Assessment 4 Part 1

Dulki Nihinsa Danthanarayana

2025-09-28

Downloading gene_expression.tsv data file

```
URL="https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/gene_expression.tsv"
download.file(URL,destfile="gene_expression.tsv")
```

The code defines the URL for a tsv file containing RNA-seq count data for three samples of interest, downloads the file as gene_expression.tsv

Downloading growth_data.csv data file

```
URL="https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/growth_data.csv"
download.file(URL,destfile="growth_data.csv")
```

The code defines the URL for a csv file containing measurements for tree circumference growing at two sites, control site and treatment site which were planted 20 years ago, downloads the file as growth_data.csv

1. Reading Gene Expression Data and Setting Row Names

```
gene_expr <- read.table("gene_expression.tsv", header=TRUE, sep="\t", row.names=1)
```

The read.table() import the TSV file, row.names = 1 set gene identifiers as row names.

```
head(gene_expr)
```

```
##                                GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                             0                        0
## ENSG00000227232.5_WASH7P                            187                      109
## ENSG00000278267.1_MIR6859-1                           0                        0
## ENSG00000243485.5_MIR1302-2HG                         1                        0
## ENSG00000237613.2_FAM138A                             0                        0
## ENSG00000268020.3_OR4G4P                              0                        1
##                                GTEX.1117F.0526.SM.5EGHJ
## ENSG00000223972.5_DDX11L1                             0
## ENSG00000227232.5_WASH7P                            143
## ENSG00000278267.1_MIR6859-1                           1
## ENSG00000243485.5_MIR1302-2HG                         0
## ENSG00000237613.2_FAM138A                             0
## ENSG00000268020.3_OR4G4P                              0
```

The head(gene_expr) command display the first six rows.

2. Calculating Gene Mean Expression

```
gene_expr$Mean <- rowMeans(gene_expr)
```

The mean across samples for each gene is calculated using rowMeans() and it is added as a new column named "Mean".

```
head(gene_expr)
```

```
##                                  GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                               0                        0
## ENSG00000227232.5_WASH7P                              187                      109
## ENSG00000278267.1_MIR6859-1                             0                        0
## ENSG00000243485.5_MIR1302-2HG                           1                        0
## ENSG00000237613.2_FAM138A                               0                        0
## ENSG00000268020.3_OR4G4P                                0                        1
##                                  GTEX.1117F.0526.SM.5EGHJ        Mean
## ENSG00000223972.5_DDX11L1                               0   0.0000000
## ENSG00000227232.5_WASH7P                              143 146.3333333
## ENSG00000278267.1_MIR6859-1                             1   0.3333333
## ENSG00000243485.5_MIR1302-2HG                           0   0.3333333
## ENSG00000237613.2_FAM138A                               0   0.0000000
## ENSG00000268020.3_OR4G4P                                0   0.3333333
```

The head(gene_expr) command shows the first six genes with the new column added.

3. Listing the 10 genes with the highest mean expression

```
top10 <- head(gene_expr[order(-gene_expr$Mean), ], 10)
top10
```

```
##                              GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000198804.2_MT-CO1                       267250                  1101779
## ENSG00000198886.2_MT-ND4                       273188                   991891
## ENSG00000198938.2_MT-CO3                       250277                  1041376
## ENSG00000198888.2_MT-ND1                       243853                   772966
## ENSG00000198899.2_MT-ATP6                      141374                   696715
## ENSG00000198727.2_MT-CYB                       127194                   638209
## ENSG00000198763.3_MT-ND2                       159303                   543786
## ENSG00000211445.11_GPX3                        464959                    39396
## ENSG00000198712.1_MT-CO2                       128858                   545360
## ENSG00000156508.17_EEF1A1                      317642                    39573
##                              GTEX.1117F.0526.SM.5EGHJ     Mean
## ENSG00000198804.2_MT-CO1                       218923 529317.3
## ENSG00000198886.2_MT-ND4                       277628 514235.7
## ENSG00000198938.2_MT-CO3                       223178 504943.7
## ENSG00000198888.2_MT-ND1                       194032 403617.0
## ENSG00000198899.2_MT-ATP6                      151166 329751.7
## ENSG00000198727.2_MT-CYB                       141359 302254.0
## ENSG00000198763.3_MT-ND2                       149564 284217.7
## ENSG00000211445.11_GPX3                        306070 270141.7
## ENSG00000198712.1_MT-CO2                       122816 265678.0
## ENSG00000156508.17_EEF1A1                      339347 232187.3
```

The above command sort the data frame by the new mean column in descending order.

```
rownames(top10)
```

```
##  [1] "ENSG00000198804.2_MT-CO1"  "ENSG00000198886.2_MT-ND4"
##  [3] "ENSG00000198938.2_MT-CO3"  "ENSG00000198888.2_MT-ND1"
##  [5] "ENSG00000198899.2_MT-ATP6" "ENSG00000198727.2_MT-CYB"
##  [7] "ENSG00000198763.3_MT-ND2"  "ENSG00000211445.11_GPX3"
##  [9] "ENSG00000198712.1_MT-CO2"  "ENSG00000156508.17_EEF1A1"
```

The above command display the names of the top 10 genes.

4. Determining the number of genes with a mean <10

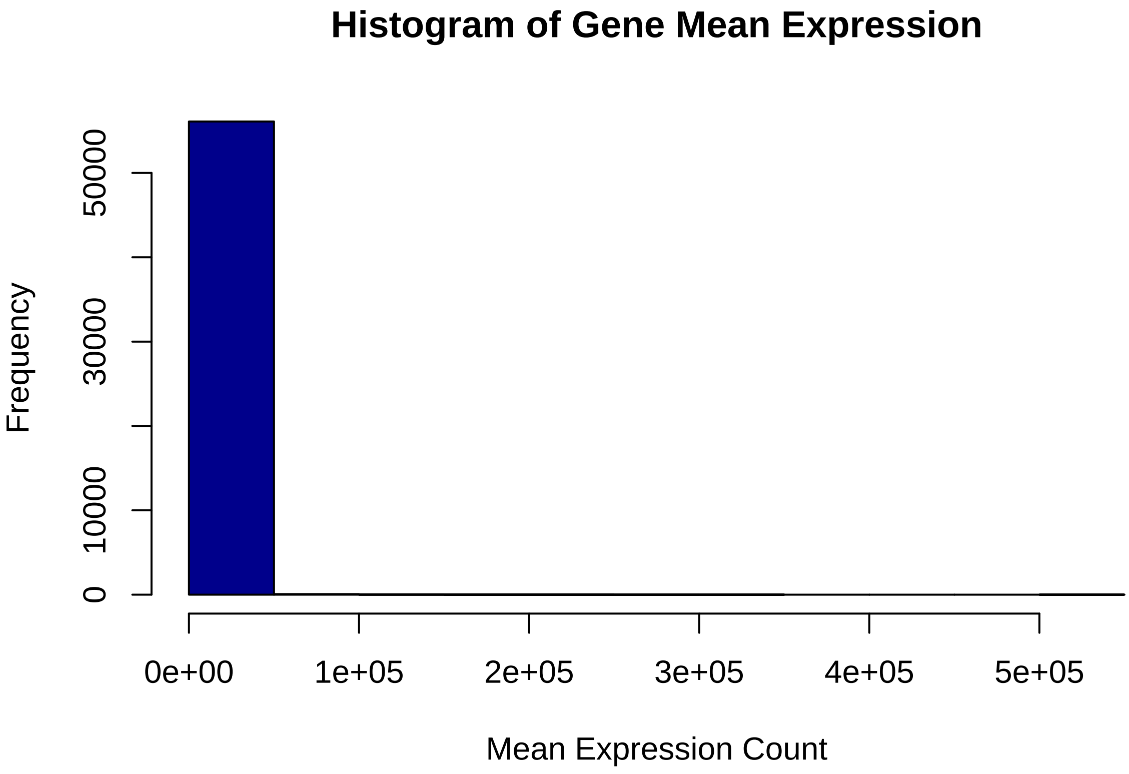
```
low_genes_count <- sum(gene_expr$Mean < 10)
low_genes_count
```

```
## [1] 35988
```

The sum(gene_expr$Mean < 10) is used on the mean column to count genes with mean expression less than 10.

5. Plotting Histogram of Mean Expression

```
hist(gene_expr$Mean,
     main="Histogram of Gene Mean Expression",
     xlab="Mean Expression Count",
     col="darkblue", border="black")
```



```r
png("myplot.png")
hist(gene_expr$Mean,
     main="Histogram of Gene Mean Expression",
     xlab="Mean Expression Count",
     col="darkblue", border="black")
dev.off()
```

```
## pdf
##   2
```

The plot is saved as a PNG using png("myplot.png").

6. Importing and Exploring Tree Growth Data

```r
growth <- read.csv("growth_data.csv", header=TRUE)
colnames(growth)
```

```
## [1] "Site"            "TreeID"          "Circumf_2005_cm" "Circumf_2010_cm"
## [5] "Circumf_2015_cm" "Circumf_2020_cm"
```

The read.csv("growth_data.csv", header=TRUE) command import csv file and column names are display by the colnames(growth).

7. Calculating the mean and standard deviation of tree circumference at the start and end of the study at both sites

```r
# Calculating mean and SD for Circumference in 2005 by Site

mean_2005_northeast <- mean(subset(growth, Site == "northeast")$Circumf_2005_cm, na.rm = TRUE)
mean_2005_northeast
```

```
## [1] 5.292
```

```r
sd_2005_northeast <- sd(subset(growth, Site == "northeast")$Circumf_2005_cm, na.rm = TRUE)
sd_2005_northeast
```

```
## [1] 0.9140267
```

```r
mean_2005_southwest <- mean(subset(growth, Site == "southwest")$Circumf_2005_cm, na.rm = TRUE)
mean_2005_southwest
```

```
## [1] 4.862
```

```r
sd_2005_southwest <- sd(subset(growth, Site == "southwest")$Circumf_2005_cm, na.rm = TRUE)
sd_2005_southwest
```

```
## [1] 1.147471
```

```r
# Calculating mean and SD for Circumference in 2020 by Site

mean_2020_northeast <- mean(subset(growth, Site == "northeast")$Circumf_2020_cm, na.rm = TRUE)
mean_2020_northeast
```

```
## [1] 54.228
```

```r
sd_2020_northeast <- sd(subset(growth, Site == "northeast")$Circumf_2020_cm, na.rm = TRUE)
sd_2020_northeast
```

```
## [1] 25.22795
```

```
mean_2020_southwest <- mean(subset(growth, Site == "southwest")$Circumf_2020_cm, na.rm = TRUE)
mean_2020_southwest
```

```
## [1] 45.596
```

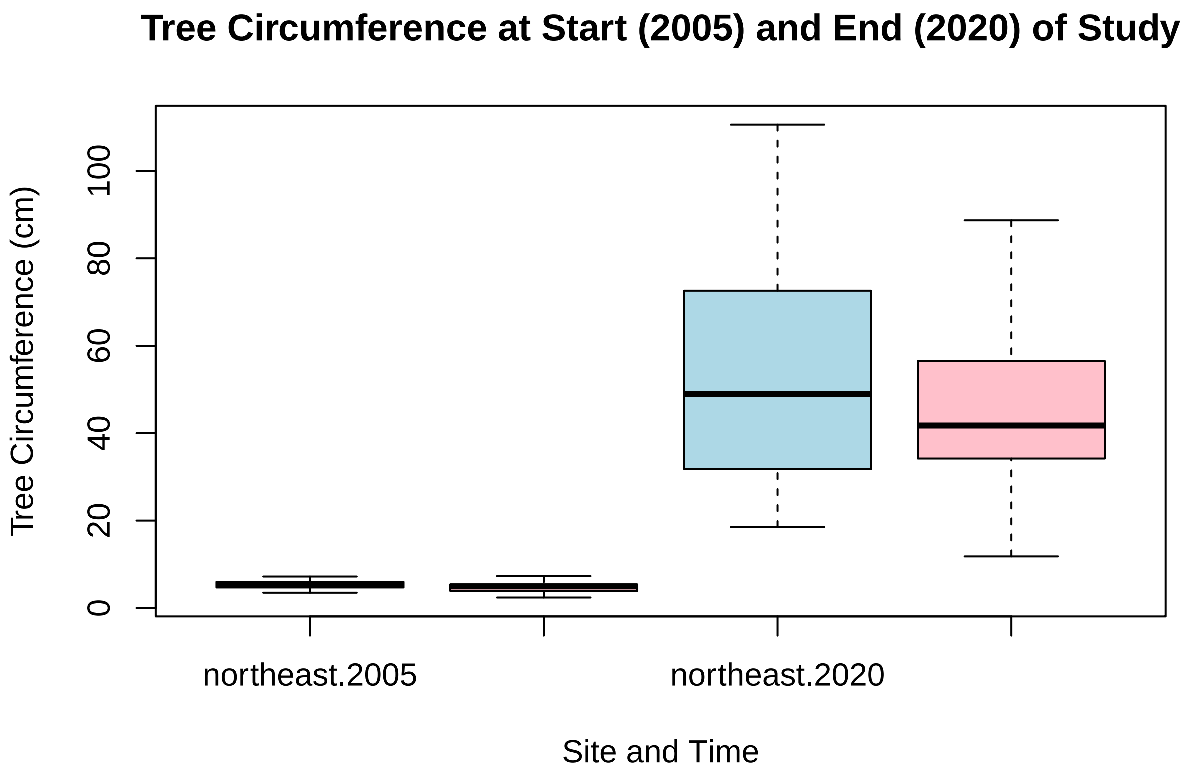
```
sd_2020_southwest <- sd(subset(growth, Site == "southwest")$Circumf_2020_cm, na.rm = TRUE)
sd_2020_southwest
```

```
## [1] 17.87345
```

8. Box plot of tree circumference at the start and end of the study at both sites

```
# Combine start and end circumference data with Site info in long format
circum_data <- data.frame(
  Circumference = c(growth$Circumf_2005_cm, growth$Circumf_2020_cm),
  Time = factor(rep(c("2005", "2020"), each = nrow(growth))),
  Site = rep(growth$Site, 2)
)

# Create boxplot of Circumference by Site and Time
boxplot(Circumference ~ Site + Time, data = circum_data,
        col = c("lightblue", "pink"),
        xlab = "Site and Time",
        ylab = "Tree Circumference (cm)",
        main = "Tree Circumference at Start (2005) and End (2020) of Study")
```



9. Calculating the mean growth over the last 10 years at each site

```r
# Calculating growth over last 10 years (2020 - 2010)
growth$Growth_10yr <- growth$Circumf_2020_cm - growth$Circumf_2010_cm

# Calculating mean growth by site
mean_growth_northeast <- mean(subset(growth, Site == "northeast")$Growth_10yr, na.rm = TRUE)
mean_growth_northeast
```

```
## [1] 42.94
```

```r
mean_growth_southwest <- mean(subset(growth, Site == "southwest")$Growth_10yr, na.rm = TRUE)
mean_growth_southwest
```

```
## [1] 35.49
```

10. T-test of 10-Year Growth at Two Sites

```r
t_result <- t.test(Growth_10yr ~ Site, data = growth)
t_result
```

```
##
##  Welch Two Sample t-test
##
## data:  Growth_10yr by Site
## t = 1.8882, df = 87.978, p-value = 0.06229
## alternative hypothesis: true difference in means between group northeast and group southwest is not e
## 95 percent confidence interval:
##  -0.3909251 15.2909251
## sample estimates:
## mean in group northeast mean in group southwest
##                   42.94                   35.49
```