Part 1: Imported files, data analysis, operations, plots and Github

Reposity link for my assignment: https://github.com/LinhChi1323/SLE712_Assignment_3

Source for part 1: https://github.com/LinhChi1323/SLE712_Assignment_3/tree/master/part1

Question 1

Download file `gene_experession.tsv` from github by `download.file`, read file by `read.csv`, make row name is column one and print first 6 genes by `head()`.

```r
# download the file gene_expression.tsv
download.file("https://github.com/markziemann/SLE712_files/raw/master/bioinfo_asst3_part1_files/gene_exp

# read in the file
df <- read.csv('gene_expression.tsv', sep='\t', stringsAsFactors = FALSE, row.names = 1)

#Try to access a gene by gene name
df['ENSG00000223972', ]
```

```
##                 SRR5150592 SRR5150593
## ENSG00000223972          1          0
```

```r
# show first 6 genes
head(df, 6)
```

```
##                 SRR5150592 SRR5150593
## ENSG00000223972          1          0
## ENSG00000227232          0          1
## ENSG00000278267          0          0
## ENSG00000243485          0          0
## ENSG00000284332          0          0
## ENSG00000237613          0          0
```

Question 2:

Make mean column which is the mean of other columns by `rowMeans`, show first 6 genes by `head()`.

```r
#make mean column
df$mean <- rowMeans(df[, 1:2])

#show first 6 genes
head(df, 6)
```

```
##                 SRR5150592 SRR5150593 mean
## ENSG00000223972          1          0  0.5
## ENSG00000227232          0          1  0.5
## ENSG00000278267          0          0  0.0
## ENSG00000243485          0          0  0.0
## ENSG00000284332          0          0  0.0
## ENSG00000237613          0          0  0.0
```

Question 3

Using `order()`to sort the mean expession from lowest to highest. Take 10 highest genes by `tail()`.

```
# create sorted dataframe by ordered mean column
sorted_df <- df[order(df$mean), ]

# take the name of top 10-highest mean expression genes
top10genes <- row.names(tail(sorted_df, 10))

# look at the result
top10genes
```

```
##  [1] "ENSG00000108821" "ENSG00000198712" "ENSG00000196924" "ENSG00000198786"
##  [5] "ENSG00000198804" "ENSG00000137801" "ENSG00000198886" "ENSG00000075624"
##  [9] "ENSG00000210082" "ENSG00000115414"
```

Question 4

Take mean column after compare this column with 10 (<10). The resutl will be a list of boolean array. Number of genes with mean lower than 10 is sum of this list, which is 43124.

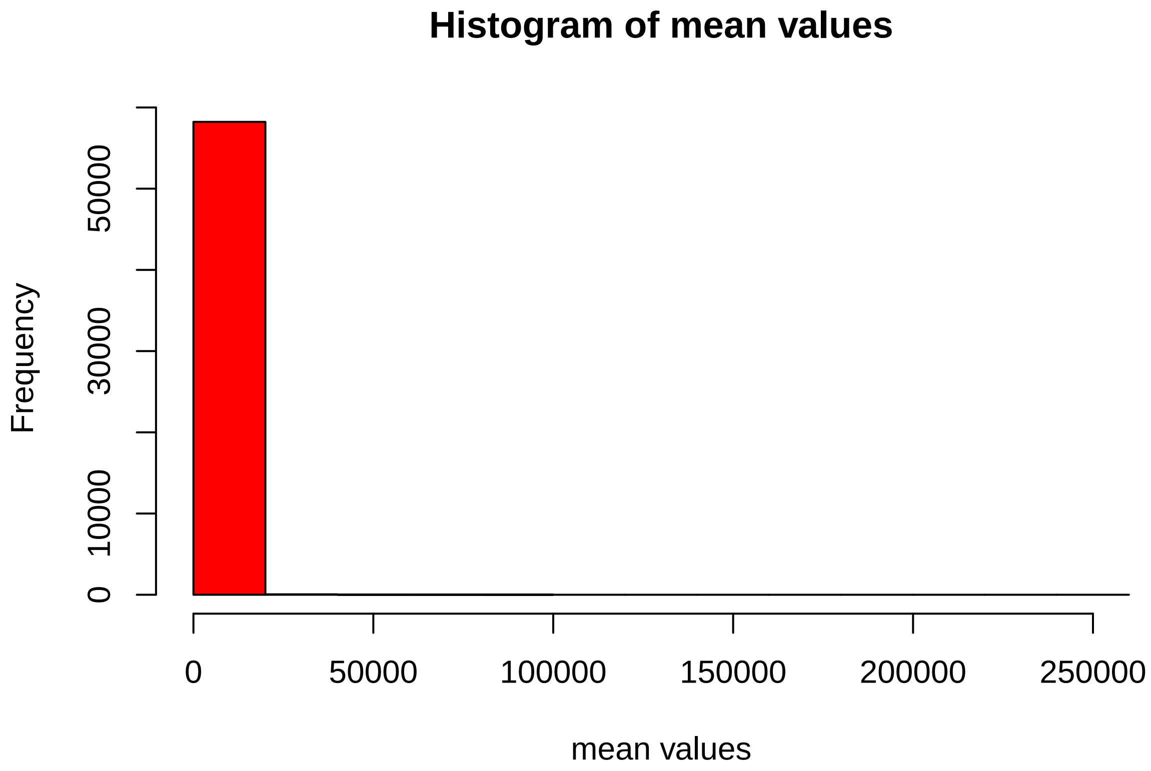
```
number_genes <- sum(df$mean < 10)
cat("The number of genes with a mean < 10: ", number_genes)
```

```
## The number of genes with a mean < 10:  43124
```

Question 5

Using `hist()` function to make a histogram plot and `png` to save it in png format. For my case, I directly show the histogram through running code on Rmarkdown file. In other cases, if you want to save the histogram into a file, `png` and `dev.off` can be used .

```
# png(file = "histogram.png")   # provide histogram file name.
#
# make ahistogram plot of mean values
hist(df$mean, main = "Histogram of mean values",xlab='mean values', col = "red")
```



```
#dev.off()    # save histogram file in png format
```

Question 6

Download growth data from github and load it into a dataframe by `read.csv`. Using command `str` and `head` to confirm that the data has been properly imported. Using `colnames` command to get the column names.

```
# dowload the file growth_data.csv
download.file("https://github.com/markziemann/SLE712_files/raw/master/bioinfo_asst3_part1_files/growth_

# load it into an R object
df <- read.csv('growth_data.csv', header = TRUE, stringsAsFactors = FALSE)

# Print column names of dataframe
colnames(df)
```

```
## [1] "Site"           "TreeID"         "Circumf_2004_cm" "Circumf_2009_cm"
## [5] "Circumf_2014_cm" "Circumf_2019_cm"
```

Question 7

Using `mean` and `sd` to calculate the mean and standard deviation (sd). Mean and sd of tree circumference at the start year (2004) at both sites are 5.077cm and 1.054462 respectively. Mean and sd of tree circumference at the end year (2019) at both sites are 49.912cm and 22.17979 respectively.

```
#Mean and standard deviation at 2004 (start) at both sites
mean_2004 <- mean(df$Circumf_2004_cm)
sd_2004 <- sd(df$Circumf_2004_cm)

cat(’Mean at 2004: ’, mean_2004)
```

```
## Mean at 2004:  5.077
```

```
cat(’\n’)
```

```
cat(’SD at 2004: ’, sd_2004)
```

```
## SD at 2004:  1.054462
```

```
# mean and standard deviation at 2019 (end) at both sites
mean_2019 <- mean(df$Circumf_2019_cm)
sd_2019 <- sd(df$Circumf_2019_cm)

cat(’\n’)
```

```
cat(’Mean at 2019: ’, mean_2019)
```

```
## Mean at 2019:  49.912
```

```
cat(’\n’)
```

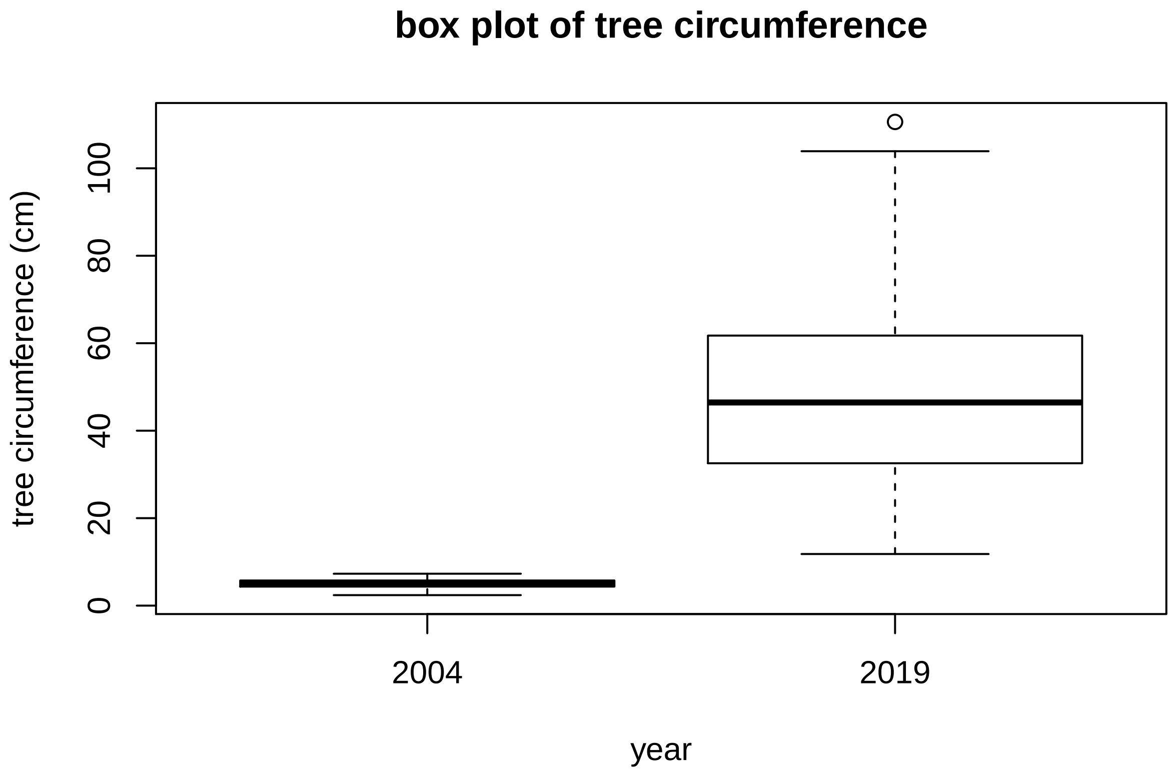
```
cat(’SD at 2019: ’, sd_2019)
```

```
## SD at 2019:  22.17979
```

Question 8

Using `boxplot` command to create a box plot of tree circumference at the start year(2004) and the end year (2019) at both sites.

```
boxplot(df$Circumf_2004_cm, df$Circumf_2019_cm,
    names=c("2004", "2019"), xlab = "year", ylab = "tree circumference (cm)", main = "box plot of tree
```



Question 9

Create a column growth (difference between 2009 and 2019) over the past 10 years for both site. Get growth values for each site and calculate mean. Mean growth over past 10 year for northeast and southwest are 30.076 cm and 48.354 cm respectively.

```
# calculate the difference of tree circumference between 2009 and 2019
df$growth <- df$Circumf_2019_cm - df$Circumf_2009_cm

# get growth values for each site
north_growth <- df[df$Site=="northeast", ]$growth
south_growth <- df[df$Site=="southwest", ]$growth

# calculate the mean for each site
mean_northeast <- mean(north_growth)
mean_southwest <- mean(south_growth)

cat("Mean growth of Northeast over the past  10 years: ", mean_northeast, '\n')
```

```
## Mean growth of Northeast over the past  10 years:  30.076
```

```
cat("Mean growth of Southwest over the past  10 years: ", mean_southwest)
```

```
## Mean growth of Southwest over the past  10 years:  48.354
```

Question 10:

Two functions `t.test` and `wilcox.test` are used to perform hypothesis tests (t test and wilcoxon test).p-value of t-test is 1.712524e-06 and p-value of wilcoxon test is 4.6264e-06.

```
# run t-test
t_test_res <- t.test(north_growth, south_growth)
# take p-value
t_test_pvalue <- t_test_res$p.value

#run Wilcoxon test
wilcox_test_res <- wilcox.test(north_growth, south_growth)
# take the p-value
wilcox_pvalue <- wilcox_test_res$p.value

cat(’p-value of t-test: ’, t_test_pvalue, ’\n’)
```

```
## p-value of t-test:  1.712524e-06
```

```
cat(’p-value of wilcoxon test: ’, wilcox_pvalue)
```

```
## p-value of wilcoxon test:  4.6264e-06
```