Project 1

Monikrishna Roy

2021-10-17

Question 1

Download the dataset the **GSE9782.RData** from web campus. This dataset measures the change in gene expression of myeloma patients. The dataset file contains a gene expression matrix and a group (treated versus untreated). Load the data into R environment;

```
# Loading data using function `load()`
load("GSE9782.RData")

treated <- which(group == "treated")
control <- which(group != "treated")
```

Question 2

Calculate log base 2 of the expression; Compute the difference in mean log base 2 expression between the two groups (named logFC).

```
# log base 2 of the expression
log2Data <- log(x = dataGSE9782, base = 2)

# a function to calculate the difference in mean
cal_mean_diff <- function(x, treated, control) {
  mean(x[treated]) - mean(x[control])
}
# used apply to call the function
logFC <- apply(log2Data, 1, cal_mean_diff, treated, control)
```

Question 3

Perform a t-test to compare the control and treated groups; output a data frame that contains the following columns: gene ids (row names of expression data matrix), p-value, t-score, logFC;

```
# Function to calculate p-value
cal_p_value <- function(x, treated, control) {
  t.test(x[treated], x[control])$p.value
}
# Function to calculate t-score
cal_t_score <- function(x, treated, control) {
  t.test(x[treated], x[control])$statistic
}

PValue <- apply(log2Data, MARGIN = 1, FUN = cal_p_value, treated, control)
TScore <- apply(log2Data, MARGIN = 1, FUN = cal_t_score, treated, control)
```

```r
# rownames used as gene ids
geneIds <- rownames(log2Data)

df <- data.frame(
  row.names = NULL,
  "GeneID" = geneIds,
  "PValue" = PValue,
  "TScore" = TScore,
  "LogFC" = logFC
)

# printing the head of data frame
head(df)
```

```
##      GeneID     PValue     TScore       LogFC
## 1         1 0.14066910  1.4826831  0.22074744
## 2        10 0.08589248 -1.7291716 -0.15378495
## 3       100 0.22755884  1.2110105  0.10761972
## 4      1000 0.63568245 -0.4747268 -0.06559339
## 5     10000 0.03276403 -2.1565886 -0.13981665
## 6 100009676 0.53913863  0.6155821  0.04901606
```

Question 4

Plot the genes with an absolute log fold change > 1 and p-value < 0.05 in red and the remaining ones in black in a volcano plot;

```r
plot(
  x = df$LogFC,
  y = -log10(df$PValue),
  xlab = 'logFC',
  ylab = '-log10(p-value)',
  main = "Volcano plot",
  col = ifelse(abs(df$LogFC) > 1 & df$PValue < 0.05, 'red', 'black'),
  xlim = c(-2, 2)
)
abline(h = -log10(0.05), col = "red")
abline(v = -1, col = "blue")
abline(v = 1, col = "blue ")
```

Question 5

Perform the analysis at point (3) above using a Wilcoxon test and find the genes that are significant at 5% based on both types of tests (t-test and Wilcoxon);

```r
# Function to calculate p-value with Wilcoxon test
cal_p_value_wilcox <- function(x, treated, control) {
  wilcox.test(x[treated], x[control])$p.value
}

# Function to calculate t-score using Wilcoxon test
cal_t_score_wilcox <- function(x, treated, control) {
  wilcox.test(x[treated], x[control])$statistic
}
```

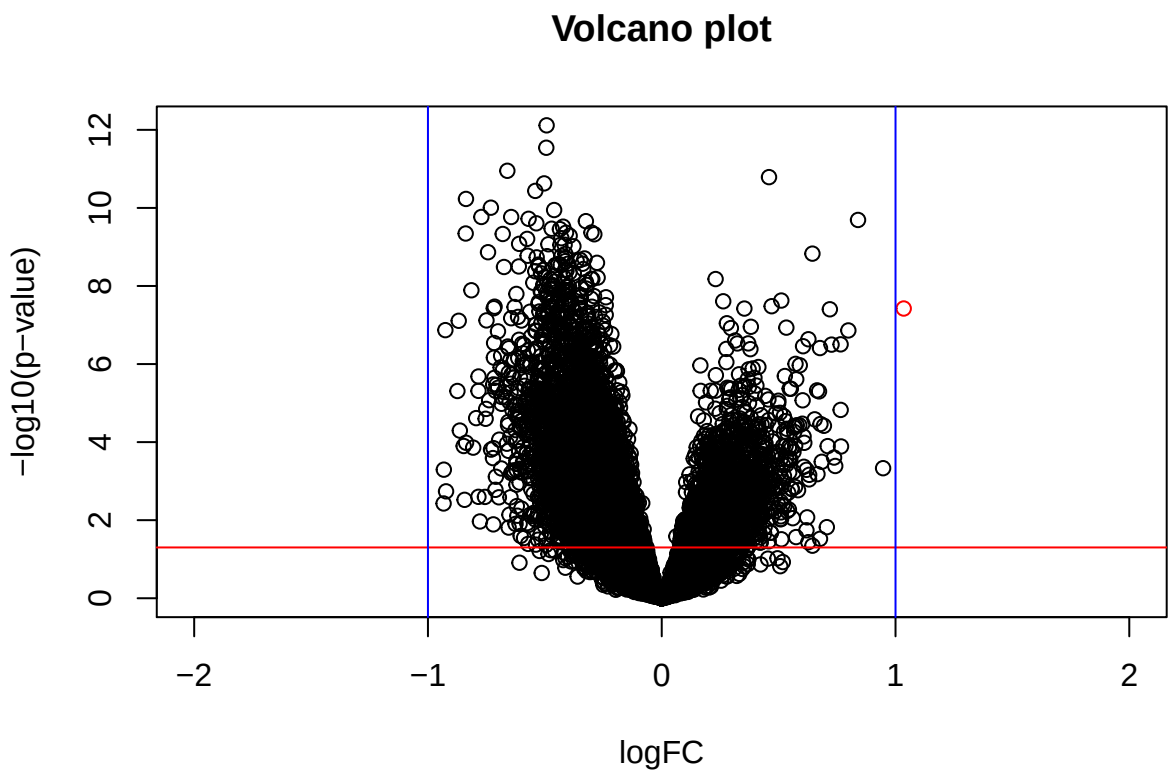

Figure 1: Volcano plot of the genes with an absolute log fold change > 1 and p-value < 0.05 in red and the remaining ones in black

```
# used apply to call the function
PValueWx <-
  apply(log2Data, MARGIN = 1, FUN = cal_p_value_wilcox, treated, control)
TScoreWx <-
  apply(log2Data, MARGIN = 1, FUN = cal_t_score_wilcox, treated, control)

# Finding the genes that are significant at 5% based on both types of tests (t-test and Wilcoxon)
dfPvalue <-
  data.frame(row.names = geneIds,
             "PValueWx" = PValueWx,
             "PValueTs" = PValue)

significantGenes <-
  row.names(dfPvalue[which(dfPvalue$PValueWx < 0.05 &
                             dfPvalue$PValueTs < 0.05),])

# Printing the length of significant genes
length(significantGenes)
```

```
## [1] 6806
```

```
# Printing the first 10 significant genes
print(significantGenes[1:10])
```

```
##  [1] "10000"     "10002"     "10005"     "10007"     "1001"      "10010"
##  [7] "100125288" "100126784" "100126791" "100127886"
```

Question 6

Plot the distribution function of log base 2 expression levels for the data provided;

```
hist(log2Data, probability = T)
```

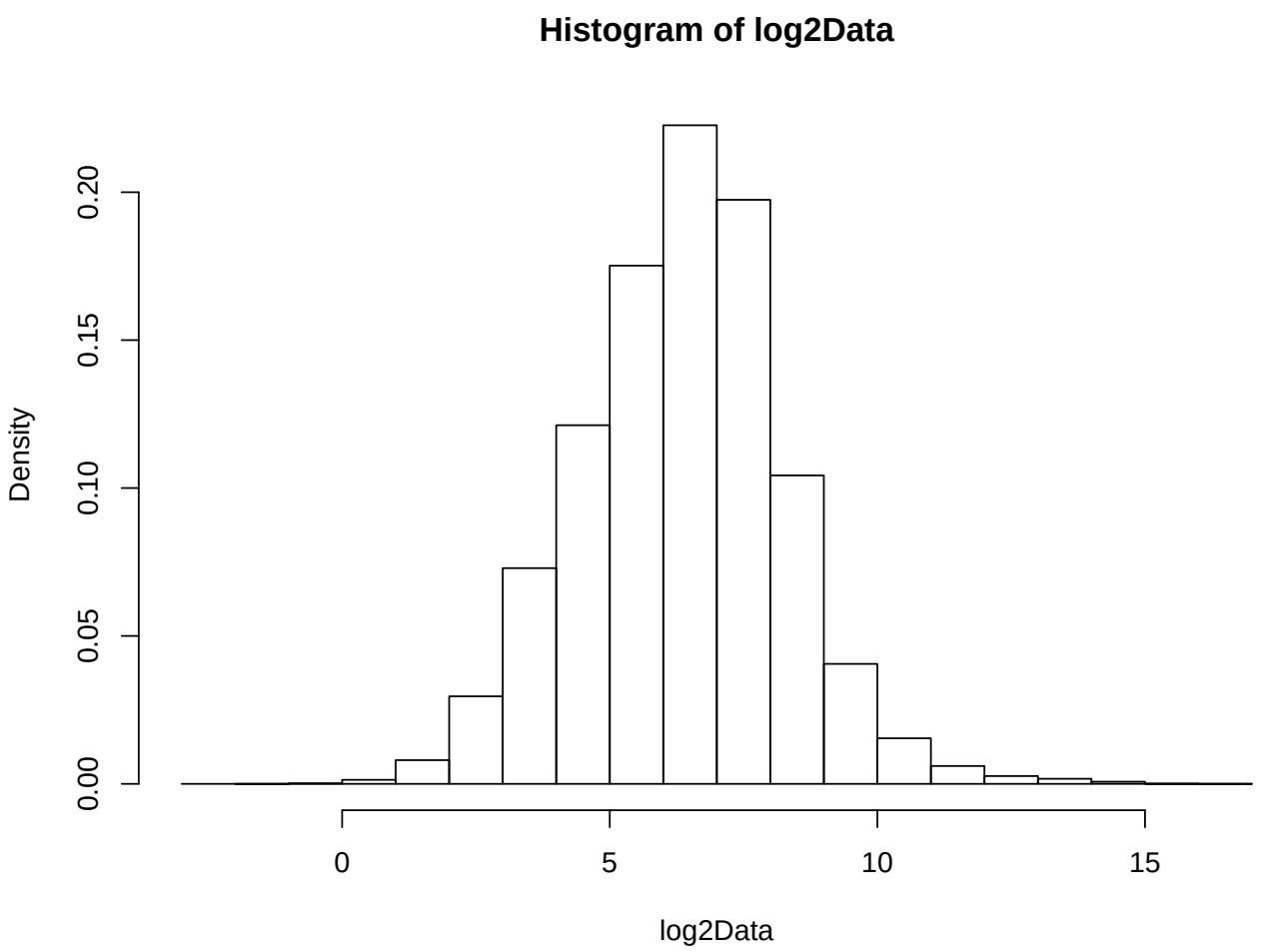

Figure 2: The distribution function of log base 2 expression levels for the data

Question 7

Plot the values of the gene with the maximum absolute expression change using box plots to compare the gene's values in control versus condition.

```
maxGene_df <- data.frame(group = group,
                         value = log2Data[which(abs(logFC) == max(abs(logFC))),])

maxGene_df %>% ggplot(aes(x = group, y = value)) + geom_boxplot() + theme_classic()
```

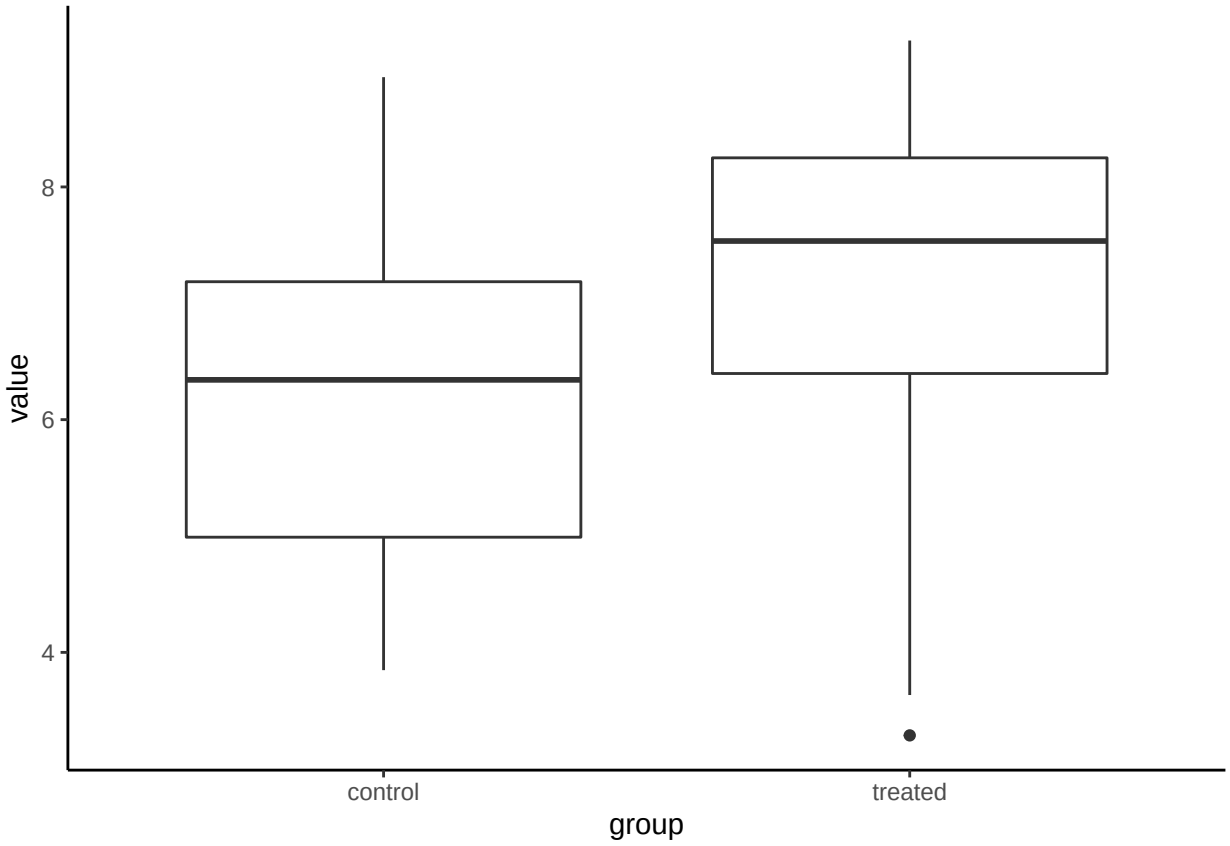

Figure 3: Box plots to compare the gene's values in control versus treated