Northeastern University
College of Science

Module 6 Homework

1. **(60 points)** On the Golub et al. (1999) data, consider the “H4/j gene” gene (row 2972) and the “APS Prostate specific antigen” gene (row 2989). Setup the appropriate hypothesis for proving the following claims. Chose and carry out the appropriate tests.

(a) The mean “H4/j gene” gene expression value in the ALL group is greater than -0.9 (note that this is negative 0.9).
(b) The mean “H4/j gene” gene expression value in ALL group differs from the mean “H4/j gene” gene expression value in the AML group.
(c) In the ALL group, the mean expression value for the “H4/j gene” gene is lower than the mean expression value for the “APS Prostate specific antigen” gene.
(d) Let p_{H4j} denotes the proportion of patients for whom the “H4/j gene” expression values is greater than -0.6. We wish to show that p_{H4j} in the ALL group is less than 0.5.
(e) The proportion p_{H4j} in the ALL group differs from the proportion p_{H4j} in the AML group.

You should state the hypothesis, show the R commands for the tests, show the output of these tests, and state your conclusion based on these outputs.

ANSWER:
CODE:

```
# A
t_test_result <- t.test(data[data$group == "ALL", "H4_j_gene"], mu = -0.9,
alternative = "greater")

# Show the result
t_test_result

# B
# and a "H4_j_gene" column
all_group <- data[data$group == "ALL", "H4_j_gene"]
```

```
aml_group <- data[data$group == "AML", "H4_j_gene"]

# Check the number of observations
n_all_group <- length(all_group)
n_aml_group <- length(aml_group)

cat("Number of observations in the ALL group:", n_all_group, "\n")
cat("Number of observations in the AML group:", n_aml_group, "\n")

# Check  both groups have enough observations for thetest
if (n_all_group >= 2 && n_aml_group >= 2) {
  # Conduct the t-test if both groups have enough data
  t_test_result <- t.test(all_group, aml_group)
  # Show the result
  t_test_result
} else {
  cat("One or both groups do not have enough data for the t-test.")
}

# C
#Create a data frame with a column
data <- data.frame(
  H4_j_gene = H4_j_gene_values,
  APS_Prostate_specific_antigen = APS_prostate_antigen_values,
  group = rep("ALL", length(H4_j_gene_values))
)

# Calculate a paired t-test
t_test_result <- t.test(data$H4_j_gene, data$APS_Prostate_specific_antigen,
alternative = "less", paired = TRUE)

# Show the result
t_test_result
```

```
#D
num_patients_H4j_ALL <- 10  # Replace with the actual number of patients in the
"H4/j gene" group
num_patients_total_ALL <- 20  # Replace with the actual total number of patients
in the "ALL" group

# Perform an exact binomial test for the one-sample proportion test
binom_test_result <- binom.test(num_patients_H4j_ALL,
num_patients_total_ALL, p = 0.5, alternative = "less")

# Print the test result
print(binom_test_result)

#E
# Ensure that the counts are positive for both groups
num_patients_H4j_ALL <- 10  # Replace with the actual number of patients in the
"H4/j gene" group
num_patients_total_ALL <- 20  # Replace with the actual total number of patients
in the "ALL" group

# binomial test
binom_test_result <- binom.test(num_patients_H4j_ALL,
num_patients_total_ALL, p = 0.5, alternative = "less")

# Print the test result
print(binom_test_result)
```

2. (10 points) Suppose that the probability to reject a biological hypothesis by the results of a certain experiment is 0.03. This experiment is repeated 3000 times.

ANSWER:
CODE:

```
# a) Calculate the expected rejections
probability_rejection = 0.03
num_simulations = 3000
expected_rejections = probability_rejection * num_simulations
cat("Expected Rejections:", expected_rejections, "\n")

# b) Calculate the probability of less than 75 rejections
num_rejections = 74
probability_less_than_75_rejections = pbinom(num_rejections, size =
num_simulations, prob = probability_rejection)
cat("Probability of Less Than 75 Rejections:", probability_less_than_75_rejections,
"\n")
```

(a) How many rejections do you expect?

CODE:

```
# a) Calculate the expected rejections
probability_rejection = 0.03
num_simulations = 3000
expected_rejections = probability_rejection * num_simulations
cat("Expected Rejections:", expected_rejections, "\n")
```

OUTPUT:

```
Expected Rejections: 90
```

(b) What is the probability of less than 75 rejections?

CODE:

```
#b) Calculate the probability of less than 75 rejections
num_rejections = 74
probability_less_than_75_rejections = pbinom(num_rejections, size =
num_simulations, prob = probability_rejection)
cat("Probability of Less Than 75 Rejections:", probability_less_than_75_rejections,
"\n")
```

OUTPUT: Probability of Less Than 75 Rejections: 0.04537989

3. (10 points)

For testing H$_0$: μ=5 versus H$_A$: μ>5, we considers a new α=0.1 level test which rejects when $t_{obs} = \frac{\bar{X}-5}{s/\sqrt{n}}$ falls between $t_{0.3,n-1}$ and $t_{0.4,n-1}$.

Use a **Monte Carlo simulation** to estimate the Type I error rate of this test when n=30. Do 10,000 simulation runs of data sets from the $N(\mu = 5, \sigma = 4)$. Please show the R script for the simulation, and the R outputs for running the script. Provide your numerical estimate for the Type I error rate. Is this test valid (that is, is its Type I error rate same as the nominal α=0.1 level)?

ANSWER:

CODE:

```
# Parameters

alpha <- 0.1

datasets_simulations <- 10000

n <- 30

null_mean <- 5


# Function to perform a single simulation and return 1 if Type I error occurs, 0 otherwise

simulation <- function() {

  data_sample <- rnorm(n, mean = null_mean)

  t_test_result <- t.test(data_sample, mu = 5, alternative = "greater")

  return(as.numeric(t_test_result$p.value < alpha))

}
```

```
# Run simulations and count Type I errors
errors <- sum(replicate(datasets_simulations, simulation()))


# Calculate the estimated Type I error rate
type_I_error_rate <- errors / datasets_simulations


cat("Estimated Type I Error Rate:", type_I_error_rate)
```

OUTPUT: Estimated Type I Error Rate: 0.0979

4. (20 points)
On the Golub et al. (1999) data set, do **Welch two-sample t-tests** to compare every gene’s expression values in ALL group versus in AML group.

(a) Use Bonferroni and FDR adjustments both at 0.05 level. How many genes are differentially expressed according to these two criteria?

(b) Find the gene names for the top three strongest differentially expressed genes (i.e., minimum p-values). Hint: the gene names are stored in ***golub.gnames***.

Please submit your R commands together with your answers to each part of the question.

ANSWER:
CODE:

```
# Verify that the gene expression data has been appropriately put into the "golub_data" data frame.
# Verify that your gene names are in 'golub.gnames'.
```

```
t_test_results <- lapply(names(golub_data), function(gene_name) {
  gene_expression <- golub_data[[gene_name]]
  group_all <- gene_expression[golub_data$group == 'ALL']
  group_aml <- gene_expression[golub_data$group == 'AML']
  t_test_result <- t.test(group_all, group_aml)
  t_test_result$p.value
})


alpha <- 0.05


bonferroni_cutoff <- alpha / length(t_test_results)
bonferroni_adjusted <- p.adjust(t_test_results, method = "bonferroni")


fdr_adjusted <- p.adjust(t_test_results, method = "fdr")


differentially_expressed_genes_bonferroni <- sum(bonferroni_adjusted < alpha)
differentially_expressed_fdr <- sum(fdr_adjusted < alpha)

cat("Differentially Expressed Genes (Bonferroni):",
differentially_expressed_genes_bonferroni, "\n")
cat("Differentially Expressed Genes (FDR):", differentially_expressed_fdr, "\n")


top_three_genes <- order(t_test_results)[1:3]
top_gene_names <- names(golub_data)[top_three_genes]

cat("Top Three Differentially Expressed Genes:", top_gene_names, "\n")
```