BS_HW2

KIM SANG HYUN(202211545)

2025-04-15

Contents

```r
set.seed(12345)
y <- matrix(rbinom(10000 * 1000, 1, 0.1), 10000, 1000)
x <- matrix(rnorm(10000 * 1000), 10000, 1000)
x[y==1] <- rnorm(sum(y==1), mean=2)
```

1. For 10,000 genes each simulation replication, first find the significant genes of three tests at a significance level of 0.05, i.e., $\alpha = 0.05$. Next, compute the proportion of significant genes among 10,000 genes for each test. Finally, report the averaged proportion of the significant genes over 1,000 replications for each test. Note that the true proportion of false H_0 is 0.1.

```r
alpha = 0.05

pvalue_t_20 = 2 * (1 - pt(abs(x), df = 20))
pvalue_t_50 = 2 * (1 - pt(abs(x), df = 50))
pvalue_z = 2 * (1 - pnorm(abs(x)))

#apply(pvalue_t_20 < alpha, 2, sum)
#apply(pvalue_t_50 < alpha, 2, sum)
#apply(pvalue_z   < alpha, 2, sum)

prop_t_20 = apply(pvalue_t_20 < alpha, 2, mean)
prop_t_50 = apply(pvalue_t_50 < alpha, 2, mean)
prop_z = apply(pvalue_z   < alpha, 2, mean)

mean_t_20 = mean(prop_t_20)
mean_t_50 = mean(prop_t_50)
mean_z = mean(prop_z)

ans_1 = data.frame(
  t_20 = mean_t_20,
  t_50 = mean_t_50,
  z = mean_z
)
rownames(ans_1) = "prp_mean"

ans_1
```

```
##               t_20    t_50         z
## prp_mean 0.0799926 0.08995 0.0967012
```

2. For each gene, compute the type I error rate (false positive rate, FPR) at three different levels of $\alpha = 0.01$, 0.05 and 0.1. FPR is the proportion of rejecting H_0 when H_0 is actually true, equivalently, the total number of rejections divided by the number of simulation replications when $y = 0$. After calculating the FPRs of 10,000 genes, report the mean value of FPRs for each test and each level.

```
alphas = c(0.01, 0.05, 0.1)
p_v = list(T20 = pvalue_t_20, T50 = pvalue_t_50, Z = pvalue_z)

ans_2 = matrix(NA, nrow = length(p_v), ncol = length(alphas))
rownames(ans_2) = c("t_20", "t_50", "z")
colnames(ans_2) = paste("alpha", alphas, sep = "_")


for (i in 1:length(alphas)) {
  alpha = alphas[i]
  for (j in 1:length(p_v)) {
    pv = p_v[[j]]
    tmp = NULL
    for(k in 1:10000) {
      tmp_y = y[k, ]
      tmp[k] = sum(tmp_y[pv[k,] < alpha] == 0) / sum(tmp_y == 0)
    }
    ans_2[j, i] = mean(tmp)
  }
}

ans_2
```

```
##       alpha_0.01 alpha_0.05  alpha_0.1
## t_20 0.004425059 0.03703651 0.08462776
## t_50 0.007396965 0.04466808 0.09379949
## z    0.009996210 0.05003354 0.10006209
```

3. For each gene, compute the statistical power (true positive rate, TPR) at three different levels of $\alpha = 0.01$, 0.05 and 0.1. TPR is the proportion of rejecting H_0 when H_0 is actually false, equivalently, the total number of rejections divided by the number of simulation replications when $y = 1$. After calculating the TPRs of 10,000 genes, report the mean value of TPRs for each test and each level.

```
ans_3 = matrix(NA, nrow = length(p_v), ncol = length(alphas))
rownames(ans_3) = c("t_20", "t_50", "z")
colnames(ans_3) = paste("alpha", alphas, sep = "_")

for (i in 1:length(alphas)) {
  alpha = alphas[i]
  for (j in 1:length(p_v)) {
    pv = p_v[[j]]
    tmp = NULL
    for (k in 1:10000) {
      tmp_y = y[k, ]
      tmp[k] = sum(tmp_y[pv[k,] < alpha] == 1) / sum(tmp_y == 1)
    }
    ans_3[j, i] = mean(tmp)
  }
}

ans_3
```

```
##      alpha_0.01 alpha_0.05 alpha_0.1
## t_20  0.1993276  0.4658230 0.6082822
## t_50  0.2496102  0.4966718 0.6268997
## z     0.2828594  0.5159006 0.6387333
```

4. For each simulation replication, first compute Bonferroni adjusted p-values and find the significant genes of three tests at three different FWER levels of 0.01, 0.05 and 0.1. Next, see if there is at least one type I error (false positive, FP) among the significant genes for each test and each level. It is not necessary to count the total number of FPs but to know whether at least one FP or not. Finally, report the proportion of replications with at least one FP among 1,000 replications for each test and each level.

```
alphas = c(0.01, 0.05, 0.1)
p_v = list(T20 = pvalue_t_20, T50 = pvalue_t_50, Z = pvalue_z)

ans_4 = matrix(NA, nrow = length(p_v), ncol = length(alphas))
rownames(ans_4) = c("t_20", "t_50", "z")
colnames(ans_4) = paste("FWER", alphas, sep = "_")

for (i in 1:length(alphas)) {
  alpha = alphas[i]

  for (j in 1:length(p_v)) {
    pv = p_v[[j]]
    fp = NULL

    for (k in 1:1000) {
      multi_pvals = pv[, k]
      adj = p.adjust(multi_pvals, method = "bonferroni")
      signifi = which(adj < alpha)
      fp[k] = ifelse((length(signifi) > 0) & any(y[signifi, k] == 0), 1, 0)

    }
    ans_4[j, i] = mean(fp)
  }
}
ans_4
```

```
##      FWER_0.01 FWER_0.05 FWER_0.1
## t_20     0.000     0.000    0.000
## t_50     0.000     0.004    0.008
## z        0.009     0.042    0.084
```

5. For each simulation replication, first compute adjusted p-values using the BH method and find the significant genes of three tests at five different FDR levels of 0.01, 0.05, 0.1, 0.2 and 0.3. Next, compute the false discovery rates (FDR) of three different tests and five different levels. The FDR of each test is equivalent to the number of false discoveries (FD) among the significant genes divided by the total number of significant genes at each level. Finally, report the mean value of FDR over 1,000 replications for each test and each level. If no significant genes are identified, i.e., no rejections among 10,000 tests, FDR cannot be defined. Therefore, exclude the replication case of no rejection for computation of the mean FDR. For example, if there are no rejections 90 times among 1,000 replications for the Z-test with a level of 0.05, the mean FDR of the Z-test at 0.05 should be averaged over only 910 replications.

```
fdr_levels = c(0.01, 0.05, 0.1, 0.2, 0.3)
p_v = list(T20 = pvalue_t_20, T50 = pvalue_t_50, Z = pvalue_z)

ans_5 = matrix(NA, nrow = length(p_v), ncol = length(fdr_levels))
rownames(ans_5) = c("t_20", "t_50", "z")
colnames(ans_5) = paste("FDR", fdr_levels, sep = "_")

for (j in 1:length(p_v)) {
  pv = p_v[[j]]

  for (i in 1:length(fdr_levels)) {
    fdr_level = fdr_levels[i]
    fdr = NULL

    for (k in 1:1000) {
      multi_pvals = pv[, k]
      adj = p.adjust(multi_pvals, method = "BH")
      signifi = which(adj < fdr_level)
      fdr[k] = ifelse(length(signifi) == 0, NA, sum(y[signifi, k] == 0) / length(signifi))
    }
    ans_5[j, i] = mean(fdr, na.rm = TRUE)
  }
}

ans_5
```

```
##        FDR_0.01    FDR_0.05    FDR_0.1     FDR_0.2    FDR_0.3
## t_20 0.00000000 0.000000000 0.00000000 0.008248924 0.05540648
## t_50 0.00000000 0.008877082 0.03430960 0.107847040 0.19615274
## z    0.01057778 0.047749745 0.09048853 0.179639239 0.26967016
```