      # pool kept baskets
      dec[K_star] = ifelse(
        as.numeric(binom.test(sum(yes_s1)+sum(yes_s2[K_star], na.rm = T),
                              sum(n1)+sum(n2_1[K_star], na.rm = T),
                                          theta_0,
                                          alternative = "greater")$p.value
                                  <= (alphaC)),1,0) }
      else {
      # decision to reject in each basket
      for (i in 1:length(K_star)){
        dec[K_star[i]] =
          as.numeric(binom.test(yes_s1[K_star[i]]+yes_s2[K_star[i]],
                                n1[K_star[i]]+n2_1[K_star[i]],
                                theta_0,alternative = "greater")$p.value <=
                     (alphaS/length(K_star)) ) }
      }
    }
  } else if (toh==0) {

      # keeping track if baskets go on to stage 2
      stage2_t2 = ifelse(sum(yes_s1) >= rC,1,0)

      if (stage2_t2 == 1){
        placement_t2 = sample(1:K, n2.0.goal, replace = T)
        n2_0 = c(length(placement_t2[placement_t2==1]),
```

```
                  length(placement_t2[placement_t2==2]),
                  length(placement_t2[placement_t2==3]),
                  length(placement_t2[placement_t2==4]),
                  length(placement_t2[placement_t2==5]))

      # generate Stage 2 responses
      yes_s2 = rbinom(K,n2_0,prob)
      no_s2 = n2_0 - yes_s2

      # decision to reject one sample
      dec = rep(ifelse(as.numeric(binom.test(sum(yes_s1+yes_s2),
                                             sum(n1+n2_0),
                                             theta_0,
                                             alternative = "greater")$p.value
                                  <= (alphaC)),1,0),K)
      }
    }
  # end trial
  return(c(dec, yes_s1, yes_s2, length(K_star), toh,
           stage2_t1,stage2_t2,n1,n2_0,n2_1))
}

# simulation function for pooling of hetergenous baskets
sim_heterogeneity2 = function(h, sim, n1.goal, K, n2.0.goal, rS, rC,
n2.1.goal, theta_0, theta_a, alphaS, alphaC, gamma){

  set.seed(31)
  result = sapply(1:sim, trial2, h = h, K = K, n1.goal = n1.goal, n2.1.goal =
n2.1.goal, n2.0.goal = n2.0.goal, theta_0 = theta_0, theta_a = theta_a, rS =
rS, rC = rC, gamma = gamma, alphaS = alphaS, alphaC = alphaC)

  # Marginal Rejection Probability
  mrp = rowSums(result[(1:(K)),],na.rm = T)/sim

  # FWER
  fwer = ifelse(h < K - 1, sum(as.numeric(colSums(result[((1+h):(K)),],na.rm
= T) >= 1))/sim, ifelse( h == K-1, sum(result[((1+h):(K)),],na.rm = T)/sim,
NA )) # only one basket under the null

  # number of trials each track
  toh1 = which(result[(3*K+2),] == 1) # Track 1
  toh0 = which(result[(3*K+2),] == 0) # Track 2
  results1 = result[,toh1]
  results0 = result[,toh0]
  track1 = length(toh1)
  track2 = length(toh0)

  # count of stage 2 for each track ###
  stage2_t1 = rowSums(result[(3*K+3):(4*K+2),],na.rm = T)
```

```
  stage2_t2 = sum(result[(4*K+3),],na.rm = T)

  # EN
  n1.obs = rowSums(result[(4*K+4):(5*K+3),])/sim
  n2.0.obs = rowSums(result[(5*K+4):(6*K+3),],na.rm = T)/stage2_t2
  temp = (n2.1.goal/K)*(stage2_t1/track1)
  EN = sum(n1.obs)+(track1/sim)*sum(temp)+(track2/sim)*
    (sum(n2.0.obs)*(stage2_t2/track2) )


  return(c(mrp, fwer, EN))
}

# testing with gamma=0.52

# no active baskets
test = sim_heterogeneity(h = 0, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
25, n2.0.goal = 75, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test

## [1]  0.028  0.028  0.027  0.030  0.028  0.057 64.090

# one active baskets
test1 = sim_heterogeneity(h = 1, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test1

## [1]  0.705  0.054  0.052  0.049  0.053  0.070 77.565

# two active baskets
test2 = sim_heterogeneity(h = 2, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test2

## [1]  0.795  0.801  0.099  0.097  0.101  0.119 85.420

# three active baskets
test3 = sim_heterogeneity(h = 3, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test3

## [1]  0.828  0.805  0.818  0.151  0.151  0.157 89.065

# four active baskets
test4 = sim_heterogeneity(h = 4, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
```

```
alphaC = .05, theta_a = 0.45)
test4
```

```
## [1]  0.869  0.836  0.832  0.858  0.269  0.269 88.540
```

```
# all active baskets
test5 = sim_heterogeneity(h = 5, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test5
```

```
## [1]  0.884  0.873  0.881  0.879  0.871     NA 80.485
```

```
# testing with gamma=0.2

# no active baskets
test = sim_heterogeneity(h = 0, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
25, n2.0.goal = 75, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test
```

```
## [1]  0.049  0.043  0.044  0.041  0.043  0.076 75.505
```

```
# one active baskets
test1 = sim_heterogeneity(h = 1, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test1
```

```
## [1]  0.579  0.146  0.146  0.145  0.147  0.158 67.145
```

```
# two active baskets
test2 = sim_heterogeneity(h = 2, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test2
```

```
## [1]  0.744  0.755  0.291  0.286  0.286  0.297 73.070
```

```
# three active baskets
test3 = sim_heterogeneity(h = 3, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test3
```

```
## [1]  0.863  0.859  0.861  0.420  0.421  0.424 76.190
```

```
# four active baskets
test4 = sim_heterogeneity(h = 4, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test4
```

```
## [1]  0.893  0.895  0.908  0.890  0.549  0.549 74.180
```

```
# all active baskets
test5 = sim_heterogeneity(h = 5, sim = 1000, K = 5, n1.goal = 35, n2.1.goal =
75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .2, alphaS = .07,
alphaC = .05, theta_a = 0.45)
test5
```

```
## [1]  0.962  0.953  0.943  0.954  0.954     NA 63.660
```

```
# testing with pooling of hetergeneous baskets
# no active baskets
test2 = sim_heterogeneity2(h = 0, sim = 1000, K = 5, n1.goal = 35, n2.1.goal
= 25, n2.0.goal = 75, theta_0 = .15, rS = 1, rC = 5, gamma = .52, alphaS =
.07, alphaC = .05, theta_a = 0.45)
test2
```

```
## [1]  0.034  0.029  0.032  0.033  0.036  0.054 64.090
```

```
# one active baskets
test2_1 = sim_heterogeneity2(h = 1, sim = 1000, K = 5, n1.goal = 35,
n2.1.goal = 75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52,
alphaS = .07, alphaC = .05, theta_a = 0.45)
test2_1
```

```
## [1]  0.623  0.242  0.247  0.230  0.230  0.297 77.565
```

```
# two active baskets
test2_2 = sim_heterogeneity2(h = 2, sim = 1000, K = 5, n1.goal = 35,
n2.1.goal = 75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52,
alphaS = .07, alphaC = .05, theta_a = 0.45)
test2_2
```

```
## [1]  0.856  0.857  0.506  0.505  0.483  0.622 85.420
```

```
# three active baskets
test2_3 = sim_heterogeneity2(h = 3, sim = 1000, K = 5, n1.goal = 35,
n2.1.goal = 75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52,
alphaS = .07, alphaC = .05, theta_a = 0.45)
test2_3
```

```
## [1]  0.913  0.916  0.918  0.633  0.602  0.793 89.065
```

```
# four active baskets
test2_4 = sim_heterogeneity2(h = 4, sim = 1000, K = 5, n1.goal = 35,
n2.1.goal = 75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52,
alphaS = .07, alphaC = .05, theta_a = 0.45)
test2_4
```

```
## [1]  0.968  0.959  0.954  0.969  0.664  0.664 88.540
```

```
# all active baskets
test2_5 = sim_heterogeneity2(h = 5, sim = 1000, K = 5, n1.goal = 35,
```

```
n2.1.goal = 75, n2.0.goal = 20, theta_0 = .15, rS = 1, rC = 5, gamma = .52,
alphaS = .07, alphaC = .05, theta_a = 0.45)
test2_5
```

```
## [1]  0.964  0.963  0.960  0.972  0.965     NA 80.485
```