Sardana_Module7HW

Prachi Sardana

2023-02-27

Problem 1

```
# loaded data golub and package multest


data(golub, package='multtest')

gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL","AML"))
gene_expression <- nrow(golub)
p.values <- rep(NA, gene_expression)
# used for loop calculate the p- value and performed wilcoxon test
for(i in 1:gene_expression){
  p.values[i] <- wilcox.test(golub[i,gol.fac=="ALL"],golub[i,gol.fac=="AML"], paired = FALSE,
                              alternative = "greater")$p.value
}
```

```
## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties

## Warning in wilcox.test.default(golub[i, gol.fac == "ALL"], golub[i, gol.fac ==
## : cannot compute exact p-value with ties
```

```r
# False discovery rate
p.fdr <- p.adjust(p=p.values, method = "fdr")
alpha = 0.05
x <- sum(p.fdr < alpha)
x
```

```
## [1] 407
```

```r
wilcoxon_p <- apply(golub, 1, function(x) wilcox.test(x ~ gol.fac)$p.value)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
top_three <- order(p.values,decreasing = FALSE)
# top 3 genes with smallest p values
golub.gnames[top_three[1:3],2]
```

```
## [1] "TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)"
## [2] "Macmarcks"
## [3] "VIL2 Villin 2 (ezrin)"
```

Problem set 2

```
# loaded golub data and package multtest
data(golub, package='multtest')
# factored golub as ALL and AML
gol.fac <- factor(golub.cl,levels=0:1, labels= c("ALL","AML"))
gol_aml <- golub[,gol.fac == "AML"]

# applied shapiro wilk test for normality
shapiro <- apply(gol_aml, 1, function(x) shapiro.test(x)$p.value)
# performed fdr adjustment
p.fdr <- p.adjust(shapiro,method = "fdr")
# calculating the genes that do not pass the test at 0.05 level
p_fdr <- sum(p.fdr < 0.05)
p_fdr
```

```
## [1] 225
```

Problem set 3

```
# Load the 'multtest' package
data(golub, package='multtest')
gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL", "AML"))

# the CD_33 gene is located at 808 in golb , factor as ALL
CD33_gene <- golub[808, gol.fac== "ALL"]

# The HOXA9_gene is located at 1391 level, factor as ALL
HOXA9_gene <- golub[1391, gol.fac=="ALL"]

# calculated wilcox.test to check if both the genes express at same level.
wilcox.test(x = HOXA9_gene, y = CD33_gene, paired = TRUE, alternative =
              "two.sided")
```

```
## Warning in wilcox.test.default(x = HOXA9_gene, y = CD33_gene, paired = TRUE, :
## cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  HOXA9_gene and CD33_gene
## V = 62, p-value = 0.01242
## alternative hypothesis: true location shift is not equal to 0
```

```
# the genes are not expressed at the same level as the p- value is 0.01242
```

Problem set 4

```
# UCBAdmissions
str(UCBAdmissions)
```

```
##  'table' num [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Gender: chr [1:2] "Male" "Female"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```
# Set the significance level
alpha  <- 0.05

# To perform chi-squared tests
for (i in 1:6){
  # Contigence table for current department
  table <- UCBAdmissions[,,i]

  # To perform Chi-square test
  chi_square <- chisq.test(table)

  # extracting p value from the test results
  p_value <- chi_square$p.value

  # print the results
  cat("Department",i,"\n")
  cat("Contigency table: \n")
  print(table)
  cat("p_value = ", p_value,"\n")
  if (p_value < alpha){
    cat("There is significant evidence of relationship between admission decision and gender department
  }else {
    cat ("There is no significant evidence of relationship between admission decision and gender depart
  }
  cat("\n")
}
```

```
## Department 1
## Contigency table:
##           Gender
## Admit      Male Female
##   Admitted  512     89
##   Rejected  313     19
## p_value =  5.205468e-05
## There is significant evidence of relationship between admission decision and gender department 1
##
## Department 2
## Contigency table:
##           Gender
```

```
## Admit      Male Female
##   Admitted  353     17
##   Rejected  207      8
## p_value =  0.7705041
## There is no significant evidence of relationship between admission decision and gender department 2
##
## Department 3
## Contigency table:
##           Gender
## Admit      Male Female
##   Admitted  120    202
##   Rejected  205    391
## p_value =  0.4261753
## There is no significant evidence of relationship between admission decision and gender department 3
##
## Department 4
## Contigency table:
##           Gender
## Admit      Male Female
##   Admitted  138    131
##   Rejected  279    244
## p_value =  0.6378283
## There is no significant evidence of relationship between admission decision and gender department 4
##
## Department 5
## Contigency table:
##           Gender
## Admit      Male Female
##   Admitted   53     94
##   Rejected  138    299
## p_value =  0.3686981
## There is no significant evidence of relationship between admission decision and gender department 5
##
## Department 6
## Contigency table:
##           Gender
## Admit      Male Female
##   Admitted   22     24
##   Rejected  351    317
## p_value =  0.6403817
## There is no significant evidence of relationship between admission decision and gender department 6
```

Problem set 5

After computing the p-value as the proportion of permutation test statistics that are less than or equal to the observed test statistic. This gives the one-sided p-value for the hypothesis that the variance in the ALL group is smaller than the variance in the AML group

```
# Library mulltest
library(multtest)
```

```
## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
data(golub,package = "multtest")
# factored golub as ALL ,AML
gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL", "AML"))
# loaded CD33 gene data
CD33_data <- golub[808,]
n <- length(CD33_data)
# observed statistics
T.obs <- (var(CD33_data[gol.fac=="ALL"]))^2/(var(CD33_data[gol.fac=="AML"]))^2
# Number of permutations = 2000
n.perm <- 2000
# permuted statistics
T.perm <- rep(NA, n.perm)

# used for loop to
for (i in 1:n.perm) {
  data.perm = sample(CD33_data, n, replace = FALSE)
  T.perm[i] = abs(var(data.perm[gol.fac=="ALL"]))^2/(var(data.perm[gol.fac=="AML"]))^2
}
# p -value
mean (T.perm<T.obs)
```

```
## [1] 0.031
```