

Lab Assignment #3

Math 437 - Modern Data Analysis

Due February 15, 2023

Instructions

There are two purposes to this lab. First, you will get comfortable with the family-wise error rate and false discovery rate. Then, we will get more practice with coding a nonparametric bootstrap estimate of standard error.

```
library(ISLR2)
library(ggplot2)
library(dplyr)
```

This lab assignment is worth a total of **15 points**.

Problem 1: Family-Wise Error Rate

Part a (Code: 1 pt; Computation and Explanation: 0.5 pts)

Run the code in ISLR Labs 13.6.1 and 13.6.2. Put each chunk from the textbook in its own chunk.

```
set.seed(6)
x <- matrix(rnorm(10 * 100), 10, 100)
x[, 1:50] <- x[, 1:50] + 0.5

t.test(x[, 1], mu = 0)

##
## One Sample t-test
##
## data:  x[, 1]
## t = 2.0841, df = 9, p-value = 0.06682
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.05171076  1.26242719
## sample estimates:
## mean of x
## 0.6053582

p.values <- rep(0,100)
for (i in 1:100)
  p.values[i] <- t.test(x[, i], mu = 0)$p.value
decision <- rep("Do not reject H0", 100)
decision[p.values <= 0.05] <- "Reject H0"

table(decision,
c(rep("H0 is False ", 50), rep("H0 is True ", 50)))
```

)

```
##
## decision          H0 is False  H0 is True
##   Do not reject H0           40      47
##   Reject H0                  10      3
```

Power = $1 - 40/50 = 0.20$

Note: Given H_0 false, Power + $\beta = 1$

Power = Reject H_0 & H_0 false Power = $P(\text{reject } H_0 \mid H_0 \text{ is false}) = \#(\text{reject } H_0 \text{ \& } H_0 \text{ is False}) / \# H_0 \text{ false}$

Beta = Do not reject H_0 & H_0 false Power = $1 - \text{Beta}$

```
x <- matrix ( rnorm (10 * 100), 10, 100)
x[, 1:50] <- x[, 1:50] + 1
for (i in 1:100)
  p.values[i] <- t.test (x[, i], mu = 0)$p.value
decision <- rep ("Do not reject H0", 100)
decision[p.values <= .05] <- " Reject H0"
table (decision ,
c( rep ("H0 is False ", 50), rep ("H0 is True ", 50))
)
```

```
##
## decision          H0 is False  H0 is True
##   Reject H0           41      1
##   Do not reject H0     9      49
```

Using the output of Lab 13.6.1, estimate the power of this test to detect each of the two alternative hypotheses $\mu = 0.5$ and $\mu = 1$.

```
#knitr didn't like it when {r Alternative hypothesis $\mu = 0.5$}
x <- matrix ( rnorm (10 * 100), 10, 100)
x[, 1:50] <- x[, 1:50] + 1
for (i in 1:100)
  p.values[i] <- t.test (x[, i], mu = 0.5)$p.value
decision <- rep ("Do not reject H0", 100)
decision[p.values <= .05] <- " Reject H0"
table (decision ,
c( rep ("H0 is False ", 50), rep ("H0 is True ", 50))
)
```

```
##
## decision          H0 is False  H0 is True
##   Reject H0           17      15
##   Do not reject H0     33      35
```

Power = $17/50 = 0.34$

```
#knitr didn't like it when {r Alternative hypothesis $\mu = 1$}
x <- matrix ( rnorm (10 * 100), 10, 100)
x[, 1:50] <- x[, 1:50] + 1
for (i in 1:100)
  p.values[i] <- t.test (x[, i], mu = 1)$p.value
decision <- rep ("Do not reject H0", 100)
decision[p.values <= .05] <- " Reject H0"
table (decision ,
```

```
c( rep ("H0 is False ", 50), rep ("H0 is True ", 50))
)
```

```
##
## decision          H0 is False  H0 is True
##   Reject H0                2        40
##   Do not reject H0         48        10
```

Power = $2/50 = 0.04$

Part b (Explanation: 1 pt)

What does the `p.adjust` function do exactly when `method = bonferroni`? What about when `method = holm`? Why does it make more sense for R to adjust the p-values rather than the significance level when controlling the FWER?

When `method = bonferroni`, the p-values are multiplied by the number of hypotheses, m , in order to obtain the adjusted p-values, but the adjusted p-values aren't allowed to exceed 1. When `method = holm`, the p-values are ordered from least to greatest. Then each p-value is multiplied by $m+1-k$, where k is the rank of the p-value. It makes more sense for R to adjust the p-values rather than the significance level when controlling the FWER because we can simply compare the adjusted p-values to the desired FWER while keeping the power of the test high.

Part c (Explanation: 1 pt)

Consider the second-to-last chunk in ISLR Lab 13.6.2 (the one using `TukeyHSD`). Is it appropriate to do a one-way ANOVA with this data? Explain why or why not. (You may want to produce some graphs and/or numerical summaries to support your answer.)

Hint: What are the assumptions for the ANOVA Test? Are they clearly violated?

I believe it would be appropriate to do an ANOVA test with this data. According to the book, the Tukey method takes as input the output of an ANOVA regression model, which means that the ANOVA method is already in use. There are no ANOVA assumptions being violated, as all the variables are independent from each other and there are no outliers.

```
fund.mini <- Fund[, 1:5]

returns <- as.vector ( as.matrix (fund.mini))
manager <- rep (c("1", "2", "3", "4", "5") , rep (50, 5))
a1 <- aov (returns ~ manager)
TukeyHSD (x = a1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = returns ~ manager)
##
## $manager
##      diff      lwr      upr      p adj
## 2-1 -3.1 -6.9865435 0.7865435 0.1861585
## 3-1 -0.2 -4.0865435 3.6865435 0.9999095
## 4-1 -2.5 -6.3865435 1.3865435 0.3948292
## 5-1 -2.7 -6.5865435 1.1865435 0.3151702
## 3-2  2.9 -0.9865435 6.7865435 0.2452611
## 4-2  0.6 -3.2865435 4.4865435 0.9932010
## 5-2  0.4 -3.4865435 4.2865435 0.9985924
```

```
## 4-3 -2.3 -6.1865435 1.5865435 0.4819994
## 5-3 -2.5 -6.3865435 1.3865435 0.3948292
## 5-4 -0.2 -4.0865435 3.6865435 0.9999095

anova(a1)

## Analysis of Variance Table
##
## Response: returns
##           Df Sum Sq Mean Sq F value Pr(>F)
## manager     4     437   109.25    2.185 0.07126 .
## Residuals 245   12250    50.00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 2: False Discovery Rate

Part a (Code: 0.5 pts)

Run the code in ISLR Lab 13.6.3. Put each chunk from the textbook in its own chunk.

```
library (ISLR2)

fund.pvalues <- rep (0, 2000)
for (i in 1:2000)
  fund.pvalues[i] <- t.test (Fund[, i], mu = 0)$p.value

q.values.BH <- p.adjust (fund.pvalues , method = "BH")
q.values.BH[1:10]

## [1] 0.08988921 0.99149100 0.12211561 0.92342997 0.95603587 0.07513802
## [7] 0.07670150 0.07513802 0.07513802 0.07513802

sum (q.values.BH <= .1)

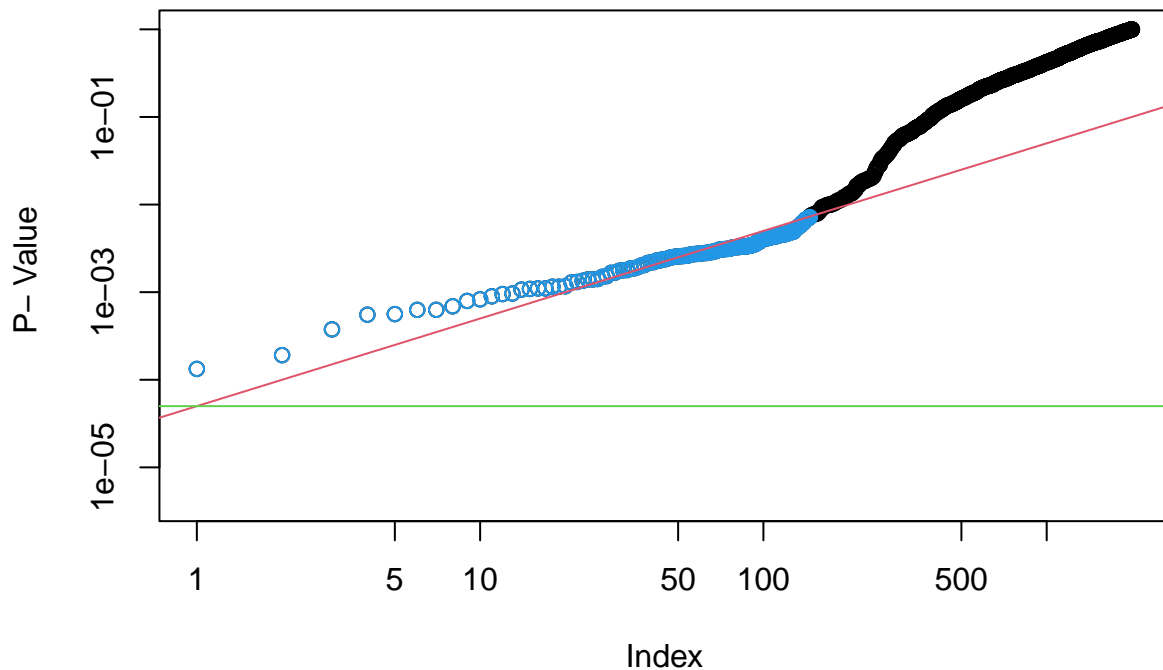
## [1] 146

sum (fund.pvalues <= (0.1 / 2000))

## [1] 0

ps <- sort(fund.pvalues)
m <- length(fund.pvalues)
q <- 0.1
wh.ps <- which(ps < q * (1:m) / m)
if ( length(wh.ps) >0) {
  wh <- 1: max(wh.ps)
} else {
  wh <- numeric(0)
}

plot (ps , log = "xy", ylim = c(4e-6, 1), ylab = "P- Value ",
      xlab = " Index ", main = "")
points (wh, ps[wh], col = 4)
abline (a = 0, b = (q / m), col = 2, ltyf = TRUE)
abline (h = 0.1 / 2000, col = 3)
```



Part b (Code: 1 pt)

Finish writing the `FDR_plot` function in the code chunk below. This function should take two arguments: `p.values`, a vector of p-values, and `q`, the desired False Discovery Rate, and produce a graph like those in Figure 13.6. (Assume that on the graphs, q and α are set to the same value.)

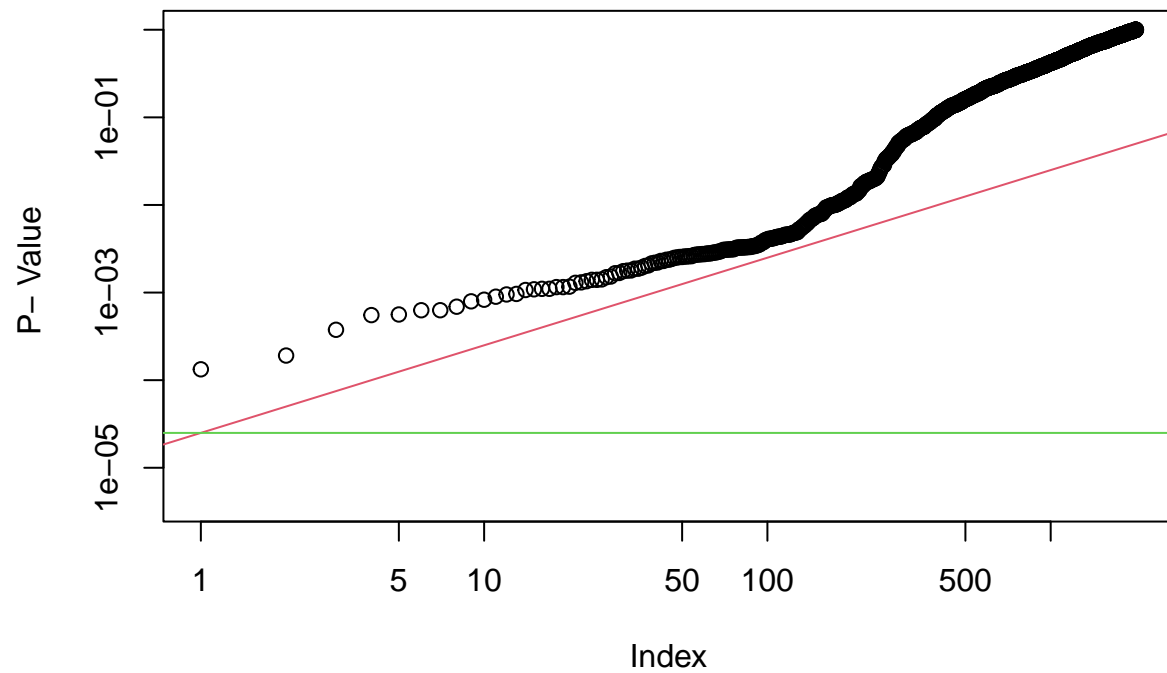
```
FDR_plot <- function(p.values, q){

  # Copy code from the last two chunks of ISLR Lab 13.6.3, but replace variable names and hard-coded va
  ps <- sort(p.values)
  m <- length(p.values)
  #q <- 0.1
  #we don't want to assign q in the function because q is an argument
  wh.ps <- which(ps < q * (1:m) / m)
  if ( length(wh.ps) >0) {
    wh <- 1: max(wh.ps)
  } else {
    wh <- numeric(0)
  }
  plot(ps, log = "xy", ylim = c(4e-6, 1), ylab = "P- Value ", xlab = " Index ", main = "")
  points (wh, ps[wh], col = 4)
  abline (a = 0, b = (q / m), col = 2, untf = TRUE)
  abline (h = q/m, col = 3)

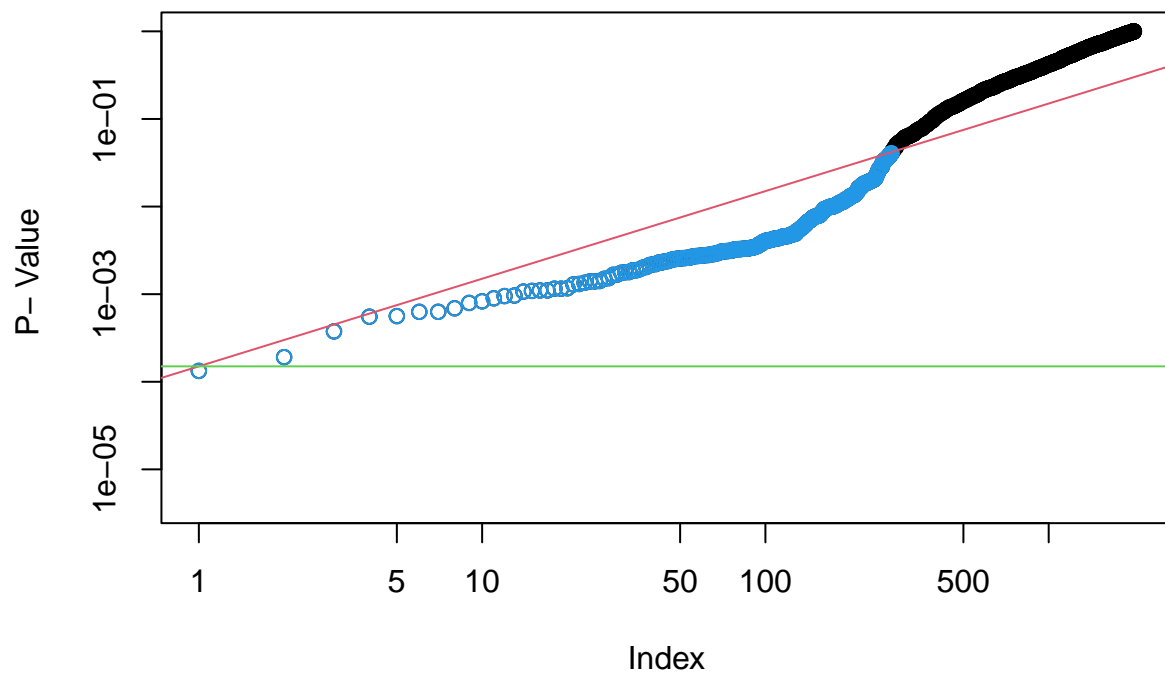
  invisible(wh.ps) # invisibly return the significant p-values
}
```

Test your function. First, attempt to duplicate the left and right panels of Figure 13.6.

```
FDR_plot(fund.pvalues, q=0.05)
```

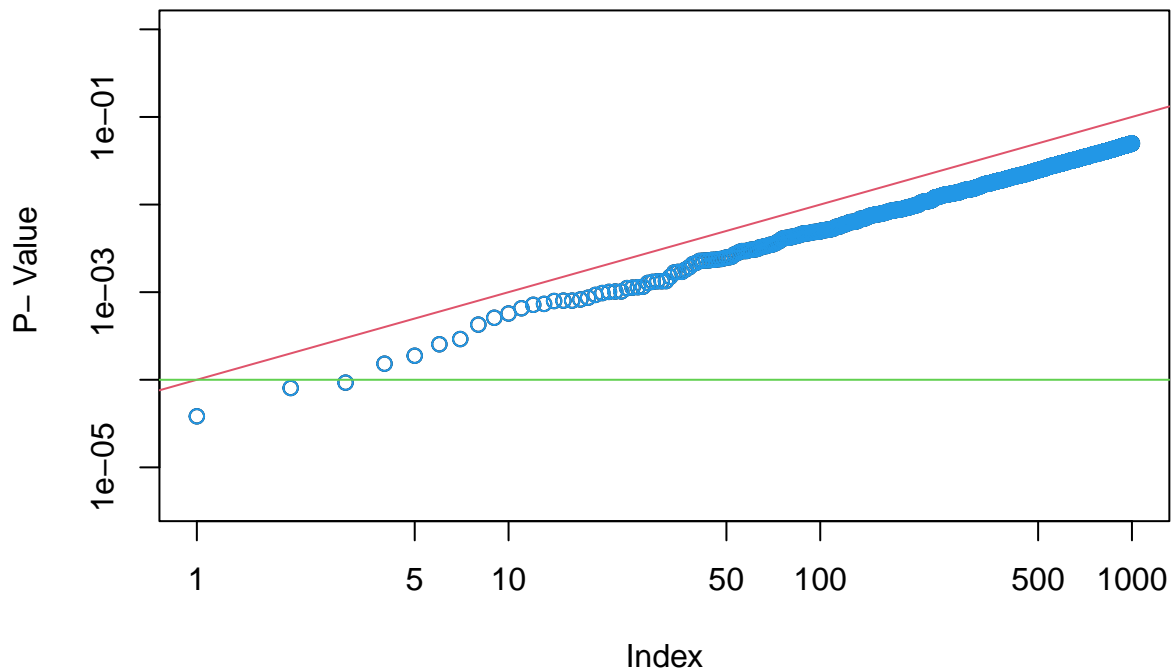


```
FDR_plot(fund.pvalues, q=0.3)
```



Then, test your function on the simulated dataset in the chunk below, using $q = 0.1$.

```
set.seed(12)
sim_pvalues <- runif(1000, min = 1e-5, max = 0.05001)
# Now put a line of code running your function on this set of "significant" p-values
FDR_plot(sim_pvalues, q=0.1)
```



#every point should be blue

Problem 3: Simulation Study of FWER and FDR

This problem is adapted from ISLR Chapter 13, Exercise 8.

Part a (Code: 1 pt)

Using the code in Exercise 13.7.8, create a 20 x 100 matrix where each column represents 20 random numbers from $N(0, 1)$.

```
set.seed(1)
n = 20
m = 100
X = matrix(rnorm(n * m), ncol = m)
```

Then, run a t-test on each column of the matrix testing $H_0 : \mu = 0$ against $H_a : \mu \neq 0$. We are going to use the `apply` function to do this rather than adapt the `for` loop from the ISLR Labs. (Recall from the class activities that the `apply` function applies a single function to each row (`MARGIN = 1`) or column (`MARGIN = 2`) of a matrix.)

Don't forget to delete the `eval = FALSE` after you've fixed this code chunk to run properly!

```
t_test_0 <- function(x){
  # x: a vector of data
  # t.test(x, alternative="two.sided", mu=0)
  p_value <- t.test(x, alternative="two.sided", mu=0)$p.value # write a line of code that extracts the p-value
```



```

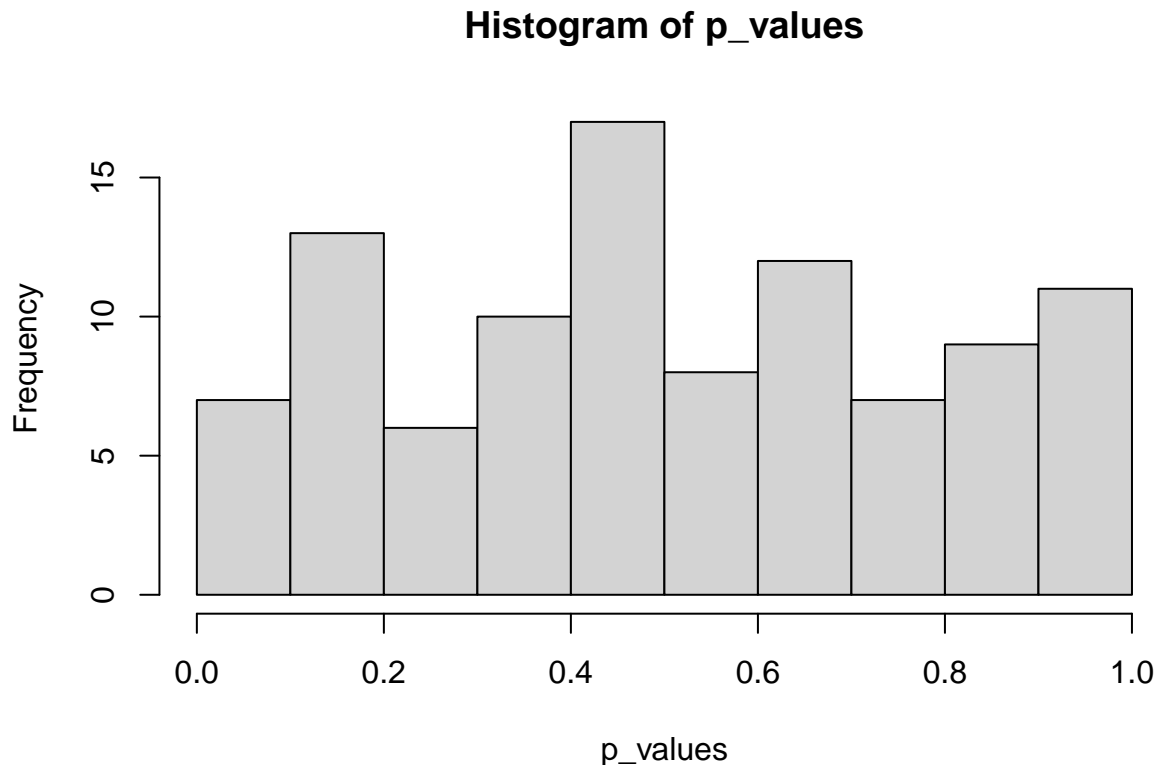
    return(p_value)
}

p_values <- apply(X, 2, t_test_0)

```

Plot a histogram of the p-values obtained.

```
hist(p_values)
```



Part b (Code: 1 pt, Explanation: 0.5 pts)

Without any adjustment for multiple hypothesis tests, how many null hypotheses would be rejected at $\alpha = 0.05$? We can take advantage of the fact that R implicitly converts *logical* (TRUE/FALSE) variables to *numerical* variables.

4 hypotheses are rejected at $\alpha = 0.05$.

```

alpha <- 0.05
sum(p_values <= alpha)

```

```
## [1] 4
```

Obtain the adjusted p-values using the Holm step-down procedure. How many null hypotheses would be rejected if we control the FWER at 0.05?

0 null hypotheses were rejected using the Holm step-down procedure.

```
#similarly to the previous chunk, we can do sum(p.adjust(...) <= alpha) to get the number of null hypotheses  
sum(p.adjust(p_values, method="holm") <= 0.05)
```

```
## [1] 0
```

Obtain the adjusted p-values using the Benjamini-Hochberg procedure. How many null hypotheses would be rejected if we control the FDR at 0.05?

0 null hypotheses were rejected using the Benjamini-Hochberg method.

```
sum(p.adjust(p_values, method="BH") <= 0.05)
```

```
## [1] 0
```

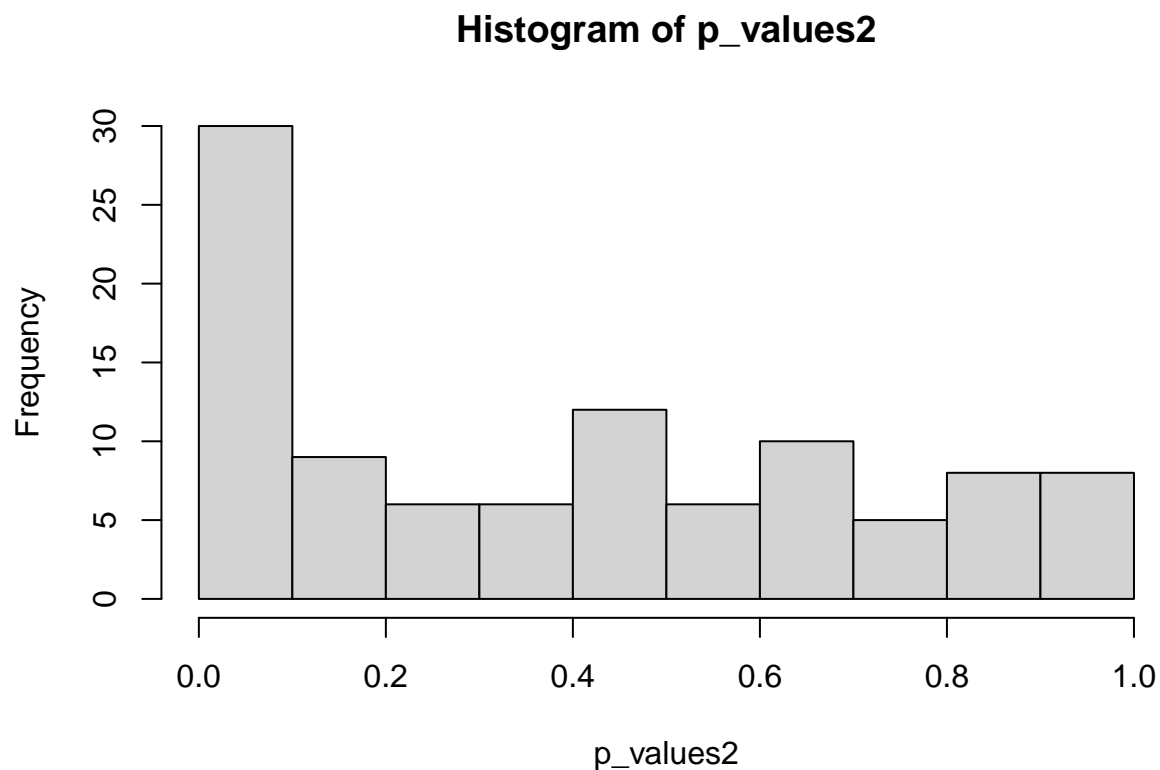
Part c (Code: 1 pt)

Create a new matrix, X2, that is exactly the same as X, except that the first 25 fund managers do actually have a slight long-term return of +1% (i.e., in rows 1-25 $\mu = 1$ not 0.01). Conduct your 100 t-tests and plot a histogram of the new p-values.

```
X2 <- X  
X2[,1:25] <- matrix(rnorm(n * 25, mean = 1), ncol = 25)
```

```
# don't rewrite the function, just apply it to the new dataset!  
p_values2 <- apply(X2, 2, t_test_0)
```

```
hist(p_values2)
```



Part d (Code: 1 pt, Explanation: 1 pt)

Without any adjustment for multiple hypothesis tests, how many of the 75 true null hypotheses would be rejected at $\alpha = 0.05$? How many of the 25 false null hypotheses would be rejected?

Three (3) of the 75 true null hypotheses would be rejected at $\alpha = 0.05$. 1 of the 25 false null hypotheses was rejected.

```
reject.falseH0 <- sum(p_values[1:25] <= alpha)
reject.trueH0 <- sum(p_values[26:100] <= alpha)
c(true = reject.trueH0, false = reject.falseH0)
```

```
## true false
##      3      1
```

Obtain the adjusted p-values using the Holm step-down procedure. How many null hypotheses of each type would be rejected if we control the FWER at 0.05?

17 null hypotheses were rejected.

```
sum(p.adjust(p_values2, method="holm") <= 0.05)
```

```
## [1] 14
```

Obtain the adjusted p-values using the Benjamini-Hochberg procedure. How many null hypotheses of each type would be rejected if we control the FDR at 0.05?

26 null hypotheses were rejected.

```
sum(p.adjust(p_values2, method="BH") <= 0.05)
```

```
## [1] 24
```

Compare your results from the three procedures. Why is it important to consider the situation where *some* null hypotheses are true, when evaluating the performance of the different procedures?

If we only considered situations where the null hypotheses were false, then we might only use procedures that work well when they are false but not when they are true.