Homework01

Tanubrata Dey

9/23/2021

Homework 1

DUE: 09/23/21

Please submit your R code and functions in one file.

Remember to comment on your code!

We will use the data file **expvalues.txt**.

Remember, the first line defines the different experiments and after that every row is a different gene and its expression (RNA transcript abundance) values. The first column is the gene name, followed by the different experiments. First three experiments are replicates of the **control** experiments and the last three are replicates for the **treatments**.

1. Descriptive Statistics

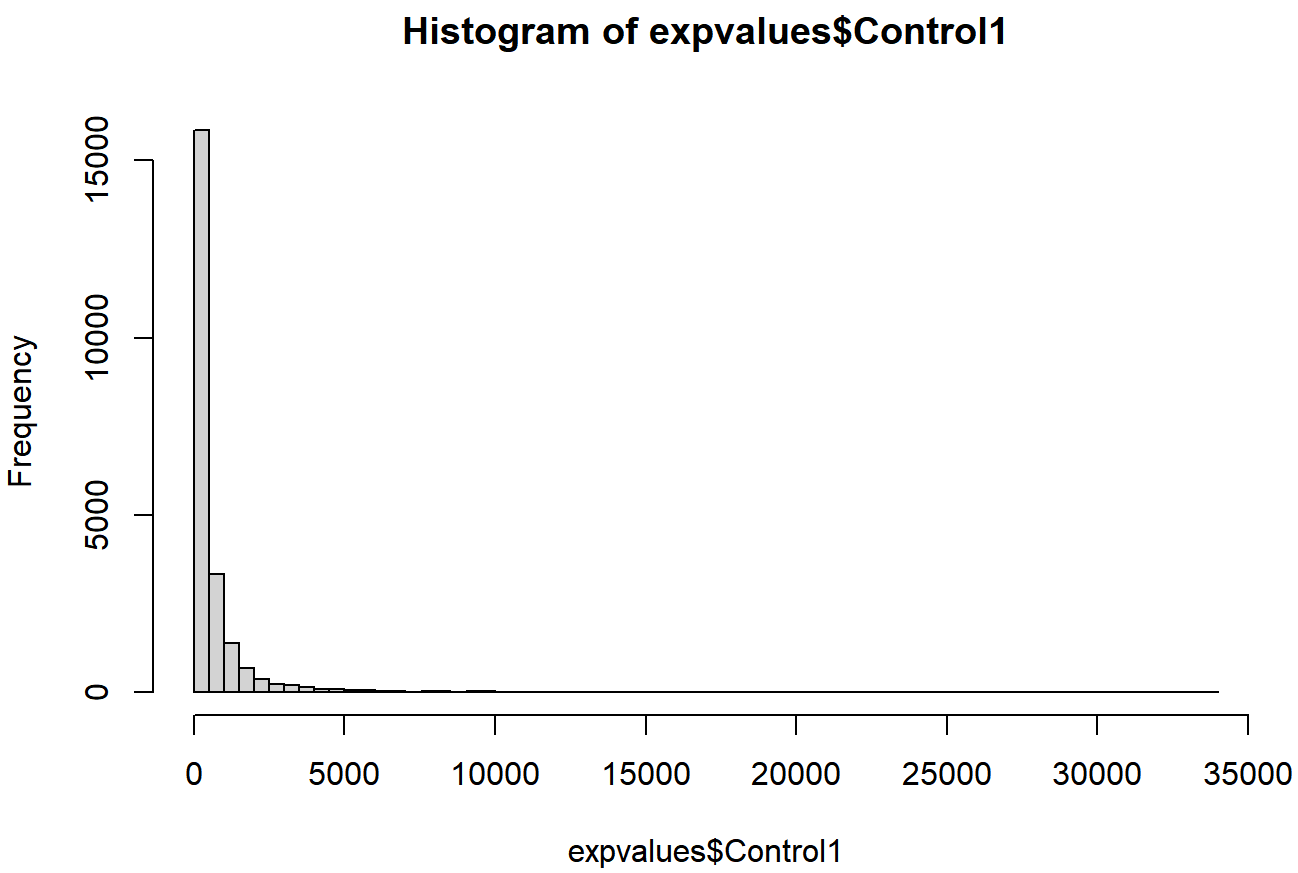
a. Load the file expvalues.txt into R and call it **expvalues**.

```
# Answer here
expvalues <- read.table("expvalues.txt") #reading the file expvalues.txt in expvalues
variable
```

b. Create a histogram of all values in the columns **Control1**. Use the argument **breaks=100** in the argument. What can you conclude about the data?

```
# Answer here

hist(expvalues$Control1, breaks = 100) #histogram of first column in expvalues variab
le
```

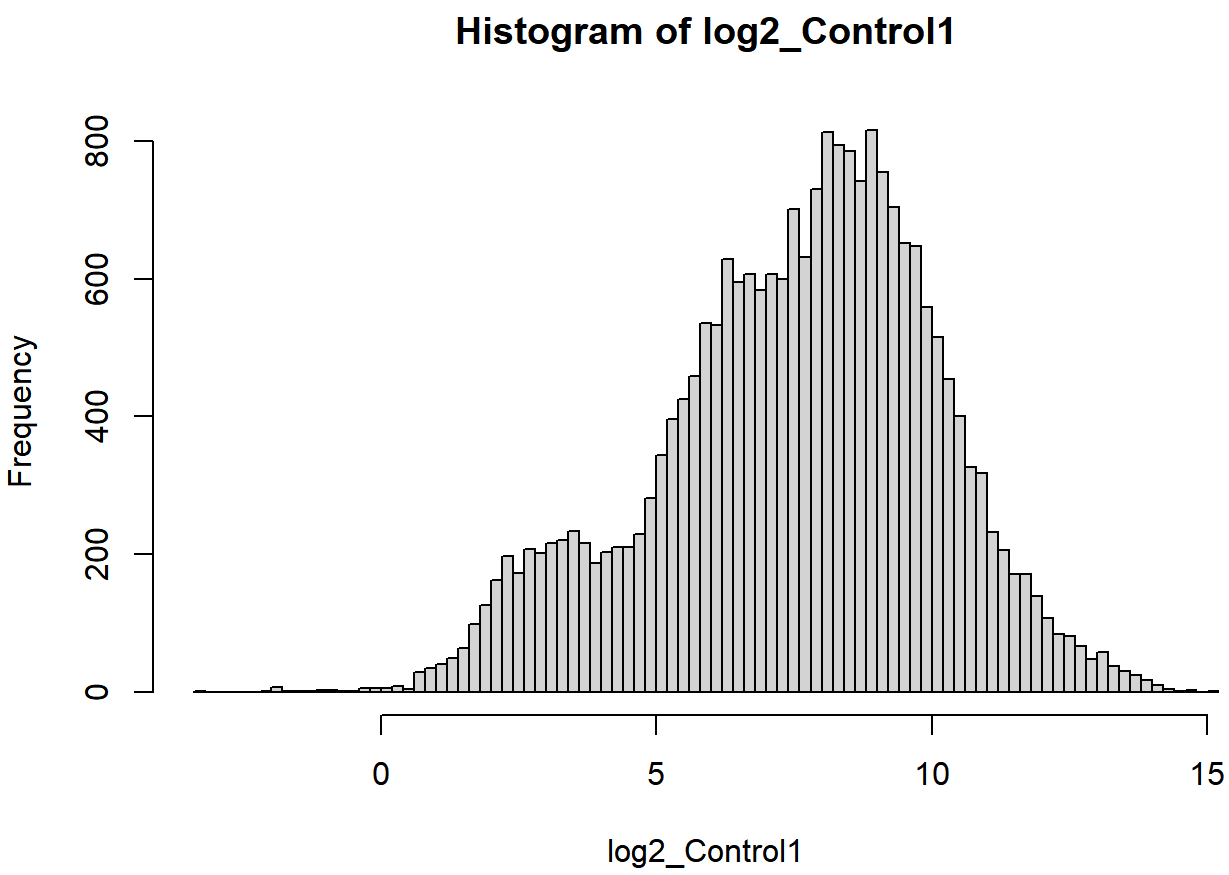


What can you conclude about the data?

The graph above shows that the data are not well distributed and are biased towards 0, making it hard to visualize the expression data.

c. Values can be transformed for many reasons. Perform a log transformation by simply taking the **log2()** of the values in the column **Control1** and then create a histogram plot again. What can you conclude about this graph?

```
# Answer here
log2_Control1 <- log2(expvalues$Control1) #taking log2() of the first column of expva
lues
hist(log2_Control1, breaks = 100) #plotting histogram of log2 values
```

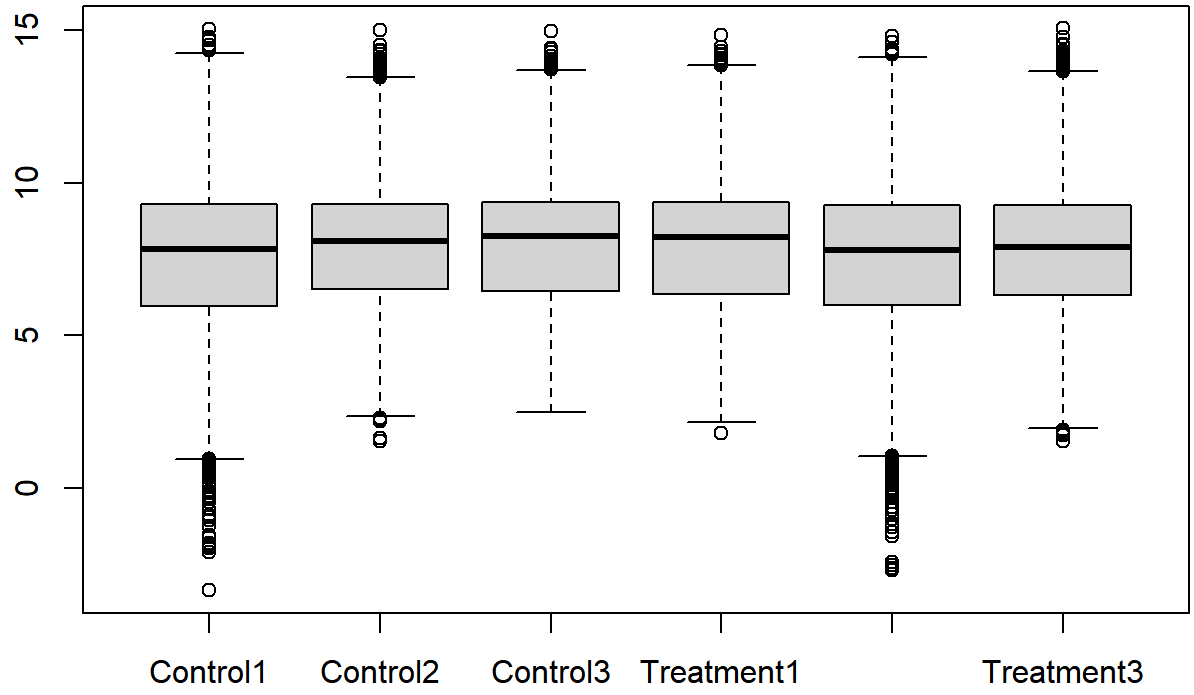


What can you conclude about this graph?

The expression values seem to be better distributed in the log2() graph because we can see in the above graph that theexpression levels are between 0 to 15 and are not biased as that one in previous graph where the curve was in 0. The expression values are well differentiated here giving us a better idea about the expression data.

d. Now that we are convinced that taking the log2 of the values gives us a better representation of the distribution of the values, create a boxplot using the log2 of all the values. Which two samples look different from the rest? Explain why.

```
# Answer here
log2_expvalues <- log2(expvalues) #taking log2 of the whole data in variable expvalue
s
boxplot(log2_expvalues) #making boxplot of the log2() transformed data
```



Which two samples look different from the rest? Explain why.

Control1 and Treatment 2 looks different than the rest because these 2 columns shows values below 0, this can be confirmed by seeing the whiskers in the boxplot that there are many values in both control1 and treatment2 that are at 0 or below 0, seems to be outliers.

2. Fold change calculation:

a. The first three columns are biological replicates of **control** samples and last three columns are **treatment** samples. Create a factor, called **expgroups** to represent this information.

```
# Answer here
expgroups <- factor(c("Control", "Control", "Control", "Treatment", "Treatment", "Tre
atment")) #creating a factor of the columns so that the data is divided into 2 levels
```

b. Using the original expvalues (not the logged values), calculate the average **treatment** and **control** for each gene. Save the results in a matrix called **expmeans**. You can use loops or apply functions. How many rows and columns are there in expmeans?

```
# Answer here
tapply(as.numeric(expvalues[1,]), expgroups, mean) #grouping and taking mean of first
row
```

```
##   Control Treatment
##  254.0256  195.5529
```

```
expmeans <- t(apply(expvalues, 1, tapply, expgroups, mean)) #creating a variable expm
eans that contains the average of control and treatment as a whole

nrow(expmeans) #printing number of rows
```

```
## [1] 22810
```

```
ncol(expmeans) #printing number of columns
```

```
## [1] 2
```

```
is.matrix(expmeans) #checking if expmeans is a matrix
```

```
## [1] TRUE
```

How many rows and columns are there in expmeans?

There are 22810 rows and 2 columns in expmeans.

c. For each gene we want to determine how much the expression has changed in the treatment relative to the expression in control. Calculate the ratio of using the treatment and control values in expmeans. Save it in a vector call expratio.

```
# Answer here
#expmeans <- as.data.frame(expmeans)
#expratio <- (expmeans$Treatment/expmeans$Control)
```

```
# Answer here
expratio = as.vector(expmeans[,2]/expmeans[,1]) #creating a vector expratio that cont
ains the ratio of treatment and control

is.vector(expratio) #checking if expratio is a vector
```

```
## [1] TRUE
```

d. Take the log2 of expratio and call it explog2ratio. You have just calculated log fold change.

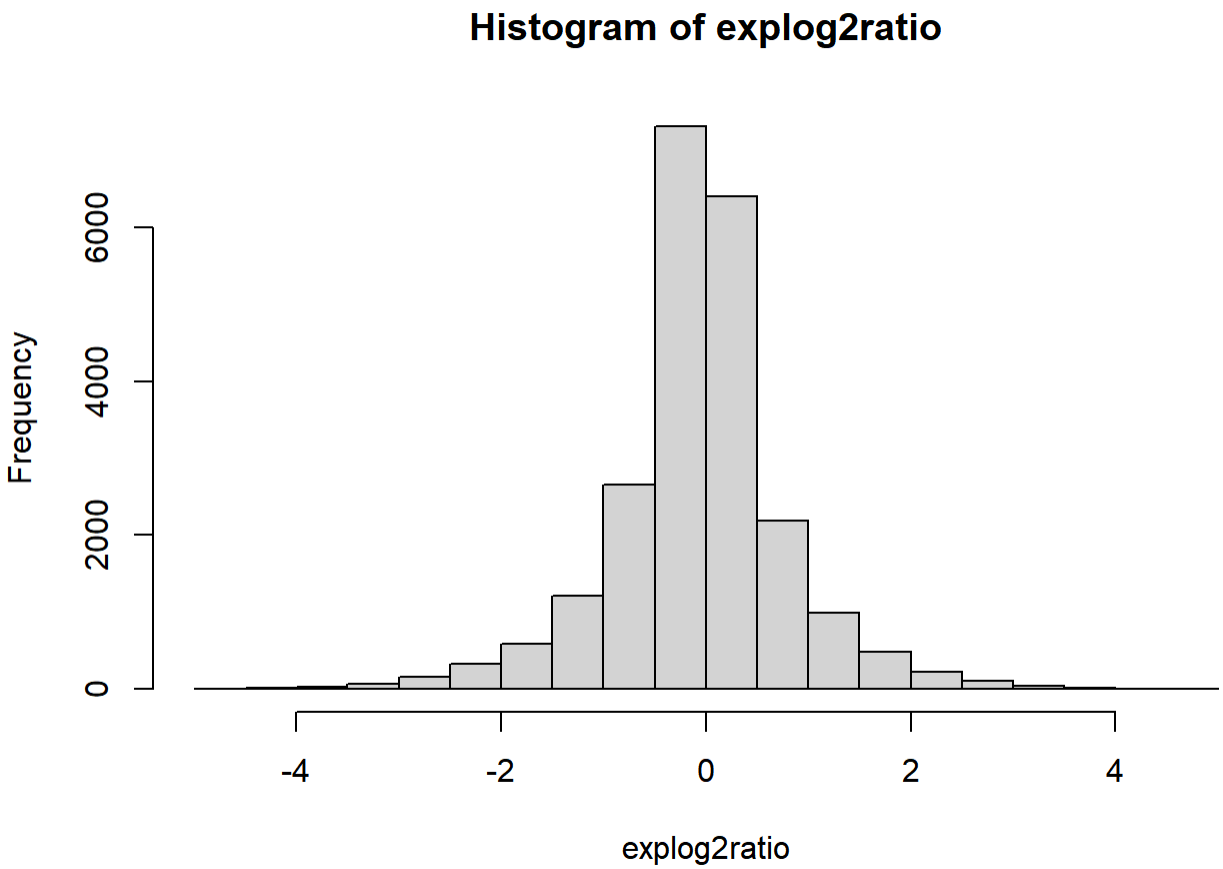
```
# Answer here

explog2ratio <- log2(expratio) #creating a variable explog2ratio that save log2() fol
d change data of the variable expratio
```

e. By taking the log2() of the expratio you have made the values much more symmetric. Positive values mean that original ratio was greater than one (treatment is higher) and a negative value means the ratio was less than one (control was higher). Create a histogram of expratio to confirm.

```
# Answer here

hist(explog2ratio) #creating a histogram of explog2ratio variable that contains fold-change for each gene
```



f. Since our values are in log2 scale, a value of 1 means the change was 2x. So a value of 1 means the value doubled, and value of -1 means the value became half. How many genes have a log2 fold change > 1 OR < -1 ?

```
# Answer here

sum(explog2ratio > 1 | explog2ratio < -1) #getting number of genes that have a log2 fold change > 1 OR < -1.
```

```
## [1] 4244
```

How many genes have a log2 fold change > 1 OR < -1 ?

There are 4244 genes that have a log2 fold change > 1 OR < -1.

g. Identify the gene (get the name of the gene) that has the highest explog2ratio and save the name as upGene. A high log fold change means that the expression increased in treatment compared to control.

```
# Answer here

explog2ratio_df <- as.data.frame(explog2ratio) #converting explog2ratio data into df
and saving it in explog2ratio_df

explog2ratio_df$gene_names <- rownames(expmeans) #adding genenames as a column in exp
log2ratio_df

upGene <- explog2ratio_df[which.max(explog2ratio_df$explog2ratio), ] #finding the gen
e that has highest explog2ratio and saving it in variable upGene

upGene$gene_names #getting the genename that has highest explog2ratio
```

```
## [1] "248837_at"
```

h. Identify the gene (get the name of the gene) that has the lowest explog2ratio and save it as downGene. The lowest (most negative) log fold change means the original ratio was less than one, which means the treatment value was much lower than the control. So the expression decreased in treatment.

```
# Answer here

downGene <- explog2ratio_df[which.min(explog2ratio_df$explog2ratio), ] #finding the g
ene that has lowest explog2ratio and saving it in variable upGene

downGene$gene_names #getting the genename that has lowest explog2ratio
```

```
## [1] "250286_at"
```

i. Obtain the original expression values of bigGene and smallGene from expvalues and plot them together in a barplot side by side. Explain how the plot validates how your analysis.

```
# Answer here


upGene_values <- expvalues[rownames(expvalues) %in% upGene$gene_names, ] #filtering o
ut values for upGene and saving it in upGene_values variable

downGene_values <- expvalues[rownames(expvalues) %in% downGene$gene_names, ] #filteri
ng out values for downGene and saving it in downGene_values variable
```

```
upGene_values #printing out the upGene values
```

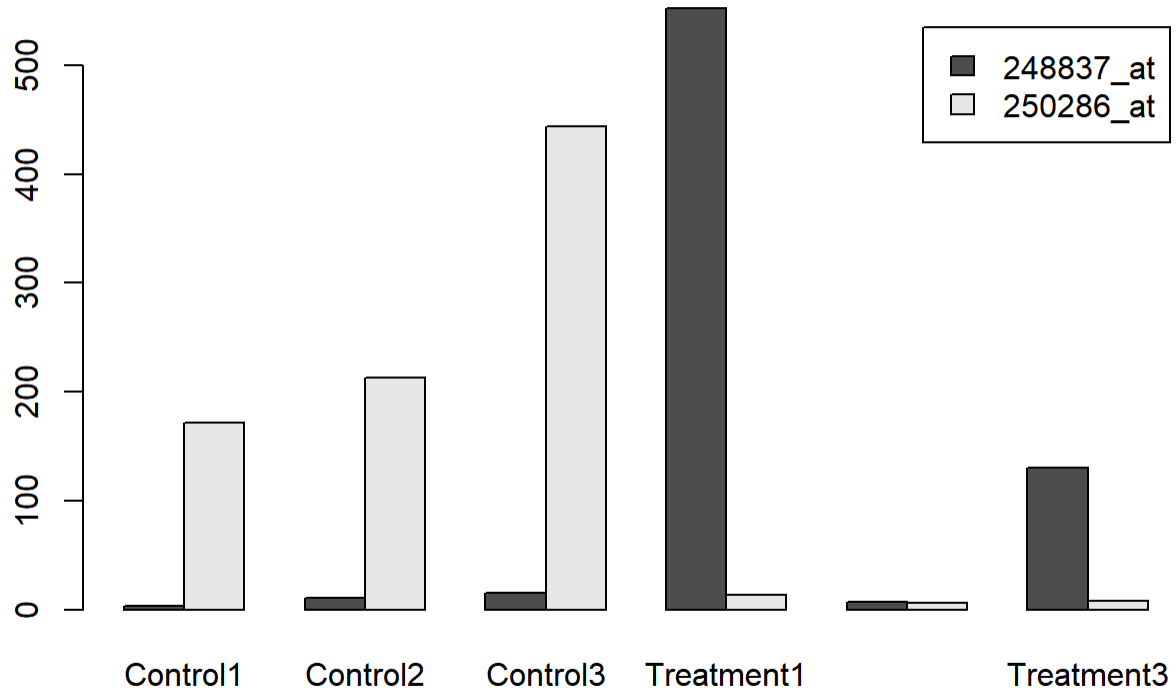
```
##           Control1 Control2 Control3 Treatment1 Treatment2 Treatment3
## 248837_at  3.65736  10.4063 15.29166   552.4859   7.231182    130.641
```

```
downGene_values #printing out the downGene values
```

```
##           Control1 Control2 Control3 Treatment1 Treatment2 Treatment3
## 250286_at 171.7833 213.4973 444.1292   13.39198   5.761501   7.627879
```

```
upGene_downGene <- rbind(upGene_values,downGene_values) #adding both upgene and downg
ene values in a df upGene_downGene
```

```
barplot(as.matrix(upGene_downGene), beside = T, legend.text = T) #plotting barplot wi
th data of both upGene and downGene side by side
```



Explain how the plot validates how your analysis.

Based on the above plot it looks like the upgene and downgene seems to have pretty different response in control vs treatment. During control 250286_at gene have higher expression levels whereas 248837_at gene have lower expression levels. Now in case of treatment its inverse, 250286_at gene have lower expression levels whereas 248837_at gene have higher expression levels which could mean 250286_at gene that is upregulated in control is inhibiting the other gene 248837_at but by treatment 250286_at gene is getting inhibited leading to over-expression of 248837_at gene.