HW2_Kim

Jaiyool Kim

9/7/2019

Problem 1 : Work through the "R Programming E" lesson parts 4-7, 14

(optional 12 - only takes 5 min). From the R command prompt:

```
install.packages("swirl")
library(swirl)
install_course("R_Programming_E")
swirl()
```

I have taken all of the coures you mentioned at PM 10:21 on sunday (Sep. 8th), and sent you e-mails related to the history I took.

Problem 2 : Create a new R Markdown file within your local GitHub repo folder

(file–>new–>R Markdown–>save as).

The filename should be: HW2_lastname, i.e. for me it would be HW2_Settlage

You will use this new R Markdown file to solve problems 3-5.

I made this R Markdown file whose names is HW2_Kim.

Problem 3 : In the lecture, there were two links to StackOverflow questions on why one should use version control.

In your own words, summarize in 2-3 sentences how you think version control can help you in the classroom.

Actually, I think version control can be very useful in a variety of ways in this class.

First, version control can make it easier for professor and me to communicate with each other via GitHub and share the information simultaneously using good tools such as Forking and rewinding and Collaboration. Even if there are some comments related to appropriateness and necessities of version control from solo data-analysts, I, as a novice of R & Github, also one of students learning the mechanisms of the infrastructure of collaborating the information among researchers, these kinds of version controls would be needed for me as I think.

Second, version control can allow myself to think numerous ways about solving problems or issues. It will be especially helpful when I consider doing different ways to conduct a certain kind of project or research as I think. Using Branches, I can keep track of all my branches of thinkings to do researches or any assignments, which can make me comfortable in terms of being arranged automatically even though I did not make the individual folder in my local repository (such as my MacBook).

In addition to the above pros, there will be so many useful and informative points when using version control as I expect !!.

Problem 4

In this exercise, you will import, munge, clean and summarize datasets from Wu and Hamada's Experiments:

Planning, Design and Analysis book you will use in the Spring. For each one, please weave your code and text to describe both your process and observations. Make sure you create a tidy dataset describing the variables, create a summary table of the data, note issues with the data.

a. Sensory data from five operators. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

c. Brain weight (g) and body weight (kg) for 62 species. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat

a. Sensory data from five operators.

```
Sensory.data <- read.csv("Sensory.csv",header=FALSE,sep=" ",stringsAsFactors = FALSE)
Sensory.data
```

```
##            V1  V2  V3  V4  V5  V6
## 1  \tOperator  NA  NA  NA  NA  NA
## 2        Item 1.0 2.0 3.0 4.0 5.0
## 3           1 4.3 4.9 3.3 5.3 4.4
## 4         4.3 4.5 4.0 5.5 3.3  NA
## 5         4.1 5.3 3.4 5.7 4.7  NA
## 6           2 6.0 5.3 4.5 5.9 4.7
## 7         4.9 6.3 4.2 5.5 4.9  NA
## 8         6.0 5.9 4.7 6.3 4.6  NA
## 9           3 2.4 2.5 2.3 3.1 2.4
## 10        3.9 3.0 2.8 2.7 1.3  NA
## 11        1.9 3.9 2.6 4.6 2.2  NA
## 12          4 7.4 8.2 6.4 6.8 6.0
## 13        7.1 7.9 5.9 7.3 6.1  NA
## 14        6.4 7.1 6.9 7.0 6.7  NA
## 15          5 5.7 6.3 5.4 6.1 5.9
## 16        5.8 5.7 5.4 6.2 6.5  NA
## 17        5.8 6.0 6.1 7.0 4.9  NA
## 18          6 2.2 2.4 1.7 3.4 1.7
## 19        3.0 1.8 2.1 4.0 1.7  NA
## 20        2.1 3.3 1.1 3.3 2.1  NA
## 21          7 1.2 1.5 1.2 0.9 0.7
## 22        1.3 2.4 0.8 1.2 1.3  NA
## 23        0.9 3.1 1.1 1.9 1.6  NA
## 24          8 4.2 4.8 4.5 4.6 3.2
## 25        3.0 4.5 4.7 4.9 4.6  NA
## 26        4.8 4.8 4.7 4.8 4.3  NA
## 27          9 8.0 8.6 9.0 9.4 8.8
## 28        9.0 7.7 6.7 9.0 7.9  NA
## 29        8.9 9.2 8.1 9.1 7.6  NA
```

```
## 30         10 5.0 4.8 3.9 5.5 3.8
## 31        5.4 5.0 3.4 4.9 4.6  NA
## 32        2.8 5.2 4.1 3.9 5.5  NA
```

```
## Set the column names as the follow :
first.data.names <- c("ID","1st op.","2nd op.","3rd op.","4th op.","5th op.")

## Check the type of the given data
typeof(Sensory.data) # list
```

```
## [1] "list"
```

```
## Change the form of the given data to matrix.
Sensory.data <-as.matrix(Sensory.data)

## Assign the column names using the character vector we made at the above line.
colnames(Sensory.data) <- first.data.names

## Show the full data set => The first two row should be eliminated for cleaning the data set.
Sensory.data
```

```
##       ID           1st op. 2nd op. 3rd op. 4th op. 5th op.
##  [1,] "\tOperator" NA      NA      NA      NA      NA
##  [2,] "Item"       "1.0"   "2.0"   "3.0"   "4.0"   "5.0"
##  [3,] "1"          "4.3"   "4.9"   "3.3"   "5.3"   "4.4"
##  [4,] "4.3"        "4.5"   "4.0"   "5.5"   "3.3"   NA
##  [5,] "4.1"        "5.3"   "3.4"   "5.7"   "4.7"   NA
##  [6,] "2"          "6.0"   "5.3"   "4.5"   "5.9"   "4.7"
##  [7,] "4.9"        "6.3"   "4.2"   "5.5"   "4.9"   NA
##  [8,] "6.0"        "5.9"   "4.7"   "6.3"   "4.6"   NA
##  [9,] "3"          "2.4"   "2.5"   "2.3"   "3.1"   "2.4"
## [10,] "3.9"        "3.0"   "2.8"   "2.7"   "1.3"   NA
## [11,] "1.9"        "3.9"   "2.6"   "4.6"   "2.2"   NA
## [12,] "4"          "7.4"   "8.2"   "6.4"   "6.8"   "6.0"
## [13,] "7.1"        "7.9"   "5.9"   "7.3"   "6.1"   NA
## [14,] "6.4"        "7.1"   "6.9"   "7.0"   "6.7"   NA
## [15,] "5"          "5.7"   "6.3"   "5.4"   "6.1"   "5.9"
## [16,] "5.8"        "5.7"   "5.4"   "6.2"   "6.5"   NA
## [17,] "5.8"        "6.0"   "6.1"   "7.0"   "4.9"   NA
## [18,] "6"          "2.2"   "2.4"   "1.7"   "3.4"   "1.7"
## [19,] "3.0"        "1.8"   "2.1"   "4.0"   "1.7"   NA
## [20,] "2.1"        "3.3"   "1.1"   "3.3"   "2.1"   NA
## [21,] "7"          "1.2"   "1.5"   "1.2"   "0.9"   "0.7"
## [22,] "1.3"        "2.4"   "0.8"   "1.2"   "1.3"   NA
## [23,] "0.9"        "3.1"   "1.1"   "1.9"   "1.6"   NA
## [24,] "8"          "4.2"   "4.8"   "4.5"   "4.6"   "3.2"
## [25,] "3.0"        "4.5"   "4.7"   "4.9"   "4.6"   NA
## [26,] "4.8"        "4.8"   "4.7"   "4.8"   "4.3"   NA
## [27,] "9"          "8.0"   "8.6"   "9.0"   "9.4"   "8.8"
## [28,] "9.0"        "7.7"   "6.7"   "9.0"   "7.9"   NA
## [29,] "8.9"        "9.2"   "8.1"   "9.1"   "7.6"   NA
## [30,] "10"         "5.0"   "4.8"   "3.9"   "5.5"   "3.8"
## [31,] "5.4"        "5.0"   "3.4"   "4.9"   "4.6"   NA
## [32,] "2.8"        "5.2"   "4.1"   "3.9"   "5.5"   NA
```

```r
## Eliminate the first two row
Sensory.data <- Sensory.data[-c(1:2),]

## Show the (a little bit arranged) data set again.
Sensory.data
```

```
##       ID    1st op. 2nd op. 3rd op. 4th op. 5th op.
##  [1,] "1"   "4.3"   "4.9"   "3.3"   "5.3"   "4.4"
##  [2,] "4.3" "4.5"   "4.0"   "5.5"   "3.3"   NA
##  [3,] "4.1" "5.3"   "3.4"   "5.7"   "4.7"   NA
##  [4,] "2"   "6.0"   "5.3"   "4.5"   "5.9"   "4.7"
##  [5,] "4.9" "6.3"   "4.2"   "5.5"   "4.9"   NA
##  [6,] "6.0" "5.9"   "4.7"   "6.3"   "4.6"   NA
##  [7,] "3"   "2.4"   "2.5"   "2.3"   "3.1"   "2.4"
##  [8,] "3.9" "3.0"   "2.8"   "2.7"   "1.3"   NA
##  [9,] "1.9" "3.9"   "2.6"   "4.6"   "2.2"   NA
## [10,] "4"   "7.4"   "8.2"   "6.4"   "6.8"   "6.0"
## [11,] "7.1" "7.9"   "5.9"   "7.3"   "6.1"   NA
## [12,] "6.4" "7.1"   "6.9"   "7.0"   "6.7"   NA
## [13,] "5"   "5.7"   "6.3"   "5.4"   "6.1"   "5.9"
## [14,] "5.8" "5.7"   "5.4"   "6.2"   "6.5"   NA
## [15,] "5.8" "6.0"   "6.1"   "7.0"   "4.9"   NA
## [16,] "6"   "2.2"   "2.4"   "1.7"   "3.4"   "1.7"
## [17,] "3.0" "1.8"   "2.1"   "4.0"   "1.7"   NA
## [18,] "2.1" "3.3"   "1.1"   "3.3"   "2.1"   NA
## [19,] "7"   "1.2"   "1.5"   "1.2"   "0.9"   "0.7"
## [20,] "1.3" "2.4"   "0.8"   "1.2"   "1.3"   NA
## [21,] "0.9" "3.1"   "1.1"   "1.9"   "1.6"   NA
## [22,] "8"   "4.2"   "4.8"   "4.5"   "4.6"   "3.2"
## [23,] "3.0" "4.5"   "4.7"   "4.9"   "4.6"   NA
## [24,] "4.8" "4.8"   "4.7"   "4.8"   "4.3"   NA
## [25,] "9"   "8.0"   "8.6"   "9.0"   "9.4"   "8.8"
## [26,] "9.0" "7.7"   "6.7"   "9.0"   "7.9"   NA
## [27,] "8.9" "9.2"   "8.1"   "9.1"   "7.6"   NA
## [28,] "10"  "5.0"   "4.8"   "3.9"   "5.5"   "3.8"
## [29,] "5.4" "5.0"   "3.4"   "4.9"   "4.6"   NA
## [30,] "2.8" "5.2"   "4.1"   "3.9"   "5.5"   NA
```

```r
## From the above shown data set, we need to fill the NA value with corresponding the value located in
## That is, for example, the (2,6)th element NA value should be exchanged with the (2,1)th element 4.3
## Plus, the (2,1)th element should be filled with '2' (second ID number)

## We need to adjust the data set for making what I have mentioned above.

## Let's make the function that allows us to do the above thing.

NA.correction.function <- function(data){

  for(i in 1 : nrow(data) )
    for(j in 1 : ncol(data) )

      if(is.na(data[i,j])==TRUE) {

        data[i,j] <- data[i,1]
```

```
        data[i,1] <- i

      }

  return(data)

}


Arranged.data <- NA.correction.function(Sensory.data)

Arranged.data
```

```
##       ID   1st op. 2nd op. 3rd op. 4th op. 5th op.
##  [1,] "1"  "4.3"   "4.9"   "3.3"   "5.3"   "4.4"
##  [2,] "2"  "4.5"   "4.0"   "5.5"   "3.3"   "4.3"
##  [3,] "3"  "5.3"   "3.4"   "5.7"   "4.7"   "4.1"
##  [4,] "2"  "6.0"   "5.3"   "4.5"   "5.9"   "4.7"
##  [5,] "5"  "6.3"   "4.2"   "5.5"   "4.9"   "4.9"
##  [6,] "6"  "5.9"   "4.7"   "6.3"   "4.6"   "6.0"
##  [7,] "3"  "2.4"   "2.5"   "2.3"   "3.1"   "2.4"
##  [8,] "8"  "3.0"   "2.8"   "2.7"   "1.3"   "3.9"
##  [9,] "9"  "3.9"   "2.6"   "4.6"   "2.2"   "1.9"
## [10,] "4"  "7.4"   "8.2"   "6.4"   "6.8"   "6.0"
## [11,] "11" "7.9"   "5.9"   "7.3"   "6.1"   "7.1"
## [12,] "12" "7.1"   "6.9"   "7.0"   "6.7"   "6.4"
## [13,] "5"  "5.7"   "6.3"   "5.4"   "6.1"   "5.9"
## [14,] "14" "5.7"   "5.4"   "6.2"   "6.5"   "5.8"
## [15,] "15" "6.0"   "6.1"   "7.0"   "4.9"   "5.8"
## [16,] "6"  "2.2"   "2.4"   "1.7"   "3.4"   "1.7"
## [17,] "17" "1.8"   "2.1"   "4.0"   "1.7"   "3.0"
## [18,] "18" "3.3"   "1.1"   "3.3"   "2.1"   "2.1"
## [19,] "7"  "1.2"   "1.5"   "1.2"   "0.9"   "0.7"
## [20,] "20" "2.4"   "0.8"   "1.2"   "1.3"   "1.3"
## [21,] "21" "3.1"   "1.1"   "1.9"   "1.6"   "0.9"
## [22,] "8"  "4.2"   "4.8"   "4.5"   "4.6"   "3.2"
## [23,] "23" "4.5"   "4.7"   "4.9"   "4.6"   "3.0"
## [24,] "24" "4.8"   "4.7"   "4.8"   "4.3"   "4.8"
## [25,] "9"  "8.0"   "8.6"   "9.0"   "9.4"   "8.8"
## [26,] "26" "7.7"   "6.7"   "9.0"   "7.9"   "9.0"
## [27,] "27" "9.2"   "8.1"   "9.1"   "7.6"   "8.9"
## [28,] "10" "5.0"   "4.8"   "3.9"   "5.5"   "3.8"
## [29,] "29" "5.0"   "3.4"   "4.9"   "4.6"   "5.4"
## [30,] "30" "5.2"   "4.1"   "3.9"   "5.5"   "2.8"
```

```
## We have completed what we wanted to do, however we have to align the ID number in order (1st column)
## So I decide to weave another function to do that.

ID.align.function <- function(data) {

  for (i in 1 : nrow(data) )
    if(data[i,1]!=1) data[i,1] <- i

    return(data)
```

```
}

## Let's apply this function to the data

Arranged.data <- ID.align.function(Arranged.data)

Arranged.data
```

```
##       ID   1st op. 2nd op. 3rd op. 4th op. 5th op.
##  [1,] "1"  "4.3"   "4.9"   "3.3"   "5.3"   "4.4"
##  [2,] "2"  "4.5"   "4.0"   "5.5"   "3.3"   "4.3"
##  [3,] "3"  "5.3"   "3.4"   "5.7"   "4.7"   "4.1"
##  [4,] "4"  "6.0"   "5.3"   "4.5"   "5.9"   "4.7"
##  [5,] "5"  "6.3"   "4.2"   "5.5"   "4.9"   "4.9"
##  [6,] "6"  "5.9"   "4.7"   "6.3"   "4.6"   "6.0"
##  [7,] "7"  "2.4"   "2.5"   "2.3"   "3.1"   "2.4"
##  [8,] "8"  "3.0"   "2.8"   "2.7"   "1.3"   "3.9"
##  [9,] "9"  "3.9"   "2.6"   "4.6"   "2.2"   "1.9"
## [10,] "10" "7.4"   "8.2"   "6.4"   "6.8"   "6.0"
## [11,] "11" "7.9"   "5.9"   "7.3"   "6.1"   "7.1"
## [12,] "12" "7.1"   "6.9"   "7.0"   "6.7"   "6.4"
## [13,] "13" "5.7"   "6.3"   "5.4"   "6.1"   "5.9"
## [14,] "14" "5.7"   "5.4"   "6.2"   "6.5"   "5.8"
## [15,] "15" "6.0"   "6.1"   "7.0"   "4.9"   "5.8"
## [16,] "16" "2.2"   "2.4"   "1.7"   "3.4"   "1.7"
## [17,] "17" "1.8"   "2.1"   "4.0"   "1.7"   "3.0"
## [18,] "18" "3.3"   "1.1"   "3.3"   "2.1"   "2.1"
## [19,] "19" "1.2"   "1.5"   "1.2"   "0.9"   "0.7"
## [20,] "20" "2.4"   "0.8"   "1.2"   "1.3"   "1.3"
## [21,] "21" "3.1"   "1.1"   "1.9"   "1.6"   "0.9"
## [22,] "22" "4.2"   "4.8"   "4.5"   "4.6"   "3.2"
## [23,] "23" "4.5"   "4.7"   "4.9"   "4.6"   "3.0"
## [24,] "24" "4.8"   "4.7"   "4.8"   "4.3"   "4.8"
## [25,] "25" "8.0"   "8.6"   "9.0"   "9.4"   "8.8"
## [26,] "26" "7.7"   "6.7"   "9.0"   "7.9"   "9.0"
## [27,] "27" "9.2"   "8.1"   "9.1"   "7.6"   "8.9"
## [28,] "28" "5.0"   "4.8"   "3.9"   "5.5"   "3.8"
## [29,] "29" "5.0"   "3.4"   "4.9"   "4.6"   "5.4"
## [30,] "30" "5.2"   "4.1"   "3.9"   "5.5"   "2.8"
```

```
## Change the type of the elememt (from character to numeric) of arranged matrix
mode(Arranged.data) <- "numeric"

Arranged.data
```

```
##       ID 1st op. 2nd op. 3rd op. 4th op. 5th op.
##  [1,]  1     4.3     4.9     3.3     5.3     4.4
##  [2,]  2     4.5     4.0     5.5     3.3     4.3
##  [3,]  3     5.3     3.4     5.7     4.7     4.1
##  [4,]  4     6.0     5.3     4.5     5.9     4.7
##  [5,]  5     6.3     4.2     5.5     4.9     4.9
##  [6,]  6     5.9     4.7     6.3     4.6     6.0
##  [7,]  7     2.4     2.5     2.3     3.1     2.4
##  [8,]  8     3.0     2.8     2.7     1.3     3.9
```

```
##  [9,]  9     3.9     2.6     4.6     2.2     1.9
## [10,] 10     7.4     8.2     6.4     6.8     6.0
## [11,] 11     7.9     5.9     7.3     6.1     7.1
## [12,] 12     7.1     6.9     7.0     6.7     6.4
## [13,] 13     5.7     6.3     5.4     6.1     5.9
## [14,] 14     5.7     5.4     6.2     6.5     5.8
## [15,] 15     6.0     6.1     7.0     4.9     5.8
## [16,] 16     2.2     2.4     1.7     3.4     1.7
## [17,] 17     1.8     2.1     4.0     1.7     3.0
## [18,] 18     3.3     1.1     3.3     2.1     2.1
## [19,] 19     1.2     1.5     1.2     0.9     0.7
## [20,] 20     2.4     0.8     1.2     1.3     1.3
## [21,] 21     3.1     1.1     1.9     1.6     0.9
## [22,] 22     4.2     4.8     4.5     4.6     3.2
## [23,] 23     4.5     4.7     4.9     4.6     3.0
## [24,] 24     4.8     4.7     4.8     4.3     4.8
## [25,] 25     8.0     8.6     9.0     9.4     8.8
## [26,] 26     7.7     6.7     9.0     7.9     9.0
## [27,] 27     9.2     8.1     9.1     7.6     8.9
## [28,] 28     5.0     4.8     3.9     5.5     3.8
## [29,] 29     5.0     3.4     4.9     4.6     5.4
## [30,] 30     5.2     4.1     3.9     5.5     2.8
```

```r
## Now, we can see the completely arranged data with the first column : ID,
## and other remaining columns meaning the each operator, row : obs.
## Naturally, each cell data means (except for first column data) sensory data corresponding to each op

## Tabulate the arranged data set
Table.Arranged.data <- as.table(Arranged.data)

## Show the tabulated arranged data set.
Table.Arranged.data
```

```
##      ID 1st op. 2nd op. 3rd op. 4th op. 5th op.
## A   1.0     4.3     4.9     3.3     5.3     4.4
## B   2.0     4.5     4.0     5.5     3.3     4.3
## C   3.0     5.3     3.4     5.7     4.7     4.1
## D   4.0     6.0     5.3     4.5     5.9     4.7
## E   5.0     6.3     4.2     5.5     4.9     4.9
## F   6.0     5.9     4.7     6.3     4.6     6.0
## G   7.0     2.4     2.5     2.3     3.1     2.4
## H   8.0     3.0     2.8     2.7     1.3     3.9
## I   9.0     3.9     2.6     4.6     2.2     1.9
## J  10.0     7.4     8.2     6.4     6.8     6.0
## K  11.0     7.9     5.9     7.3     6.1     7.1
## L  12.0     7.1     6.9     7.0     6.7     6.4
## M  13.0     5.7     6.3     5.4     6.1     5.9
## N  14.0     5.7     5.4     6.2     6.5     5.8
## O  15.0     6.0     6.1     7.0     4.9     5.8
## P  16.0     2.2     2.4     1.7     3.4     1.7
## Q  17.0     1.8     2.1     4.0     1.7     3.0
## R  18.0     3.3     1.1     3.3     2.1     2.1
## S  19.0     1.2     1.5     1.2     0.9     0.7
## T  20.0     2.4     0.8     1.2     1.3     1.3
## U  21.0     3.1     1.1     1.9     1.6     0.9
```

```
## V  22.0     4.2     4.8     4.5     4.6     3.2
## W  23.0     4.5     4.7     4.9     4.6     3.0
## X  24.0     4.8     4.7     4.8     4.3     4.8
## Y  25.0     8.0     8.6     9.0     9.4     8.8
## Z  26.0     7.7     6.7     9.0     7.9     9.0
## A1 27.0     9.2     8.1     9.1     7.6     8.9
## B1 28.0     5.0     4.8     3.9     5.5     3.8
## C1 29.0     5.0     3.4     4.9     4.6     5.4
## D1 30.0     5.2     4.1     3.9     5.5     2.8
```

```
## Summary table for arranged.data for each variable.

summary(Arranged.data[,2]) # summary of sensory data from 1st operator
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   3.450   5.000   4.967   6.000   9.200
```

```
summary(Arranged.data[,3]) # summary of sensory data from 2nd operator
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.800   2.650   4.700   4.403   5.775   8.600
```

```
summary(Arranged.data[,4]) # summary of sensory data from 3rd operator
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   3.450   4.850   4.900   6.275   9.100
```

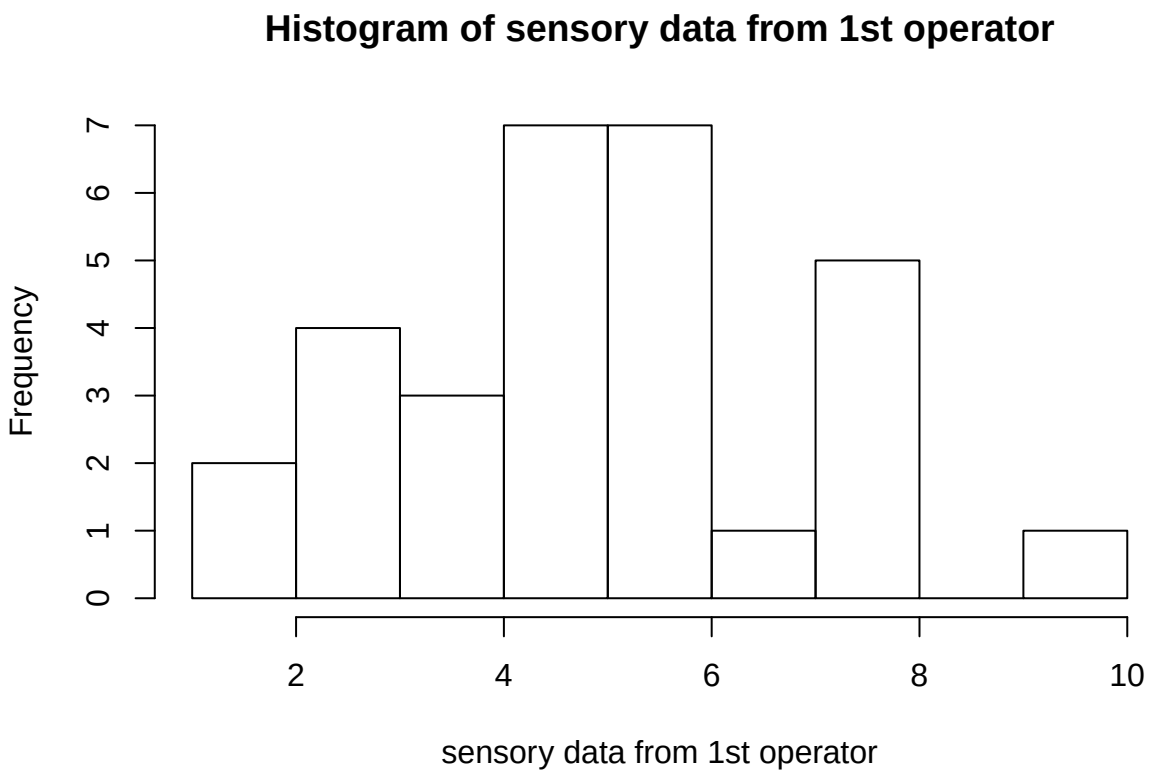
```
summary(Arranged.data[,5]) # summary of sensory data from 4th operator
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.90    3.15    4.65    4.58    6.05    9.40
```

```
summary(Arranged.data[,6]) # summary of sensory data from 5th operator
```

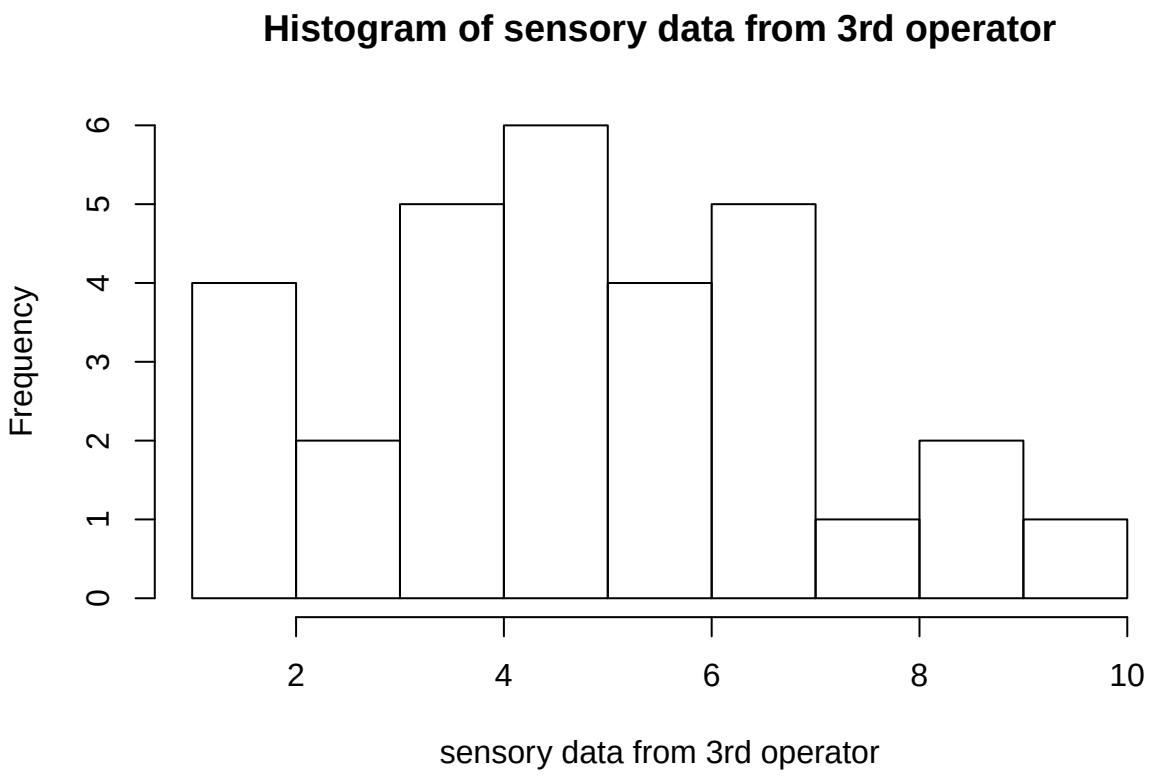
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.700   2.850   4.350   4.433   5.875   9.000
```

```
## Visualization for each variable.

# Histogram.

hist(Arranged.data[,2],main="Histogram of sensory data from 1st operator",xlab="sensory data from 1st o
```

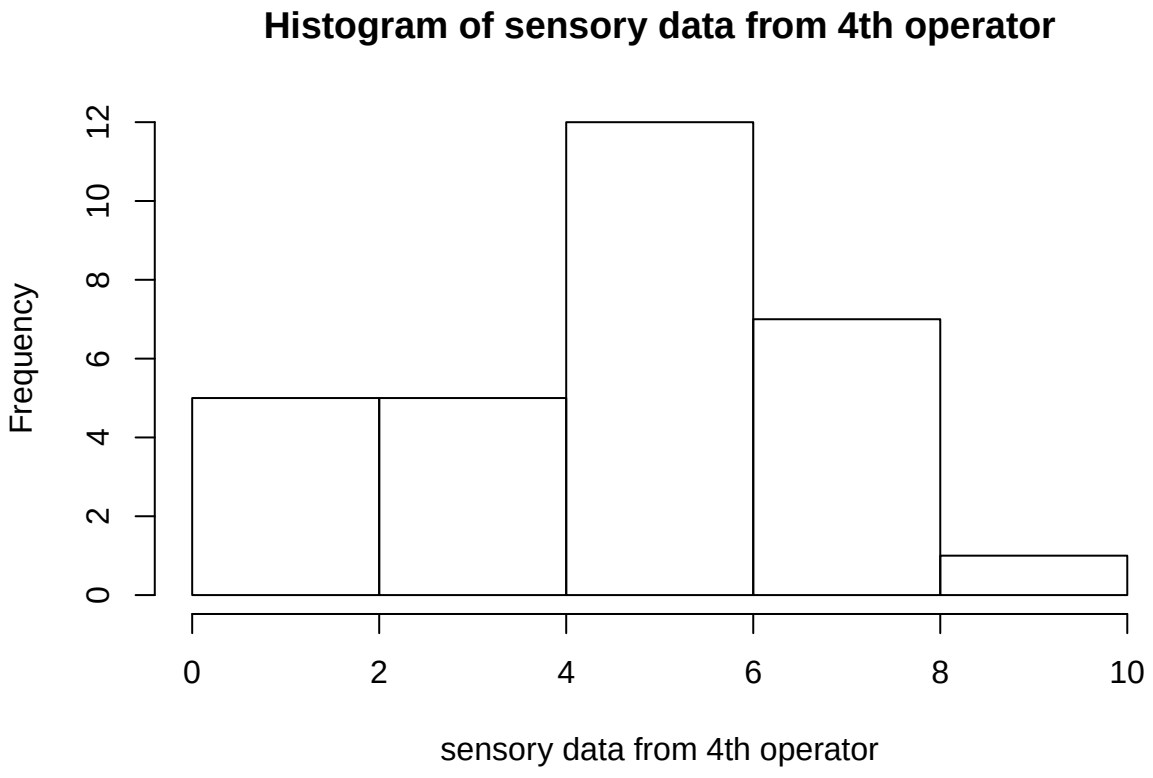



```
hist(Arranged.data[,3],main="Histogram of sensory data from 2nd operator",xlab="sensory data from 2nd op
```

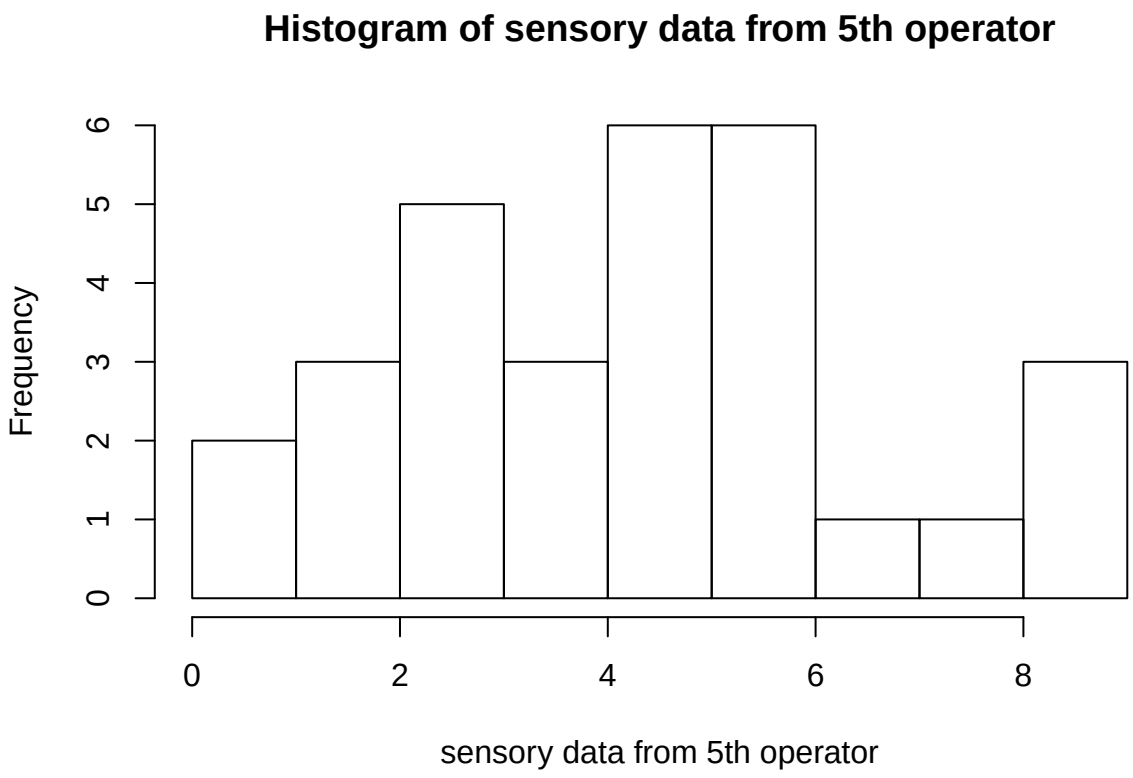

```
hist(Arranged.data[,4],main="Histogram of sensory data from 3rd operator",xlab="sensory data from 3rd op
```



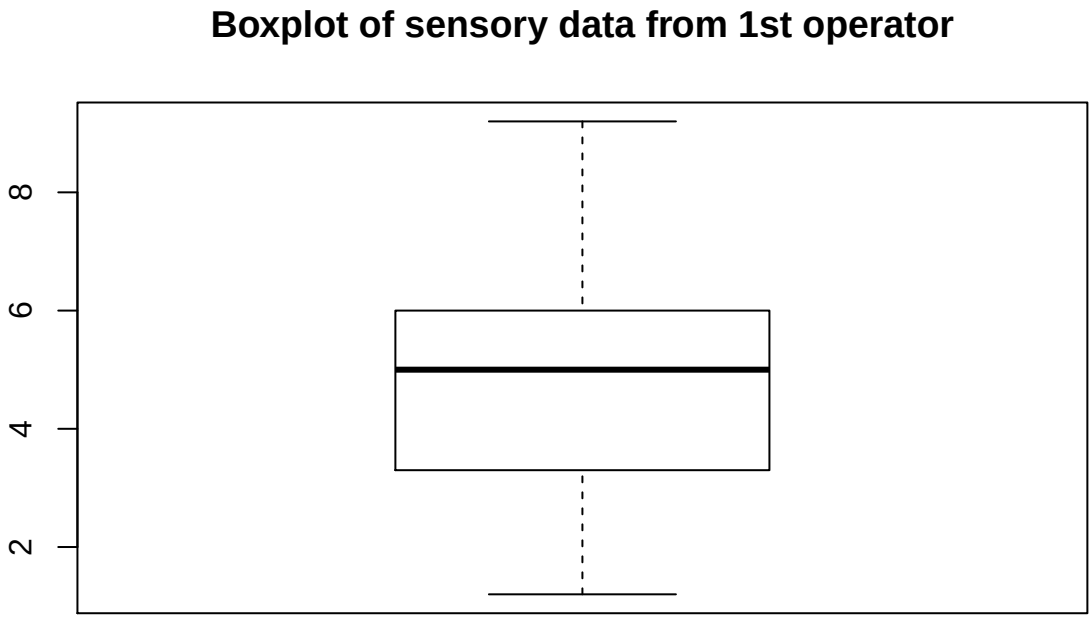

```
hist(Arranged.data[,5],main="Histogram of sensory data from 4th operator",xlab="sensory data from 4th op
```



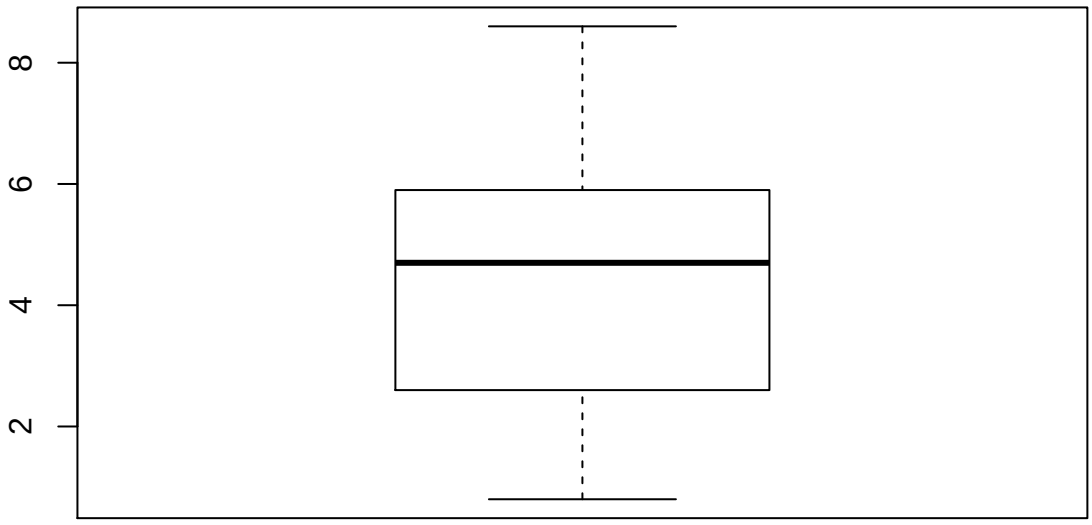

```
hist(Arranged.data[,6],main="Histogram of sensory data from 5th operator",xlab="sensory data from 5th op
```



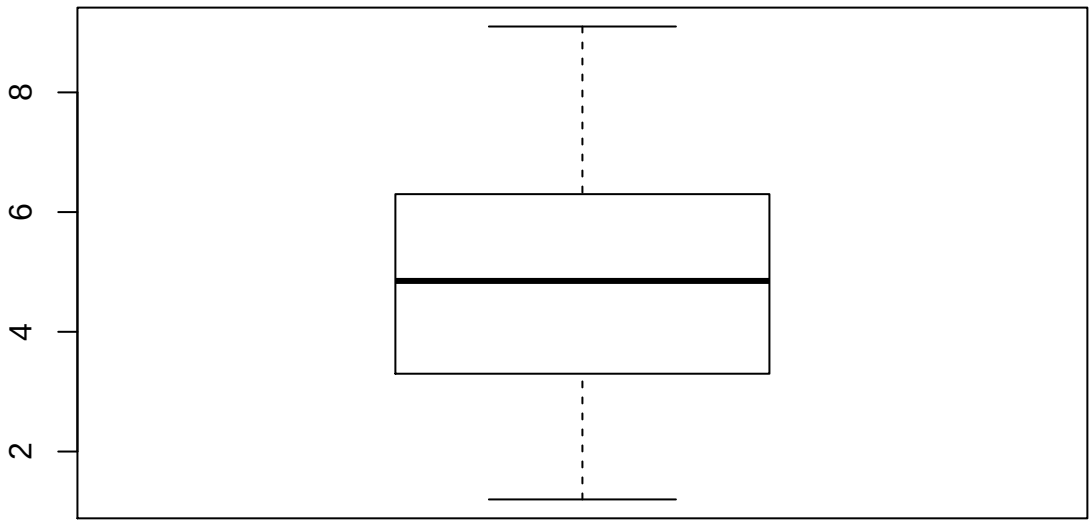

```
## Box plot.

boxplot(Arranged.data[,2],main="Boxplot of sensory data from 1st operator")
```



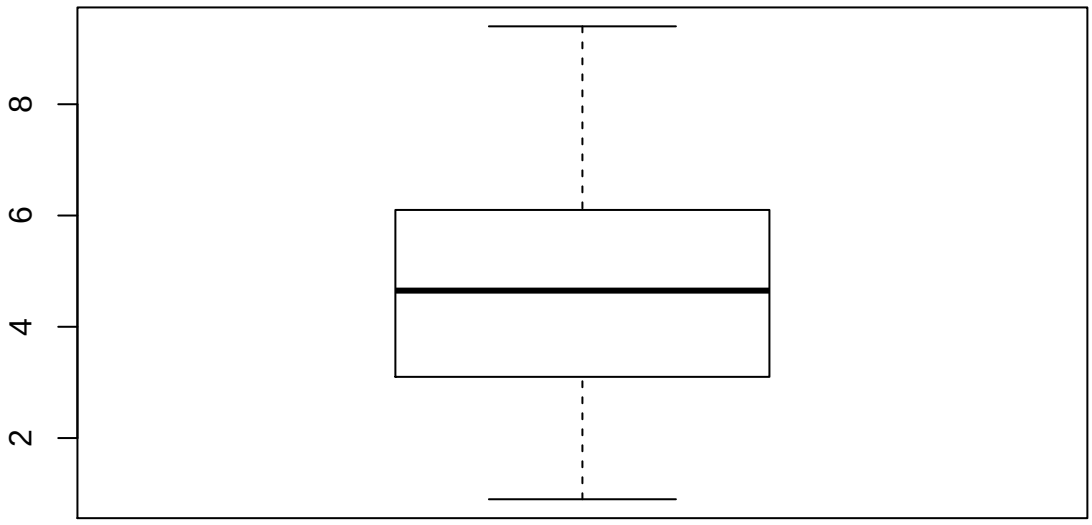

```
boxplot(Arranged.data[,3],main="Boxplot of sensory data from 2nd operator")
```



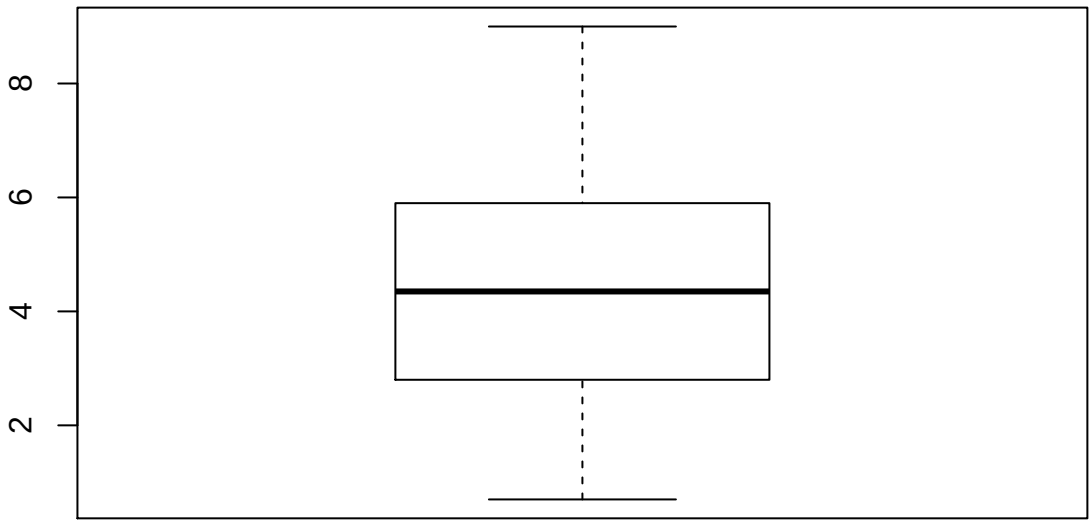

```
boxplot(Arranged.data[,4],main="Boxplot of sensory data from 3rd operator")
```




```
boxplot(Arranged.data[,5],main="Boxplot of sensory data from 4th operator")
```




```
boxplot(Arranged.data[,6],main="Boxplot of sensory data from 5th operator")
```




```
## There seems to be nothing special about each sensory data set obtained from each operator.
```

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.

```
Longjump.data <- read.csv("LongJumpData.csv",header=T,sep=" ",stringsAsFactors = FALSE)
Longjump.data
```

```
##   Year   Long Jump Year.1 Long.1 Jump.1 Year.2 Long.2 Jump.2 Year.3 Long.3
## 1   -4 249.75   24 293.13     56 308.25     80 336.25     NA     NA     NA
## 2    0 282.88   28 304.75     60 319.75     84 336.25     NA     NA     NA
## 3    4 289.00   32 300.75     64 317.75     88 343.25     NA     NA     NA
```

```
## 4    8 294.50   36 317.31     68 350.50     92 342.50     NA     NA     NA
## 5   12 299.25   48 308.00     72 324.50     NA     NA     NA     NA     NA
## 6   20 281.50   52 298.00     76 328.50     NA     NA     NA     NA     NA
##   Jump.3
## 1     NA
## 2     NA
## 3     NA
## 4     NA
## 5     NA
## 6     NA
```

```
nrow(Longjump.data) # 6
```

```
## [1] 6
```

```
ncol(Longjump.data) # 12
```

```
## [1] 12
```

```
## We need to transform the given data set with the # row : 22 (# of obs), and # col : 2 (Year (1900=0

## First, we seperate the data by variables by Year, Long jump


## By concatenating the 1st, 3rd, 5th, and 7th column data, we can get the vector of year data.
year.expected.data <- c(Longjump.data[,1],Longjump.data[,3],Longjump.data[,5],Longjump.data[,7])

length(year.expected.data)
```

```
## [1] 24
```

```
## By concatenating the 2nd, 4th, 6th,and 8th column data, we can get the vector of Long jump data.
Long.jump.expected.data <- c(Longjump.data[,2],Longjump.data[,4],Longjump.data[,6],Longjump.data[,8])

length(Long.jump.expected.data)
```

```
## [1] 24
```

```
## Make arranged data with matrix form (column : variable, row : observation)
Arranged.data2 <- matrix(c(year.expected.data,Long.jump.expected.data),nrow=24,ncol=2,byrow=FALSE)

Arranged.data2
```

```
##       [,1]   [,2]
##  [1,]   -4 249.75
##  [2,]    0 282.88
##  [3,]    4 289.00
##  [4,]    8 294.50
##  [5,]   12 299.25
##  [6,]   20 281.50
##  [7,]   24 293.13
##  [8,]   28 304.75
##  [9,]   32 300.75
## [10,]   36 317.31
## [11,]   48 308.00
## [12,]   52 298.00
## [13,]   56 308.25
## [14,]   60 319.75
```

```
## [15,]   64 317.75
## [16,]   68 350.50
## [17,]   72 324.50
## [18,]   76 328.50
## [19,]   80 336.25
## [20,]   84 336.25
## [21,]   88 343.25
## [22,]   92 342.50
## [23,]   NA     NA
## [24,]   NA     NA
```

```
## We arrange the given data, roughly, but we have to change the first column data into 'day'form.
## At first, we need to eliminate the last two row data because they are NAs which are useless.
Arranged.data2 <- Arranged.data2[-c((nrow(Arranged.data2)-1):nrow(Arranged.data2)),]

Arranged.data2
```

```
##       [,1]   [,2]
##  [1,]   -4 249.75
##  [2,]    0 282.88
##  [3,]    4 289.00
##  [4,]    8 294.50
##  [5,]   12 299.25
##  [6,]   20 281.50
##  [7,]   24 293.13
##  [8,]   28 304.75
##  [9,]   32 300.75
## [10,]   36 317.31
## [11,]   48 308.00
## [12,]   52 298.00
## [13,]   56 308.25
## [14,]   60 319.75
## [15,]   64 317.75
## [16,]   68 350.50
## [17,]   72 324.50
## [18,]   76 328.50
## [19,]   80 336.25
## [20,]   84 336.25
## [21,]   88 343.25
## [22,]   92 342.50
```

```
## Procedures to change the 1st column data to day data.
edates <- Arranged.data2[,1]

edates
```

```
##  [1] -4  0  4  8 12 20 24 28 32 36 48 52 56 60 64 68 72 76 80 84 88 92
```

```
edates[edates>=60] <- edates[edates>=60]
edates <- as.Date(edates, origin="1900-01-01")

edates
```

```
##  [1] "1899-12-28" "1900-01-01" "1900-01-05" "1900-01-09" "1900-01-13"
##  [6] "1900-01-21" "1900-01-25" "1900-01-29" "1900-02-02" "1900-02-06"
## [11] "1900-02-18" "1900-02-22" "1900-02-26" "1900-03-02" "1900-03-06"
## [16] "1900-03-10" "1900-03-14" "1900-03-18" "1900-03-22" "1900-03-26"
```

```
## [21] "1900-03-30" "1900-04-03"
```

```
## Success !!

## Eliminate NA data in second column (Long jump data)data using the below code.
Long.jump.expected.data <- Long.jump.expected.data[-c((length(Long.jump.expected.data)-1):length(Long.ju

Long.jump.expected.data
```

```
##  [1] 249.75 282.88 289.00 294.50 299.25 281.50 293.13 304.75 300.75 317.31
## [11] 308.00 298.00 308.25 319.75 317.75 350.50 324.50 328.50 336.25 336.25
## [21] 343.25 342.50
```

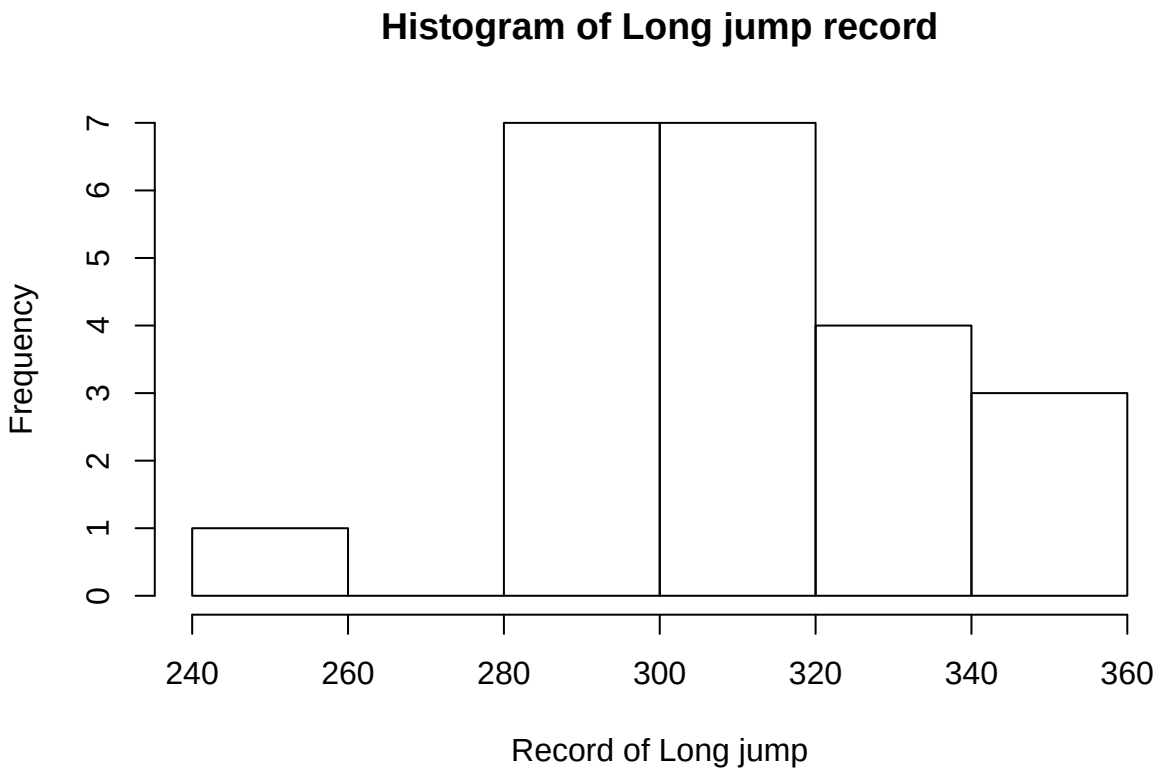
```
## Now, we have to recombine this day data and Longjump data into matrix form.

Arranged.data2 <- data.frame(
   Id = c (1:22),
   Date=edates,

   Long.jump.record = Long.jump.expected.data,

   stringsAsFactors = FALSE
)

Arranged.data2
```

```
##    Id       Date Long.jump.record
## 1   1 1899-12-28           249.75
## 2   2 1900-01-01           282.88
## 3   3 1900-01-05           289.00
## 4   4 1900-01-09           294.50
## 5   5 1900-01-13           299.25
## 6   6 1900-01-21           281.50
## 7   7 1900-01-25           293.13
## 8   8 1900-01-29           304.75
## 9   9 1900-02-02           300.75
## 10 10 1900-02-06           317.31
## 11 11 1900-02-18           308.00
## 12 12 1900-02-22           298.00
## 13 13 1900-02-26           308.25
## 14 14 1900-03-02           319.75
## 15 15 1900-03-06           317.75
## 16 16 1900-03-10           350.50
## 17 17 1900-03-14           324.50
## 18 18 1900-03-18           328.50
## 19 19 1900-03-22           336.25
## 20 20 1900-03-26           336.25
## 21 21 1900-03-30           343.25
## 22 22 1900-04-03           342.50
```

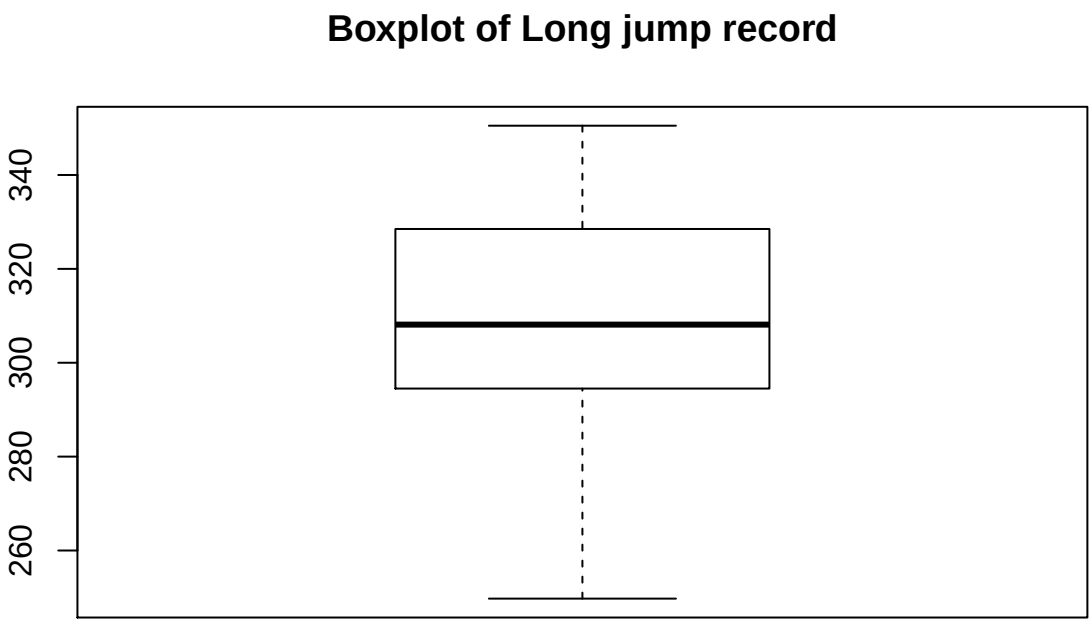
```
## Summary and Visualization

summary(Arranged.data2$Long.jump.record)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   249.8   295.4   308.1   310.3   327.5   350.5
```

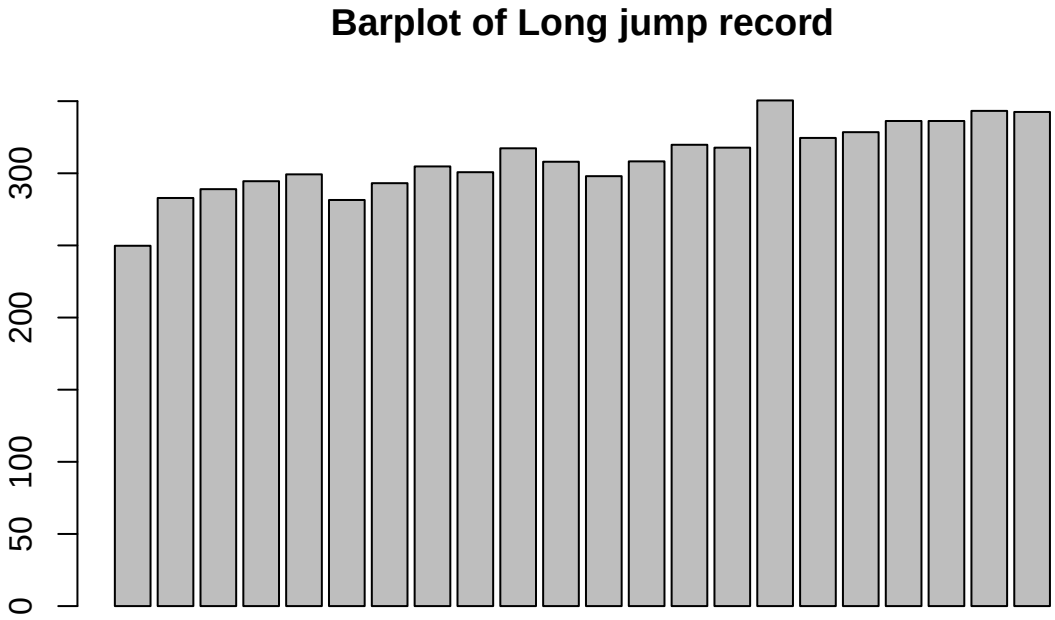
```r
hist(Arranged.data2$Long.jump.record,main="Histogram of Long jump record",xlab = "Record of Long jump")
```



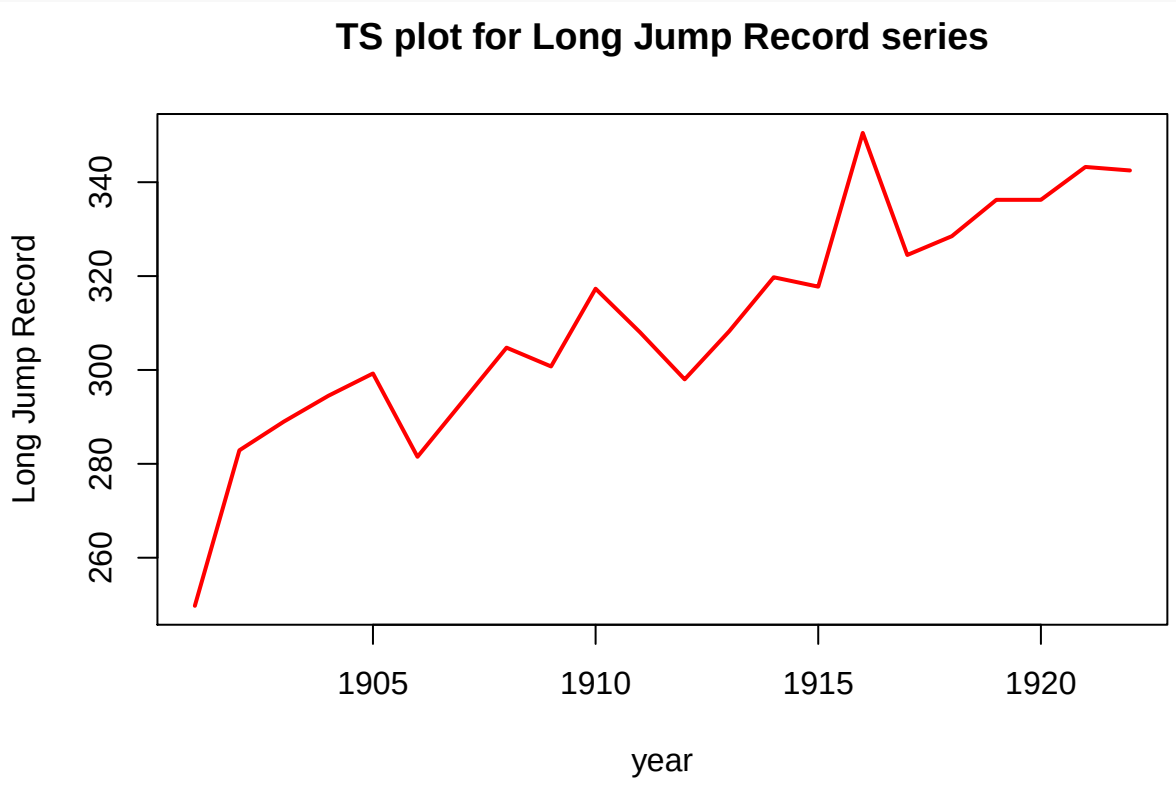

```r
boxplot(Arranged.data2$Long.jump.record,main="Boxplot of Long jump record")
```




```r
barplot(Arranged.data2$Long.jump.record,main="Barplot of Long jump record")
```




```
## Time series plot

Arranged.data2.ts <- ts(Arranged.data2[,-1])

Arranged.data2.ts <- ts(Arranged.data2$Long.jump.record, frequency = 1, start=1901)

plot.ts(Arranged.data2.ts,xlab="year",ylab="Long Jump Record",main="TS plot for Long Jump Record series")
```



c. Brain,body, and weight data

```
Brain.Body.weight.data <- read.csv("BrainandBodyWeight.csv",header=T,sep=" ",stringsAsFactors = FALSE)
Brain.Body.weight.data
```

```
##         Body     Wt    Brain   Wt.1  Body.1   Wt.2 Brain.1 Wt.3 Body.2 Wt.4
## 1      3.385   44.5  521.000  655.0   2.500  12.10      NA    NA      NA    NA
## 2      0.480   15.5    0.785    3.5  55.500 175.00      NA    NA      NA    NA
## 3      1.350    8.1   10.000  115.0 100.000 157.00      NA    NA      NA    NA
## 4    465.000  423.0    3.300   25.6  52.160 440.00      NA    NA      NA    NA
## 5     36.330  119.5    0.200    5.0  10.550 179.50      NA    NA      NA    NA
## 6     27.660  115.0    1.410   17.5   0.550   2.40      NA    NA      NA    NA
## 7     14.830   98.2  529.000  680.0  60.000  81.00      NA    NA      NA    NA
## 8      1.040    5.5  207.000  406.0   3.600  21.00      NA    NA      NA    NA
## 9      4.190   58.0   85.000  325.0   4.288  39.20      NA    NA      NA    NA
## 10     0.425    6.4    0.750   12.3   0.280   1.90      NA    NA      NA    NA
## 11     0.101    4.0   62.000 1320.0   0.075   1.20      NA    NA      NA    NA
## 12     0.920    5.7 6654.000 5712.0   0.122   3.00      NA    NA      NA    NA
## 13     1.000    6.6    3.500    3.9   0.048   0.33      NA    NA      NA    NA
## 14     0.005    0.1    6.800  179.0 192.000 180.00      NA    NA      NA    NA
## 15     0.060    1.0   35.000   56.0   3.000  25.00      NA    NA      NA    NA
## 16     3.500   10.8    4.050   17.0 160.000 169.00      NA    NA      NA    NA
## 17     2.000   12.3    0.120    1.0   0.900   2.60      NA    NA      NA    NA
## 18     1.700    6.3    0.023    0.4   1.620  11.40      NA    NA      NA    NA
## 19  2547.000 4603.0    0.010    0.3   0.104   2.50      NA    NA      NA    NA
## 20     0.023    0.3    1.400   12.5   4.235  50.40      NA    NA      NA    NA
## 21   187.100  419.0  250.000  490.0      NA     NA      NA    NA      NA    NA
##    Brain.2 Wt.5
## 1       NA   NA
## 2       NA   NA
## 3       NA   NA
## 4       NA   NA
## 5       NA   NA
## 6       NA   NA
## 7       NA   NA
## 8       NA   NA
## 9       NA   NA
## 10      NA   NA
## 11      NA   NA
## 12      NA   NA
## 13      NA   NA
## 14      NA   NA
## 15      NA   NA
## 16      NA   NA
## 17      NA   NA
## 18      NA   NA
## 19      NA   NA
## 20      NA   NA
## 21      NA   NA
```

```
ncol(Brain.Body.weight.data) # 12
```

```
## [1] 12
```

```r
nrow(Brain.Body.weight.data) # 21
```

```
## [1] 21
```

```r
## We need to transform the given data set with the # row : 62, and # col : 2 (Brain Weight (g), Body W
## First, we seperate the data by variables by Brain weight, Body weight


## By concatenating the 3rd, 4th, and 6th column data, we can get the vector of brain weight data.
Brain.Weight.expected.data <- c(Brain.Body.weight.data[,3],Brain.Body.weight.data[,4],Brain.Body.weight

## By concatenating the 1st, 2nd, and 5th column data, we can get the vector of Body weight data.
Body.Weight.expected.data <- c(Brain.Body.weight.data[,1],Brain.Body.weight.data[,2],Brain.Body.weight.

## Make arranged data with matrix form (column : variable, row : observation)
Arranged.data3 <- matrix(c(Brain.Weight.expected.data,Body.Weight.expected.data),nrow=63,ncol=2,byrow=F


## Set the column names and row number
colnames(Arranged.data3) <- c("Brain.Weight(g)","Body.Weight(kg)")

rownames(Arranged.data3) <- c(1:63)


# Arranged.data2

Table.Arranged.data3 <- as.table(Arranged.data3)

## Show the arranged data set.
Table.Arranged.data3
```

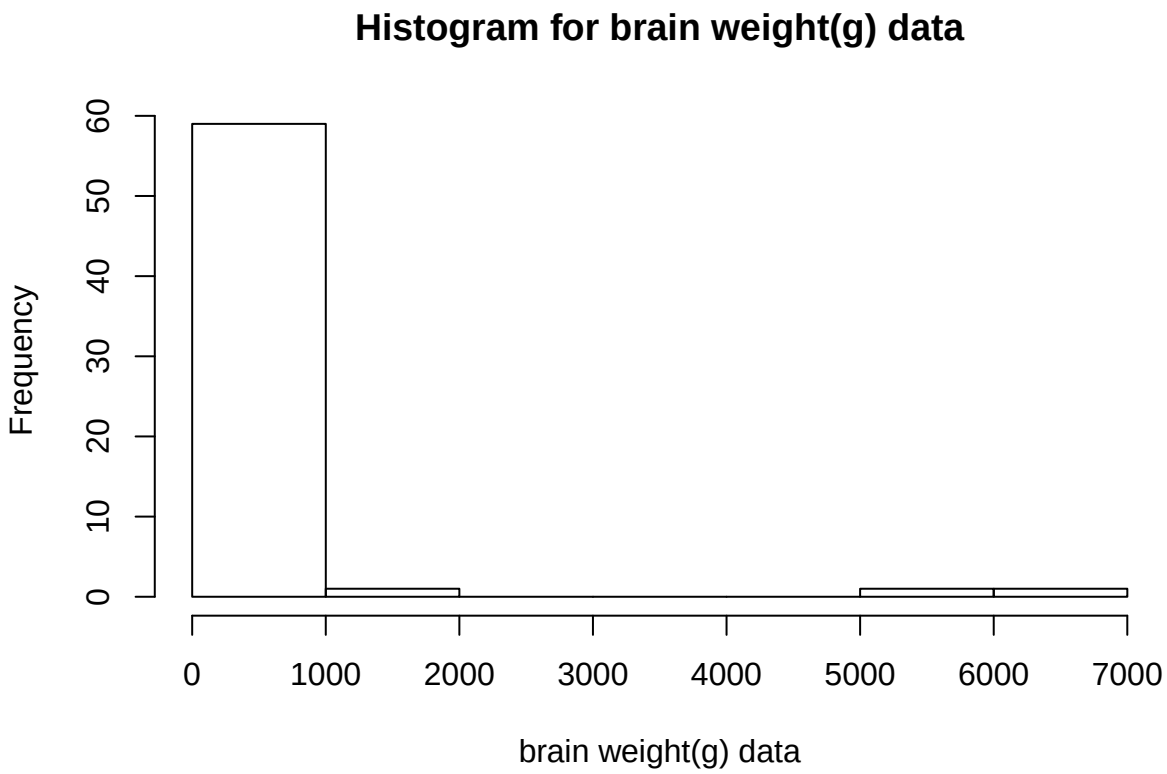
```
##    Brain.Weight(g) Body.Weight(kg)
## 1          521.000           3.385
## 2            0.785           0.480
## 3           10.000           1.350
## 4            3.300         465.000
## 5            0.200          36.330
## 6            1.410          27.660
## 7          529.000          14.830
## 8          207.000           1.040
## 9           85.000           4.190
## 10           0.750           0.425
## 11          62.000           0.101
## 12        6654.000           0.920
## 13           3.500           1.000
## 14           6.800           0.005
## 15          35.000           0.060
## 16           4.050           3.500
## 17           0.120           2.000
## 18           0.023           1.700
## 19           0.010        2547.000
## 20           1.400           0.023
## 21         250.000         187.100
```

```
## 22         655.000          44.500
## 23           3.500          15.500
## 24         115.000           8.100
## 25          25.600         423.000
## 26           5.000         119.500
## 27          17.500         115.000
## 28         680.000          98.200
## 29         406.000           5.500
## 30         325.000          58.000
## 31          12.300           6.400
## 32        1320.000           4.000
## 33        5712.000           5.700
## 34           3.900           6.600
## 35         179.000           0.100
## 36          56.000           1.000
## 37          17.000          10.800
## 38           1.000          12.300
## 39           0.400           6.300
## 40           0.300        4603.000
## 41          12.500           0.300
## 42         490.000         419.000
## 43          12.100           2.500
## 44         175.000          55.500
## 45         157.000         100.000
## 46         440.000          52.160
## 47         179.500          10.550
## 48           2.400           0.550
## 49          81.000          60.000
## 50          21.000           3.600
## 51          39.200           4.288
## 52           1.900           0.280
## 53           1.200           0.075
## 54           3.000           0.122
## 55           0.330           0.048
## 56         180.000         192.000
## 57          25.000           3.000
## 58         169.000         160.000
## 59           2.600           0.900
## 60          11.400           1.620
## 61           2.500           0.104
## 62          50.400           4.235
## 63
```

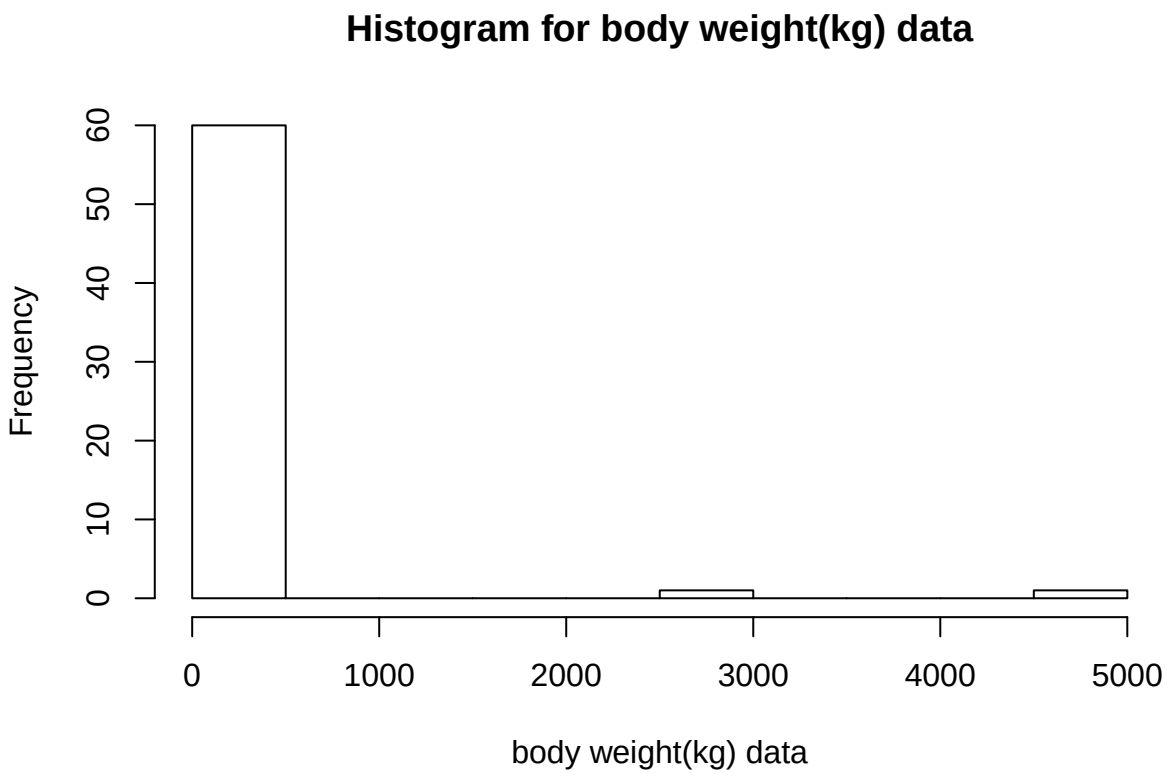
```
## Summary table for arranged.data for each variable.
summary(Arranged.data3)
```

```
##  Brain.Weight(g)    Body.Weight(kg)
##  Min.   :   0.010   Min.   :   0.005
##  1st Qu.:   2.525   1st Qu.:   0.940
##  Median :  17.250   Median :   4.261
##  Mean   : 322.046   Mean   : 159.878
##  3rd Qu.: 178.000   3rd Qu.:  50.245
##  Max.   :6654.000   Max.   :4603.000
##  NA's   :1          NA's   :1
```

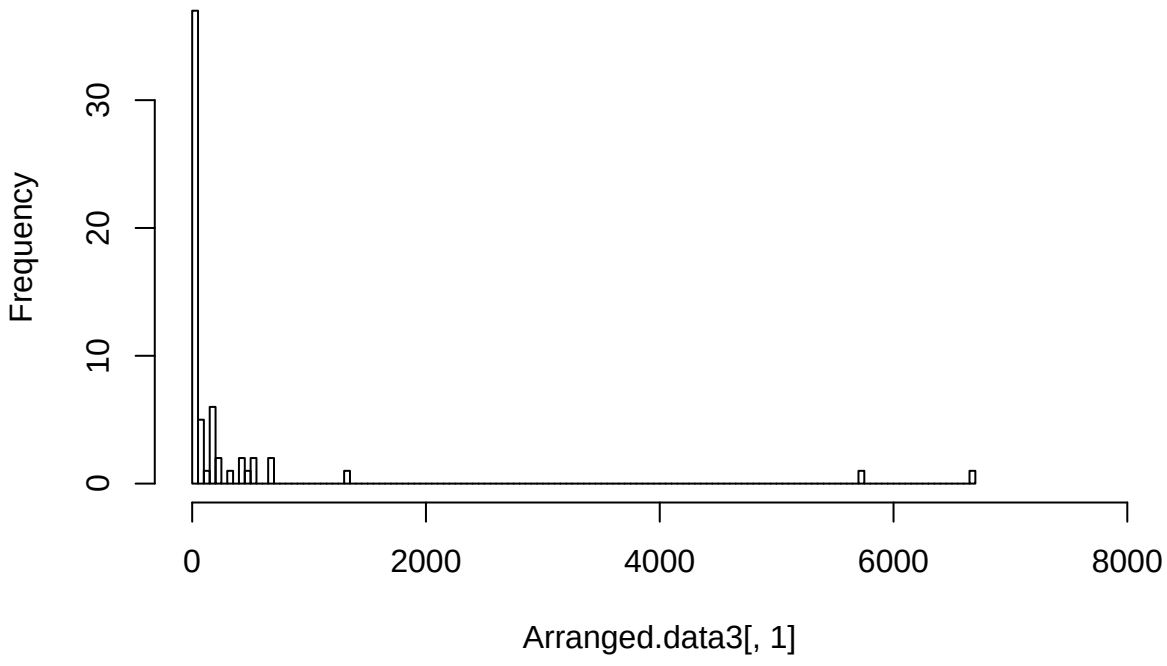
```r
## Visualization for each variable.

# Histogram.

hist(Arranged.data3[,1],main="Histogram for brain weight(g) data",xlab="brain weight(g) data") # for br
```



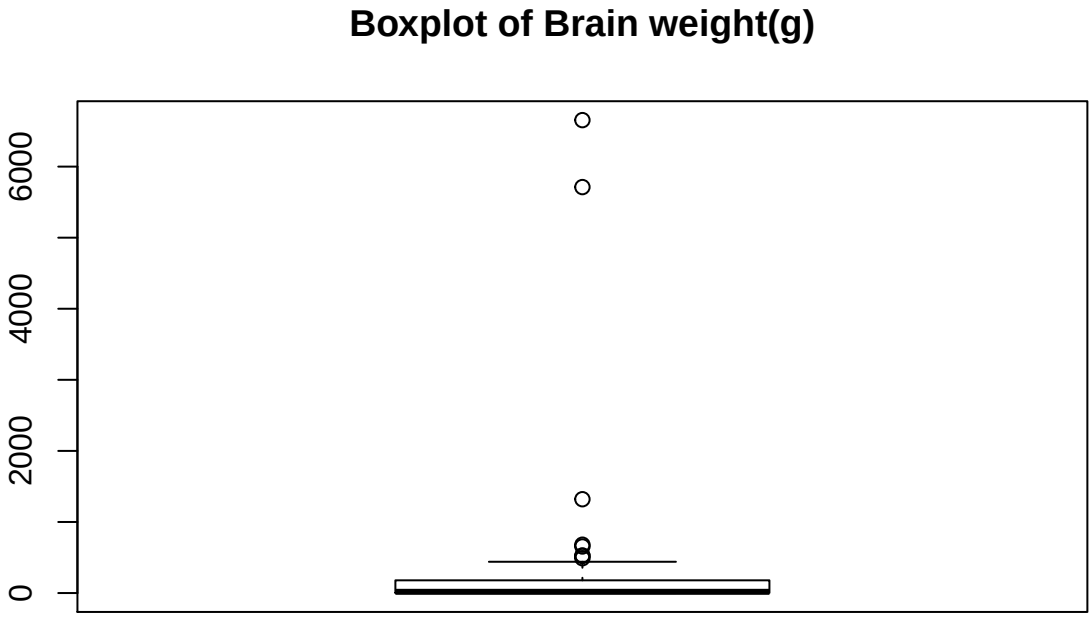

```r
hist(Arranged.data3[,2],main="Histogram for body weight(kg) data",xlab="body weight(kg) data") # for bo
```



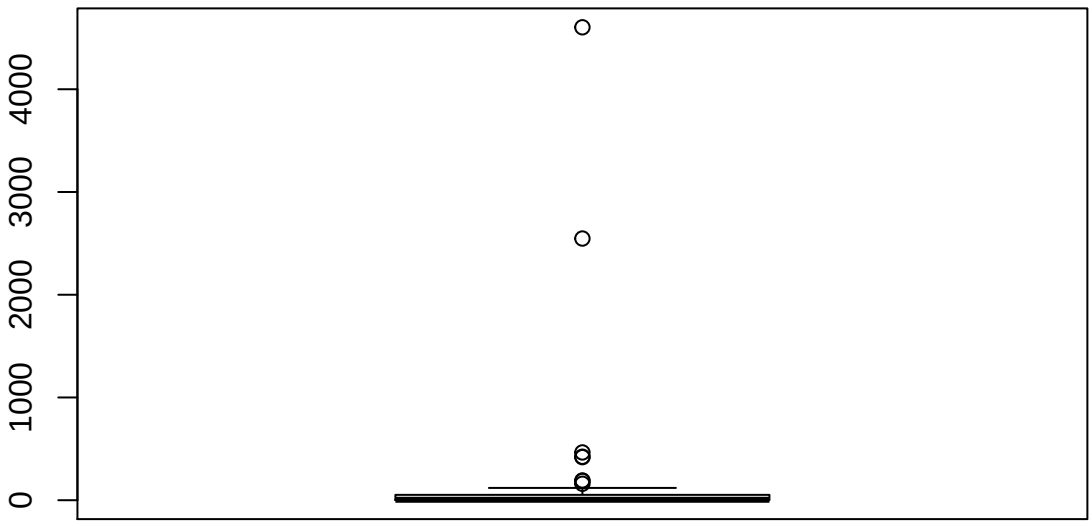

```
## From the each variable's histograms, we can find the fact that most of the observed data
## seems to aggregate in the interval (0,1000)

## C.f) What if taking the 'hist' function with different breaks and xlimit ?
hist(Arranged.data3[,1],breaks=200,xlim=c(0,8000),main="c.f) histogram for figuring out the plurality")
```



```r
## We should be careful of plurality !!! => we shouldn't say the dis. of this data easily.
## because chances are that the shape of the histogram can be seen differently according to the number
## of breaks in 'hist' built-in function.


## Box plot for each variable
boxplot(Arranged.data3[,1],main="Boxplot of Brain weight(g)")# for brain weight
```




```r
boxplot(Arranged.data3[,2],main="Boxplot of Body weight(kg)")# for body weight
```




```r
## There seems to be some outliars within the each variable data sets.
```

d. Triplicate measurements of tomato yield for two varieties of tomatoes at three planting densities.

```
library(stringr)
## In terms of this data set, I will use the 'readLines' Command for reading the raw data set.
Triplicate.meas.tomatoes.at3planting.den.data <- readLines(con="tomato.csv")

## Arrange the raw data set a little bit.
Triplicate.meas.tomatoes.at3planting.den.data <- Triplicate.meas.tomatoes.at3planting.den.data[-1]

Triplicate.meas.tomatoes.at3planting.den.data<-as.matrix(Triplicate.meas.tomatoes.at3planting.den.data,

Triplicate.meas.tomatoes.at3planting.den.data
```

```
##      [,1]
## [1,] "                      10000              20000     30000"
## [2,] "Ife\\#1           16.1,15.3,17.5   16.6,19.2,18.5   20.8,18.0,21.0"
## [3,] "PusaEarlyDwarf   8.1,8.6,10.1,    12.7,13.7,11.5   14.4,15.4,13.7 "
```

```
## We need to handle with the repeated measure.

## The way I choose to deal with repeated measure is assigning the column vector for each kind of tomat

## Let's make the numeric vector of measure of first kind of tomato

tempo.first.tomato.repeated.measure <- Triplicate.meas.tomatoes.at3planting.den.data[2,]

## Split the 'tempo.first.tomato.repeated.measure' data for obtaining the individual data point.

tempo.first.tomato.repeated.measure2 <- str_split(tempo.first.tomato.repeated.measure, ",") # sep : com

tempo.first.tomato.repeated.measure2
```

```
## [[1]]
## [1] "Ife\\#1           16.1" "15.3"
## [3] "17.5   16.6"            "19.2"
## [5] "18.5   20.8"            "18.0"
## [7] "21.0"
```

```
## Even if we have done the above procedure, there seems to need more efforts to get the individual obs


## 2nd, 5th, 8th, and 9th obs. data from first kind of tomato can be received by just following indexin
Second.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure2[[1]][2]

Fifth.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure2[[1]][4]

Eightth.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure2[[1]][6]

Nineth.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure2[[1]][7]

## The thing is that we should break into multi parts what is composed of several things (such as combi
tempo.first.tomato.repeated.measure.first. <- str_split(tempo.first.tomato.repeated.measure2[[1]][1], "

tempo.first.tomato.repeated.measure.first.
```

```
## [[1]]
##  [1] "Ife\\#1" ""        ""        ""        ""        ""        ""
##  [8] ""        ""        ""        ""        "16.1"
```

```
First.data.in.the.first.tomato.repeated.measure <-  tempo.first.tomato.repeated.measure.first.[[1]][12]

First.data.in.the.first.tomato.repeated.measure
```

```
## [1] "16.1"
```

```
tempo.first.tomato.repeated.measure.third. <- str_split(tempo.first.tomato.repeated.measure2[[1]][3], "

tempo.first.tomato.repeated.measure.third.
```

```
## [[1]]
## [1] "17.5" ""     ""     "16.6"
```

```
Third.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure.third.[[1]][1]

Third.data.in.the.first.tomato.repeated.measure
```

```
## [1] "17.5"
```

```
Fourth.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure.third.[[1]][4]

Fourth.data.in.the.first.tomato.repeated.measure
```

```
## [1] "16.6"
```

```
tempo.first.tomato.repeated.measure.fifth. <- str_split(tempo.first.tomato.repeated.measure2[[1]][5], "


tempo.first.tomato.repeated.measure.fifth.
```

```
## [[1]]
## [1] "18.5" ""     ""     "20.8"
```

```
Sixth.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure.fifth.[[1]][1]

Sixth.data.in.the.first.tomato.repeated.measure
```

```
## [1] "18.5"
```

```
Seventh.data.in.the.first.tomato.repeated.measure <- tempo.first.tomato.repeated.measure.fifth.[[1]][4]

Seventh.data.in.the.first.tomato.repeated.measure
```

```
## [1] "20.8"
```

```
# By weaving the above code, we can get the individual obs !!


## Now, we can concatenate repeated measures of first tomato (# : 9)
First.tomato.repeated.measure.vector <- c(First.data.in.the.first.tomato.repeated.measure, Second.data.

First.tomato.repeated.measure.vector <- as.numeric(First.tomato.repeated.measure.vector)


First.tomato.repeated.measure.vector
```

```
## [1] 16.1 15.3 17.5 16.6 19.2 18.5 20.8 18.0 21.0
```

```
## It is the numeric vector composed of obs. of first kind of tomato.


## Do the same procedure to make the numeric vector of measure of Second kind of tomato

tempo.second.tomato.repeated.measure <- Triplicate.meas.tomatoes.at3planting.den.data[3,]


tempo.second.tomato.repeated.measure2 <- str_split(tempo.second.tomato.repeated.measure, ",")

tempo.second.tomato.repeated.measure2
```

```
## [[1]]
## [1] "PusaEarlyDwarf   8.1" "8.6"                 "10.1"
## [4] "    12.7"            "13.7"                "11.5   14.4"
## [7] "15.4"                "13.7 "
```

```
Second.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][2]

Third.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][3]


Fourth.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][4]


Fifth.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][5]

Eightth.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][7]

Nineth.data.in.the.second.tomato.repeated.measure <- tempo.second.tomato.repeated.measure2[[1]][8]



tempo.second.tomato.repeated.measure.first. <- str_split(tempo.second.tomato.repeated.measure2[[1]][1],

tempo.second.tomato.repeated.measure.first.
```

```
## [[1]]
## [1] "PusaEarlyDwarf" ""               ""               "8.1"
```

```
First.data.in.the.second.tomato.repeated.measure <-  tempo.second.tomato.repeated.measure.first.[[1]][4]

First.data.in.the.second.tomato.repeated.measure
```

```
## [1] "8.1"
```

```
tempo.second.repeated.measure.sixth. <- str_split(tempo.second.tomato.repeated.measure2[[1]][6], " ")

tempo.second.repeated.measure.sixth.
```

```
## [[1]]
## [1] "11.5" ""     ""     "14.4"
```

```
Sixth.data.in.the.second.tomato.repeated.measure <- tempo.second.repeated.measure.sixth.[[1]][1]

Sixth.data.in.the.second.tomato.repeated.measure
```

```
## [1] "11.5"
```

```
Seventh.data.in.the.second.tomato.repeated.measure <- tempo.second.repeated.measure.sixth.[[1]][4]

Seventh.data.in.the.second.tomato.repeated.measure
```

```
## [1] "14.4"
```

```
## Now, we can concatenate repeated measures of the second kind of tomato (# : 9)

Second.tomato.repeated.measure.vector <- c(First.data.in.the.second.tomato.repeated.measure, Second.data



Second.tomato.repeated.measure.vector <- as.numeric(Second.tomato.repeated.measure.vector)


Second.tomato.repeated.measure.vector
```

```
## [1]  8.1  8.6 10.1 12.7 13.7 11.5 14.4 15.4 13.7
```

```
## The next thing we should do is the arranging the data set can look comfortable.
## I will use the 'list' form to look this arranged data set nice to viewers.



## Assign each (repeated) data set to the each kind of tomato and each density 'list' space (refer to t


 First.Tomato.First.Density <- new.env()

 First.Tomato.First.Density$first.obs<- First.tomato.repeated.measure.vector[1]

 First.Tomato.First.Density$second.obs<- First.tomato.repeated.measure.vector[2]

 First.Tomato.First.Density$third.obs<- First.tomato.repeated.measure.vector[3]

 First.Tomato.First.Density <- as.list(First.Tomato.First.Density)


 First.Tomato.Second.Density <- new.env()

 First.Tomato.Second.Density$first.obs<- First.tomato.repeated.measure.vector[4]

 First.Tomato.Second.Density$second.obs<- First.tomato.repeated.measure.vector[5]

 First.Tomato.Second.Density$third.obs<- First.tomato.repeated.measure.vector[6]

 First.Tomato.Second.Density <- as.list(First.Tomato.Second.Density)
```

```
First.Tomato.Third.Density <- new.env()

First.Tomato.Third.Density$first.obs<- First.tomato.repeated.measure.vector[7]

First.Tomato.Third.Density$second.obs<- First.tomato.repeated.measure.vector[8]

First.Tomato.Third.Density$third.obs<- First.tomato.repeated.measure.vector[9]

First.Tomato.Third.Density <- as.list(First.Tomato.Third.Density)




 Second.Tomato.First.Density <- new.env()

Second.Tomato.First.Density$first.obs<- Second.tomato.repeated.measure.vector[1]

Second.Tomato.First.Density$second.obs<- Second.tomato.repeated.measure.vector[2]

Second.Tomato.First.Density$third.obs<- Second.tomato.repeated.measure.vector[3]

Second.Tomato.First.Density <- as.list(Second.Tomato.First.Density)



Second.Tomato.Second.Density <- new.env()

Second.Tomato.Second.Density$first.obs<- Second.tomato.repeated.measure.vector[4]

Second.Tomato.Second.Density$second.obs<- Second.tomato.repeated.measure.vector[5]

Second.Tomato.Second.Density$third.obs<- Second.tomato.repeated.measure.vector[6]

Second.Tomato.Second.Density <- as.list(Second.Tomato.Second.Density)



Second.Tomato.Third.Density <- new.env()

Second.Tomato.Third.Density$first.obs<- Second.tomato.repeated.measure.vector[7]

Second.Tomato.Third.Density$second.obs<- Second.tomato.repeated.measure.vector[8]

Second.Tomato.Third.Density$third.obs<- Second.tomato.repeated.measure.vector[9]

Second.Tomato.Third.Density <- as.list(Second.Tomato.Third.Density)
```

```r
Arranged.data4 <- list(First.Tomato.First.Density, First.Tomato.Second.Density, First.Tomato.Third.Dens
                  Second.Tomato.First.Density, Second.Tomato.Second.Density, Second.Tomato.Third.Densi



names(Arranged.data4)[[1]] <- "First Tomato obtained from 1st density"

names(Arranged.data4)[[2]] <- "First Tomato obtained from 2nd density"

names(Arranged.data4)[[3]] <- "First Tomato obtained from 3rd density"

names(Arranged.data4)[[4]] <- "Second Tomato obtained from 1st density"

names(Arranged.data4)[[5]] <- "Second Tomato obtained from 2nd density"

names(Arranged.data4)[[6]] <- "Second Tomato obtained from 3rd density"


Arranged.data4
```

```
## $`First Tomato obtained from 1st density`
## $`First Tomato obtained from 1st density`$third.obs
## [1] 17.5
##
## $`First Tomato obtained from 1st density`$first.obs
## [1] 16.1
##
## $`First Tomato obtained from 1st density`$second.obs
## [1] 15.3
##
##
## $`First Tomato obtained from 2nd density`
## $`First Tomato obtained from 2nd density`$third.obs
## [1] 18.5
##
## $`First Tomato obtained from 2nd density`$first.obs
## [1] 16.6
##
## $`First Tomato obtained from 2nd density`$second.obs
## [1] 19.2
##
##
## $`First Tomato obtained from 3rd density`
## $`First Tomato obtained from 3rd density`$third.obs
## [1] 21
##
## $`First Tomato obtained from 3rd density`$first.obs
## [1] 20.8
##
## $`First Tomato obtained from 3rd density`$second.obs
## [1] 18
##
```

```
##
## $`Second Tomato obtained from 1st density`
## $`Second Tomato obtained from 1st density`$third.obs
## [1] 10.1
##
## $`Second Tomato obtained from 1st density`$first.obs
## [1] 8.1
##
## $`Second Tomato obtained from 1st density`$second.obs
## [1] 8.6
##
##
## $`Second Tomato obtained from 2nd density`
## $`Second Tomato obtained from 2nd density`$third.obs
## [1] 11.5
##
## $`Second Tomato obtained from 2nd density`$first.obs
## [1] 12.7
##
## $`Second Tomato obtained from 2nd density`$second.obs
## [1] 13.7
##
##
## $`Second Tomato obtained from 3rd density`
## $`Second Tomato obtained from 3rd density`$third.obs
## [1] 13.7
##
## $`Second Tomato obtained from 3rd density`$first.obs
## [1] 14.4
##
## $`Second Tomato obtained from 3rd density`$second.obs
## [1] 15.4
```

```
 ## The above arranged data set has the form of list which can allow us to seize the structure of the g
```

```
## Summary table for arranged.data for each tomato from each density

summary(as.numeric(Arranged.data4$`First Tomato obtained from 1st density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    15.3    15.7    16.1    16.3    16.8    17.5
```

```
summary(as.numeric(Arranged.data4$`First Tomato obtained from 2nd density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.60   17.55   18.50   18.10   18.85   19.20
```

```
summary(as.numeric(Arranged.data4$`First Tomato obtained from 3rd density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   19.40   20.80   19.93   20.90   21.00
```

```
summary(as.numeric(Arranged.data4$`Second Tomato obtained from 1st density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   8.100   8.350   8.600   8.933   9.350  10.100
```

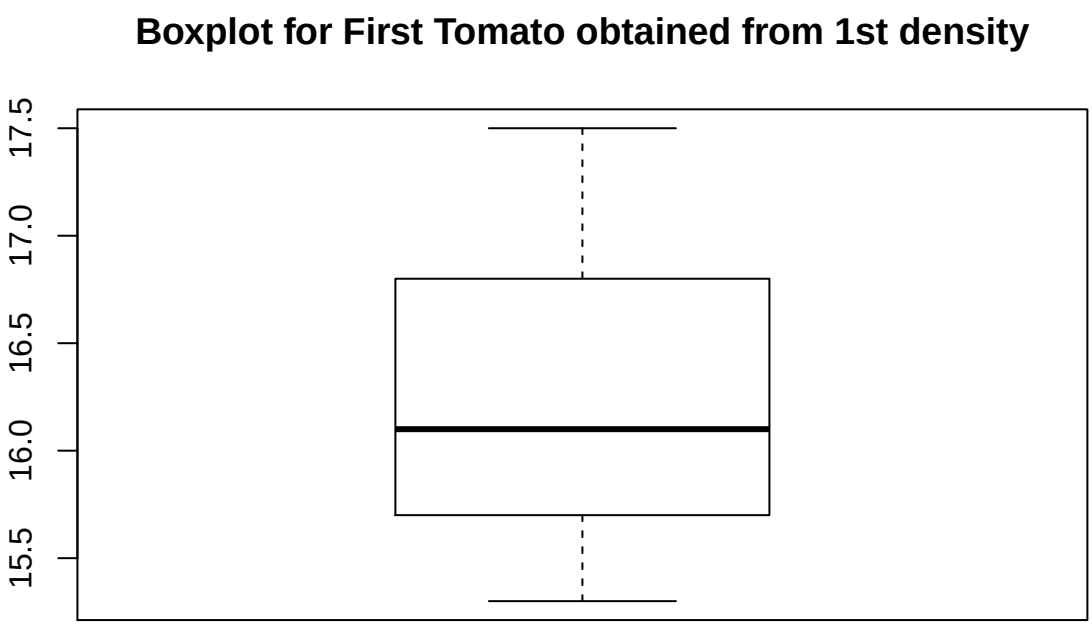
```
summary(as.numeric(Arranged.data4$`Second Tomato obtained from 2nd density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.50   12.10   12.70   12.63   13.20   13.70
```

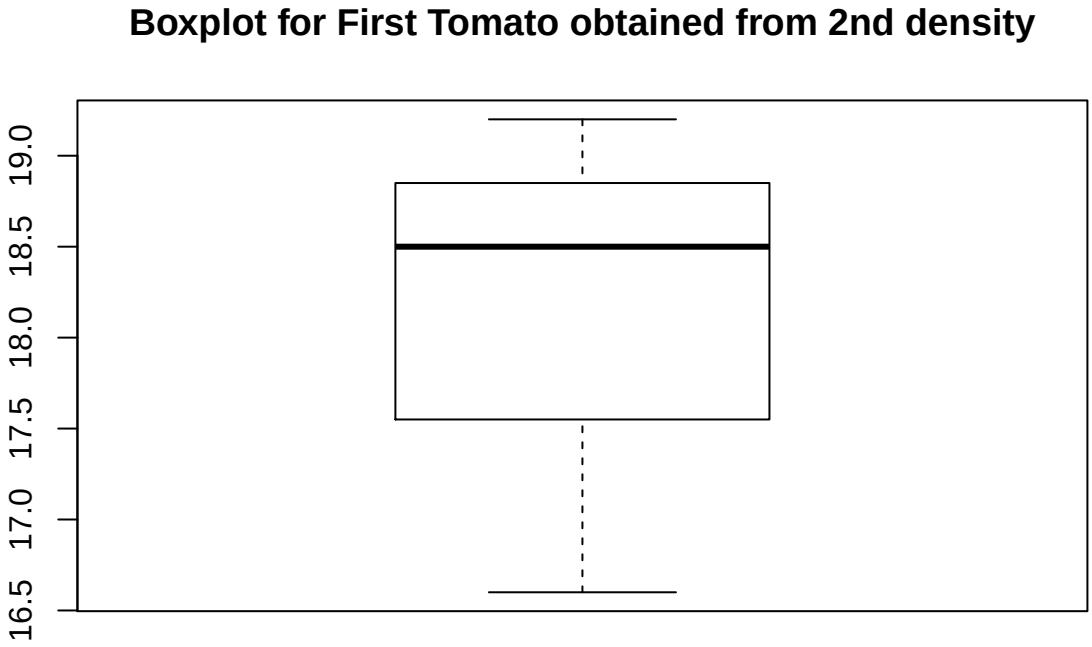
```
summary(as.numeric(Arranged.data4$`Second Tomato obtained from 3rd density`))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.70   14.05   14.40   14.50   14.90   15.40
```

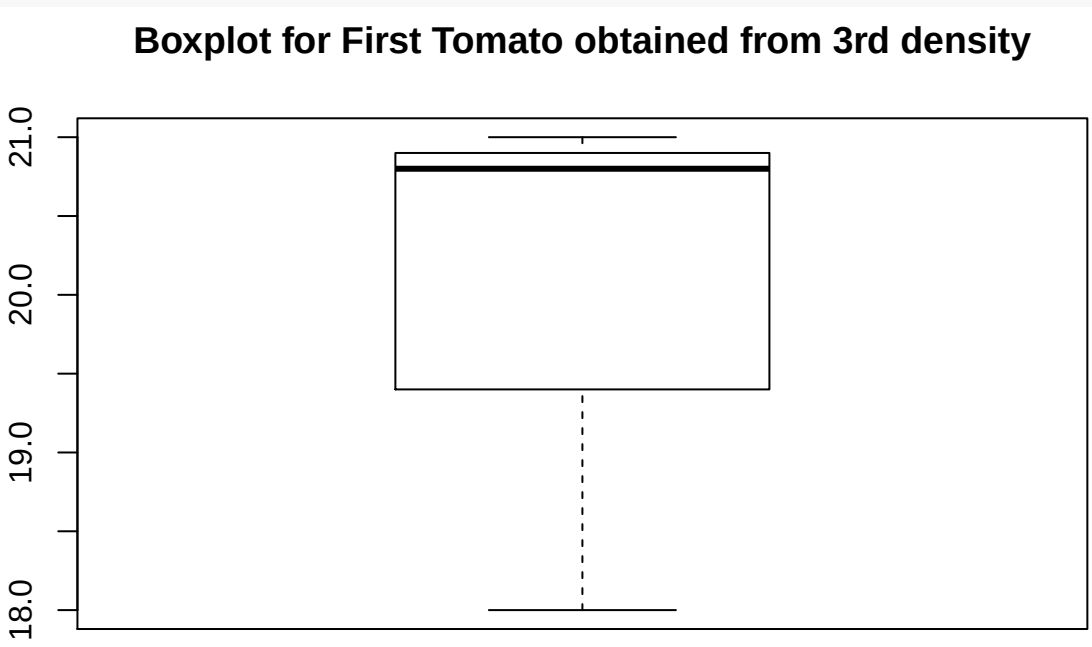
```
## Visualization for each tomato from each density.



## Box plot for each tomato from each density

boxplot(as.numeric(Arranged.data4$`First Tomato obtained from 1st density`),main="Boxplot for First Toma
```



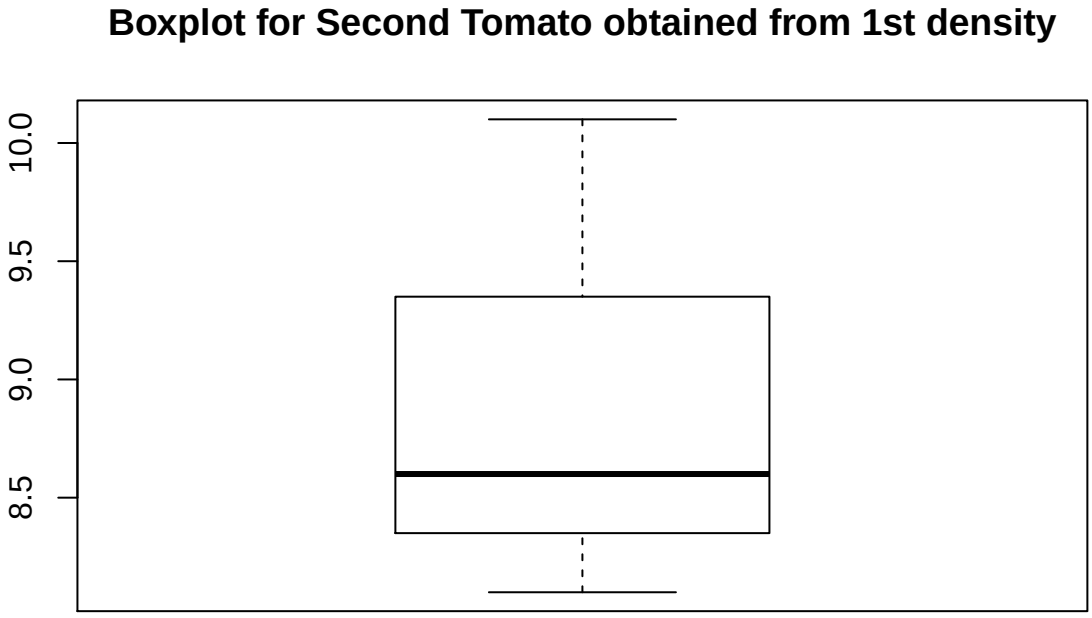

```
boxplot(as.numeric(Arranged.data4$`First Tomato obtained from 2nd density`),main="Boxplot for First Toma
```



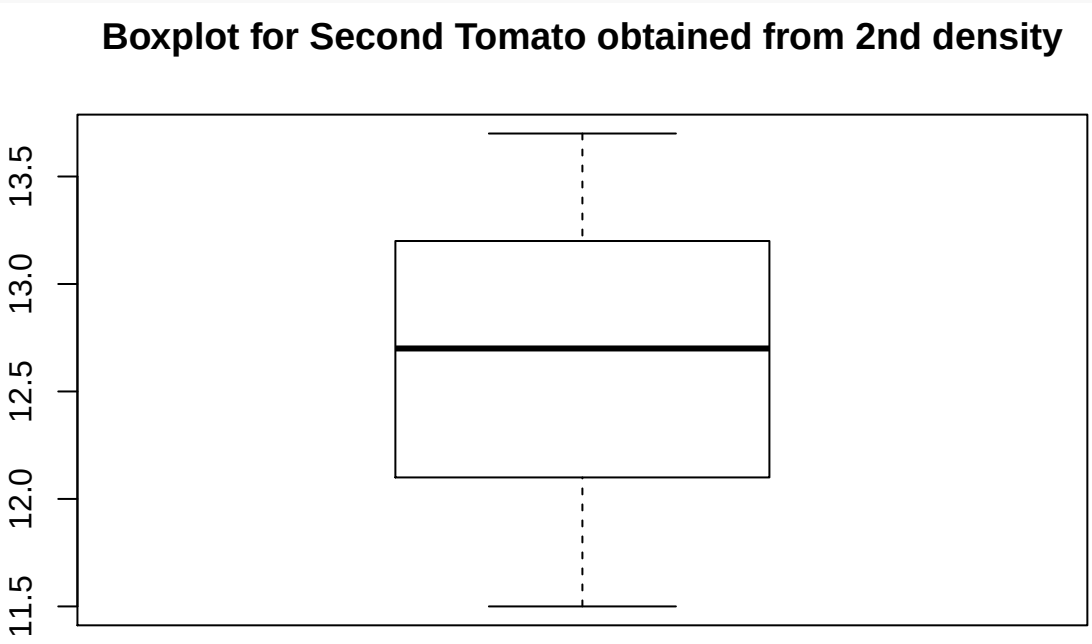

```
boxplot(as.numeric(Arranged.data4$`First Tomato obtained from 3rd density`),main="Boxplot for First Toma
```



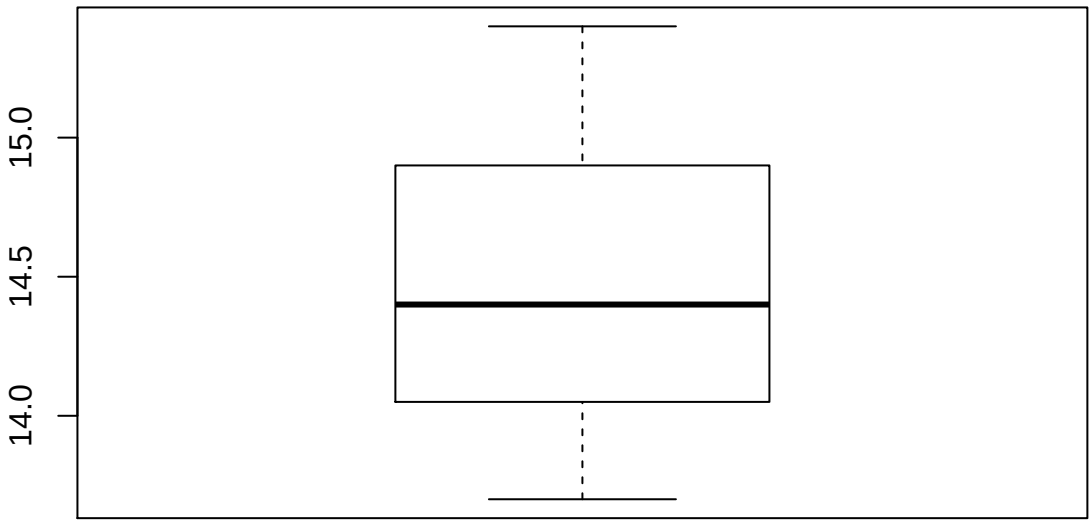

```
boxplot(as.numeric(Arranged.data4$`Second Tomato obtained from 1st density`),main="Boxplot for Second To
```




```
boxplot(as.numeric(Arranged.data4$`Second Tomato obtained from 2nd density`),main="Boxplot for Second To
```




```
boxplot(as.numeric(Arranged.data4$`Second Tomato obtained from 3rd density`),main="Boxplot for Second To
```




```
## There doensn't seem to be no outliar or weird points when we see the a variety of plots.
```

Problem 5

In the swirl lessons, you played with a dataset "plants". Our ultimate goal is to see if there is a relationship between pH and Foliage_Color. Consider a statistic that combines the information in pH_Min and pH_Max.

Clean, summarize and transform the data as appropriate. Use function 'lm' to test for a relationship. Report both the coefficients and ANOVA results in table form.

Note that if you didn't just do the swirl lesson, it is now not available. Add the following code to your project to retrieve it.

```
library(swirl)
```

```
##
## | Hi! I see that you have some variables saved in your workspace. To keep
## | things running smoothly, I recommend you clean up before starting swirl.
##
## | Type ls() to see a list of the variables in your workspace. Then, type
## | rm(list=ls()) to clear your workspace.
##
## | Type swirl() when you are ready to begin.
```

```
# Path to data
.datapath <- file.path(path.package('swirl'), 'Courses',
'R_Programming_E', 'Looking_at_Data','plant-data.txt')

# Read in data
plants <- read.csv(.datapath, strip.white=TRUE, na.strings="")

# Remove annoying columns
.cols2rm <- c('Accepted.Symbol', 'Synonym.Symbol')
plants <- plants[, !(names(plants) %in% .cols2rm)]
```

```r
# Make names pretty
names(plants) <- c('Scientific_Name', 'Duration', 'Active_Growth_Period',
'Foliage_Color', 'pH_Min', 'pH_Max','Precip_Min', 'Precip_Max',
'Shade_Tolerance', 'Temp_Min_F')


## plants : too messy data set (because of too many NAs) => need to be cleaned for appropriate analysis

plants.NA.removed <- na.omit(plants)

## plants.NA.removed


## Statistic : range = plants$pH_Max - plants$pH_Min ( we will use this statistic value as Dependent Va

# plants$pH_Min
# plants$pH_Max

plants.range <- plants.NA.removed$pH_Max - plants.NA.removed$pH_Min

plants.range
```

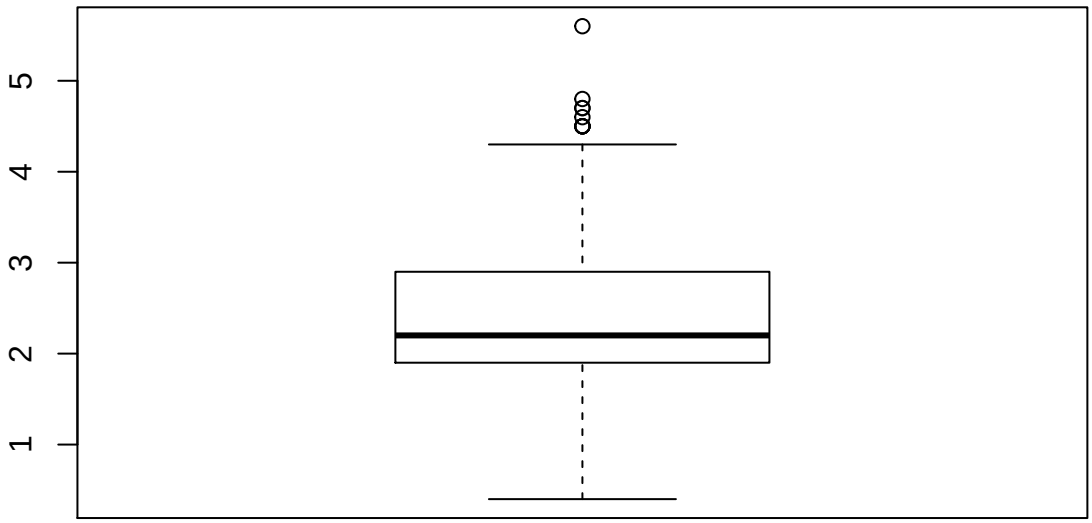
```
##   [1] 2.0 1.5 1.1 2.8 2.8 2.1 2.4 1.2 2.6 3.3 4.2 2.2 2.0 2.0 2.0 2.0 2.0
##  [18] 2.0 2.5 2.0 2.5 2.0 2.5 2.6 3.5 2.5 2.0 2.0 2.4 3.8 2.0 2.5 2.0 2.0
##  [35] 2.9 2.0 2.0 2.5 3.5 3.5 3.5 2.2 2.7 2.0 2.2 2.0 2.7 1.9 2.5 2.0 3.5
##  [52] 4.0 2.2 1.5 1.5 1.3 3.5 2.1 2.0 1.5 2.5 2.2 1.8 2.5 1.0 2.2 2.0 3.0
##  [69] 1.9 2.2 3.0 2.0 2.5 2.0 2.5 1.5 3.2 2.3 2.4 1.2 1.7 4.0 3.2 3.5 3.2
##  [86] 3.0 2.0 1.9 1.9 1.9 2.0 2.0 2.0 3.6 2.2 1.9 1.7 1.4 2.5 1.6 1.3 3.0
## [103] 1.2 1.2 1.2 3.0 2.3 1.9 2.5 2.0 2.5 2.5 1.3 2.2 3.5 3.3 3.5 3.0 2.2
## [120] 1.6 1.9 2.3 1.5 2.0 1.9 3.5 1.8 1.7 2.0 2.0 3.5 1.1 3.5 1.5 2.2 2.6
## [137] 1.7 2.1 2.2 2.9 3.5 1.9 2.0 2.1 2.2 1.6 2.5 1.3 4.0 2.2 1.2 1.5 2.3
## [154] 1.8 2.1 2.6 1.2 2.1 1.8 0.8 1.9 2.1 2.0 2.0 2.0 2.2 2.1 2.3 2.2 1.8
## [171] 1.7 3.0 2.2 3.5 2.1 2.0 2.2 1.3 2.3 1.3 2.0 3.0 3.0 2.1 3.4 2.6 2.5
## [188] 3.0 1.6 3.3 1.0 1.9 2.0 1.5 2.5 2.2 2.5 2.5 3.3 1.8 3.9 2.6 2.9 3.5
## [205] 2.0 2.8 1.0 2.0 2.5 2.5 2.0 2.0 1.5 4.5 2.3 1.6 1.8 2.5 1.8 2.0 2.0
## [222] 2.0 3.0 2.0 2.0 2.4 1.0 2.6 2.5 2.0 1.4 2.9 1.4 2.6 1.4 2.7 2.0 2.0
## [239] 2.7 2.7 2.7 3.0 1.5 3.0 1.9 2.0 1.5 1.5 2.0 2.0 2.0 3.3 2.0 2.2 0.6
## [256] 1.5 2.5 2.2 3.7 2.0 3.0 1.0 1.4 2.5 3.5 2.5 2.5 2.5 3.0 3.0 2.2 1.5
## [273] 2.8 3.6 3.0 3.0 2.8 0.7 0.7 4.5 2.7 5.6 2.7 3.5 2.9 2.5 4.6 4.0 0.8
## [290] 2.2 3.0 2.9 2.6 2.7 3.4 2.0 3.0 2.5 3.0 1.5 2.5 4.0 4.0 3.5 3.0 2.4
## [307] 3.6 2.5 2.7 2.5 2.0 3.1 2.5 1.7 2.5 2.8 3.8 3.4 2.8 1.8 2.0 2.2 2.4
## [324] 2.4 3.4 2.5 2.5 2.0 2.2 1.7 1.4 1.4 3.0 2.0 2.2 2.2 2.3 3.2 3.5 3.5
## [341] 3.0 4.0 2.3 1.7 2.6 3.5 3.0 3.0 2.3 2.0 1.5 3.0 2.0 1.9 2.0 1.7 3.5
## [358] 3.0 2.5 1.8 3.5 1.4 2.6 1.4 2.4 3.5 1.6 2.4 2.6 2.6 2.4 2.1 3.0 2.5
## [375] 2.5 3.0 1.0 0.6 2.5 3.5 0.7 1.0 3.6 2.8 2.7 2.5 2.2 1.2 3.0 1.4 2.0
## [392] 3.3 1.3 2.3 1.5 3.0 1.2 1.9 1.5 2.5 2.0 1.8 2.5 3.3 2.2 1.0 2.0 3.5
## [409] 2.0 2.4 3.3 3.4 2.5 1.0 1.0 0.4 3.0 3.7 4.0 2.0 3.5 2.5 1.1 1.8 1.9
## [426] 1.7 1.4 1.8 2.5 0.9 2.5 1.5 3.0 2.5 2.0 3.0 2.0 3.0 2.9 2.9 2.5 2.5
## [443] 2.3 3.0 2.0 3.5 1.3 2.6 2.0 1.3 2.0 2.4 3.0 1.8 2.0 2.5 1.9 2.2 2.5
## [460] 2.5 1.6 1.7 2.0 1.7 3.5 3.0 1.6 2.9 2.0 2.5 3.7 2.3 2.0 2.0 2.4 1.0
## [477] 2.0 2.2 2.0 3.0 1.6 2.0 2.0 1.2 1.5 2.0 2.5 2.5 2.5 2.0 2.0 0.9 3.4
## [494] 2.5 2.0 4.0 3.5 2.5 2.2 1.0 2.5 3.5 2.5 1.5 1.6 4.0 2.6 3.0 3.5 3.0
## [511] 3.0 3.8 1.3 1.5 2.0 2.0 2.5 2.2 2.8 1.7 1.2 0.9 1.8 2.0 4.2 2.5 2.5
## [528] 2.0 3.3 2.0 4.2 2.3 1.8 2.0 2.5 1.5 1.6 2.0 2.5 2.5 3.0 3.0 1.2 2.5
```

```
## [545] 1.6 2.5 3.2 1.1 2.2 2.0 2.6 3.4 1.1 2.7 3.0 2.5 4.3 4.5 4.5 2.7 4.5
## [562] 3.0 3.8 2.1 2.5 1.9 2.4 1.3 3.5 4.7 1.5 2.1 2.7 1.6 1.5 1.2 1.2 1.2
## [579] 1.2 1.2 2.0 2.6 2.0 2.5 1.9 3.0 3.5 1.5 3.2 2.2 2.5 1.5 2.3 2.2 2.4
## [596] 2.2 1.5 2.3 3.0 3.0 1.0 2.0 3.0 2.5 1.5 2.0 2.0 3.0 2.6 1.7 2.0 2.5
## [613] 3.5 1.5 2.5 1.6 3.6 4.0 1.5 1.5 1.5 1.5 1.2 3.0 2.2 2.2 1.7 2.3 2.0
## [630] 3.6 2.0 3.5 3.0 3.0 3.0 2.0 2.5 2.0 2.9 2.2 2.5 2.0 2.0 2.5 2.5 2.5
## [647] 2.3 2.0 1.0 2.5 3.0 2.5 3.2 3.5 3.5 2.2 2.3 4.2 3.3 4.3 3.0 3.0 1.1
## [664] 3.8 1.4 3.2 1.0 3.0 1.8 3.9 2.0 2.0 2.8 3.4 3.5 3.8 2.0 2.1 4.0 2.4
## [681] 2.5 2.1 3.7 3.5 3.0 1.6 2.0 3.0 3.0 2.0 4.0 3.0 3.0 2.5 3.0 1.5 1.5
## [698] 2.7 2.5 1.7 4.0 2.0 1.0 2.5 2.5 2.0 1.2 1.5 2.0 3.2 2.0 2.0 2.0 2.4
## [715] 3.5 1.6 1.7 2.0 2.5 2.5 2.5 1.5 2.5 2.0 3.6 1.5 1.2 2.0 2.3 1.7 2.5
## [732] 3.7 1.8 1.4 2.0 1.8 2.0 2.0 1.5 2.0 1.6 1.7 3.0 1.4 3.0 2.0 2.0 2.7
## [749] 1.5 2.2 3.5 3.0 1.8 3.0 1.6 2.4 0.9 4.0 1.0 2.6 2.0 1.9 2.0 2.1 2.8
## [766] 2.4 2.5 1.5 4.8 3.2 2.0 3.0 2.2 2.5 2.5 2.7 1.5 3.4 2.0 1.4 2.8 2.8
## [783] 3.0 2.1 3.5 1.8 2.7 2.8 2.0 2.1 2.0 2.1 1.8 2.7 2.6 1.3 2.1 0.9 1.5
## [800] 2.0 2.2 1.3 1.0 1.8 1.7 1.6 2.4 2.0 1.5 3.0 2.0 1.0 4.7
```

```r
summary(plants.range)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.400   1.900   2.200   2.347   2.900   5.600
```

```r
boxplot(plants.range)
```




```r
Foliage_Color <- plants.NA.removed$Foliage_Color  ## We use this factor or explanatory variable (X) for

summary(Foliage_Color)
```

```
##   Dark Green   Gray-Green        Green          Red   White-Gray
##           82           24          675            3            9
## Yellow-Green
##           20
```

```r
## Data cleaning

## What we need is the set of pHs and Foliage_color data set

required.data.set <- data.frame(plants.range,Foliage_Color)
```

```
required.data.set
```

```
##     plants.range Foliage_Color
## 1            2.0         Green
## 2            1.5         Green
## 3            1.1         Green
## 4            2.8         Green
## 5            2.8         Green
## 6            2.1         Green
## 7            2.4         Green
## 8            1.2  Yellow-Green
## 9            2.6         Green
## 10           3.3         Green
## 11           4.2         Green
## 12           2.2         Green
## 13           2.0         Green
## 14           2.0         Green
## 15           2.0         Green
## 16           2.0         Green
## 17           2.0    Dark Green
## 18           2.0         Green
## 19           2.5         Green
## 20           2.0         Green
## 21           2.5         Green
## 22           2.0         Green
## 23           2.5         Green
## 24           2.6         Green
## 25           3.5         Green
## 26           2.5         Green
## 27           2.0         Green
## 28           2.0         Green
## 29           2.4         Green
## 30           3.8         Green
## 31           2.0         Green
## 32           2.5    White-Gray
## 33           2.0         Green
## 34           2.0         Green
## 35           2.9    Dark Green
## 36           2.0         Green
## 37           2.0         Green
## 38           2.5         Green
## 39           3.5         Green
## 40           3.5         Green
## 41           3.5         Green
## 42           2.2         Green
## 43           2.7    Dark Green
## 44           2.0         Green
## 45           2.2         Green
## 46           2.0    Dark Green
## 47           2.7         Green
## 48           1.9         Green
## 49           2.5         Green
## 50           2.0         Green
## 51           3.5         Green
```

```
## 52          4.0         Green
## 53          2.2         Green
## 54          1.5         Green
## 55          1.5         Green
## 56          1.3         Green
## 57          3.5         Green
## 58          2.1         Green
## 59          2.0    Dark Green
## 60          1.5         Green
## 61          2.5         Green
## 62          2.2    Dark Green
## 63          1.8         Green
## 64          2.5    Dark Green
## 65          1.0         Green
## 66          2.2         Green
## 67          2.0         Green
## 68          3.0   White-Gray
## 69          1.9    Dark Green
## 70          2.2         Green
## 71          3.0         Green
## 72          2.0         Green
## 73          2.5    Dark Green
## 74          2.0         Green
## 75          2.5         Green
## 76          1.5         Green
## 77          3.2         Green
## 78          2.3    Dark Green
## 79          2.4         Green
## 80          1.2   Gray-Green
## 81          1.7    Dark Green
## 82          4.0         Green
## 83          3.2         Green
## 84          3.5         Green
## 85          3.2         Green
## 86          3.0         Green
## 87          2.0         Green
## 88          1.9         Green
## 89          1.9         Green
## 90          1.9         Green
## 91          2.0         Green
## 92          2.0         Green
## 93          2.0         Green
## 94          3.6         Green
## 95          2.2         Green
## 96          1.9         Green
## 97          1.7         Green
## 98          1.4         Green
## 99          2.5         Green
## 100         1.6         Green
## 101         1.3         Green
## 102         3.0         Green
## 103         1.2         Green
## 104         1.2    Dark Green
## 105         1.2    Dark Green
```

```
## 106          3.0         Green
## 107          2.3         Green
## 108          1.9         Green
## 109          2.5         Green
## 110          2.0         Green
## 111          2.5         Green
## 112          2.5         Green
## 113          1.3         Green
## 114          2.2         Green
## 115          3.5    Gray-Green
## 116          3.3         Green
## 117          3.5         Green
## 118          3.0         Green
## 119          2.2         Green
## 120          1.6         Green
## 121          1.9         Green
## 122          2.3         Green
## 123          1.5         Green
## 124          2.0         Green
## 125          1.9         Green
## 126          3.5         Green
## 127          1.8         Green
## 128          1.7         Green
## 129          2.0         Green
## 130          2.0         Green
## 131          3.5         Green
## 132          1.1         Green
## 133          3.5         Green
## 134          1.5         Green
## 135          2.2         Green
## 136          2.6         Green
## 137          1.7         Green
## 138          2.1    Gray-Green
## 139          2.2         Green
## 140          2.9         Green
## 141          3.5         Green
## 142          1.9         Green
## 143          2.0         Green
## 144          2.1         Green
## 145          2.2         Green
## 146          1.6         Green
## 147          2.5    Gray-Green
## 148          1.3         Green
## 149          4.0         Green
## 150          2.2    Dark Green
## 151          1.2         Green
## 152          1.5         Green
## 153          2.3    Gray-Green
## 154          1.8         Green
## 155          2.1    Dark Green
## 156          2.6         Green
## 157          1.2         Green
## 158          2.1         Green
## 159          1.8         Green
```

```
## 160          0.8         Green
## 161          1.9         Green
## 162          2.1         Green
## 163          2.0         Green
## 164          2.0    Dark Green
## 165          2.0         Green
## 166          2.2         Green
## 167          2.1         Green
## 168          2.3         Green
## 169          2.2         Green
## 170          1.8    Dark Green
## 171          1.7         Green
## 172          3.0         Green
## 173          2.2         Green
## 174          3.5         Green
## 175          2.1         Green
## 176          2.0         Green
## 177          2.2         Green
## 178          1.3         Green
## 179          2.3         Green
## 180          1.3         Green
## 181          2.0         Green
## 182          3.0         Green
## 183          3.0         Green
## 184          2.1         Green
## 185          3.4         Green
## 186          2.6         Green
## 187          2.5         Green
## 188          3.0  Yellow-Green
## 189          1.6         Green
## 190          3.3         Green
## 191          1.0  Yellow-Green
## 192          1.9         Green
## 193          2.0         Green
## 194          1.5    Dark Green
## 195          2.5         Green
## 196          2.2         Green
## 197          2.5         Green
## 198          2.5         Green
## 199          3.3         Green
## 200          1.8         Green
## 201          3.9         Green
## 202          2.6    Gray-Green
## 203          2.9         Green
## 204          3.5         Green
## 205          2.0         Green
## 206          2.8         Green
## 207          1.0  Yellow-Green
## 208          2.0         Green
## 209          2.5         Green
## 210          2.5         Green
## 211          2.0         Green
## 212          2.0         Green
## 213          1.5         Green
```

```
## 214          4.5         Green
## 215          2.3         Green
## 216          1.6         Green
## 217          1.8         Green
## 218          2.5         Green
## 219          1.8         Green
## 220          2.0         Green
## 221          2.0  Yellow-Green
## 222          2.0         Green
## 223          3.0         Green
## 224          2.0         Green
## 225          2.0         Green
## 226          2.4         Green
## 227          1.0         Green
## 228          2.6         Green
## 229          2.5    Dark Green
## 230          2.0         Green
## 231          1.4    Dark Green
## 232          2.9         Green
## 233          1.4         Green
## 234          2.6         Green
## 235          1.4         Green
## 236          2.7         Green
## 237          2.0    White-Gray
## 238          2.0         Green
## 239          2.7         Green
## 240          2.7         Green
## 241          2.7         Green
## 242          3.0  Yellow-Green
## 243          1.5         Green
## 244          3.0         Green
## 245          1.9         Green
## 246          2.0         Green
## 247          1.5         Green
## 248          1.5    Dark Green
## 249          2.0         Green
## 250          2.0         Green
## 251          2.0         Green
## 252          3.3         Green
## 253          2.0         Green
## 254          2.2         Green
## 255          0.6         Green
## 256          1.5         Green
## 257          2.5         Green
## 258          2.2         Green
## 259          3.7         Green
## 260          2.0         Green
## 261          3.0         Green
## 262          1.0         Green
## 263          1.4         Green
## 264          2.5    Gray-Green
## 265          3.5         Green
## 266          2.5         Green
## 267          2.5         Green
```

```
## 268          2.5         Green
## 269          3.0         Green
## 270          3.0    Gray-Green
## 271          2.2    Dark Green
## 272          1.5    Dark Green
## 273          2.8    Dark Green
## 274          3.6         Green
## 275          3.0         Green
## 276          3.0    Dark Green
## 277          2.8    Dark Green
## 278          0.7         Green
## 279          0.7         Green
## 280          4.5         Green
## 281          2.7         Green
## 282          5.6         Green
## 283          2.7         Green
## 284          3.5    White-Gray
## 285          2.9    White-Gray
## 286          2.5    White-Gray
## 287          4.6    Dark Green
## 288          4.0    Dark Green
## 289          0.8    Dark Green
## 290          2.2    Dark Green
## 291          3.0    Gray-Green
## 292          2.9         Green
## 293          2.6         Green
## 294          2.7    Dark Green
## 295          3.4         Green
## 296          2.0         Green
## 297          3.0         Green
## 298          2.5         Green
## 299          3.0         Green
## 300          1.5         Green
## 301          2.5         Green
## 302          4.0         Green
## 303          4.0         Green
## 304          3.5         Green
## 305          3.0         Green
## 306          2.4         Green
## 307          3.6    Dark Green
## 308          2.5         Green
## 309          2.7         Green
## 310          2.5         Green
## 311          2.0    Dark Green
## 312          3.1         Green
## 313          2.5         Green
## 314          1.7         Green
## 315          2.5         Green
## 316          2.8         Green
## 317          3.8         Green
## 318          3.4  Yellow-Green
## 319          2.8         Green
## 320          1.8         Green
## 321          2.0         Green
```

```
## 322          2.2         Green
## 323          2.4         Green
## 324          2.4         Green
## 325          3.4         Green
## 326          2.5    Dark Green
## 327          2.5    Dark Green
## 328          2.0         Green
## 329          2.2         Green
## 330          1.7         Green
## 331          1.4         Green
## 332          1.4         Green
## 333          3.0         Green
## 334          2.0         Green
## 335          2.2         Green
## 336          2.2         Green
## 337          2.3         Green
## 338          3.2         Green
## 339          3.5         Green
## 340          3.5         Green
## 341          3.0         Green
## 342          4.0         Green
## 343          2.3         Green
## 344          1.7         Green
## 345          2.6         Green
## 346          3.5         Green
## 347          3.0         Green
## 348          3.0    Dark Green
## 349          2.3         Green
## 350          2.0         Green
## 351          1.5    Dark Green
## 352          3.0  Yellow-Green
## 353          2.0    Gray-Green
## 354          1.9         Green
## 355          2.0         Green
## 356          1.7         Green
## 357          3.5    Gray-Green
## 358          3.0    White-Gray
## 359          2.5         Green
## 360          1.8         Green
## 361          3.5         Green
## 362          1.4    Gray-Green
## 363          2.6         Green
## 364          1.4         Green
## 365          2.4         Green
## 366          3.5         Green
## 367          1.6         Green
## 368          2.4         Green
## 369          2.6         Green
## 370          2.6         Green
## 371          2.4         Green
## 372          2.1         Green
## 373          3.0    Dark Green
## 374          2.5         Green
## 375          2.5         Green
```

```
## 376          3.0         Green
## 377          1.0         Green
## 378          0.6         Green
## 379          2.5         Green
## 380          3.5         Green
## 381          0.7         Green
## 382          1.0         Green
## 383          3.6         Green
## 384          2.8         Green
## 385          2.7         Green
## 386          2.5         Green
## 387          2.2         Green
## 388          1.2         Green
## 389          3.0         Green
## 390          1.4         Green
## 391          2.0           Red
## 392          3.3    Dark Green
## 393          1.3         Green
## 394          2.3         Green
## 395          1.5         Green
## 396          3.0    Dark Green
## 397          1.2         Green
## 398          1.9         Green
## 399          1.5         Green
## 400          2.5         Green
## 401          2.0         Green
## 402          1.8         Green
## 403          2.5         Green
## 404          3.3         Green
## 405          2.2    Dark Green
## 406          1.0         Green
## 407          2.0         Green
## 408          3.5         Green
## 409          2.0         Green
## 410          2.4         Green
## 411          3.3         Green
## 412          3.4    Gray-Green
## 413          2.5         Green
## 414          1.0  Yellow-Green
## 415          1.0    Dark Green
## 416          0.4  Yellow-Green
## 417          3.0         Green
## 418          3.7    Dark Green
## 419          4.0         Green
## 420          2.0         Green
## 421          3.5         Green
## 422          2.5         Green
## 423          1.1    Dark Green
## 424          1.8         Green
## 425          1.9         Green
## 426          1.7         Green
## 427          1.4    Dark Green
## 428          1.8         Green
## 429          2.5         Green
```

```
## 430          0.9         Green
## 431          2.5         Green
## 432          1.5         Green
## 433          3.0         Green
## 434          2.5         Green
## 435          2.0         Green
## 436          3.0         Green
## 437          2.0         Green
## 438          3.0         Green
## 439          2.9         Green
## 440          2.9         Green
## 441          2.5         Green
## 442          2.5         Green
## 443          2.3         Green
## 444          3.0         Green
## 445          2.0         Green
## 446          3.5    Gray-Green
## 447          1.3         Green
## 448          2.6         Green
## 449          2.0         Green
## 450          1.3    Dark Green
## 451          2.0         Green
## 452          2.4         Green
## 453          3.0         Green
## 454          1.8         Green
## 455          2.0         Green
## 456          2.5         Green
## 457          1.9    Dark Green
## 458          2.2         Green
## 459          2.5    Dark Green
## 460          2.5         Green
## 461          1.6    Dark Green
## 462          1.7         Green
## 463          2.0         Green
## 464          1.7         Green
## 465          3.5         Green
## 466          3.0         Green
## 467          1.6         Green
## 468          2.9         Green
## 469          2.0    Gray-Green
## 470          2.5    Dark Green
## 471          3.7         Green
## 472          2.3         Green
## 473          2.0         Green
## 474          2.0         Green
## 475          2.4  Yellow-Green
## 476          1.0  Yellow-Green
## 477          2.0         Green
## 478          2.2         Green
## 479          2.0         Green
## 480          3.0         Green
## 481          1.6         Green
## 482          2.0         Green
## 483          2.0         Green
```

```
## 484          1.2         Green
## 485          1.5         Green
## 486          2.0         Green
## 487          2.5         Green
## 488          2.5         Green
## 489          2.5         Green
## 490          2.0    Dark Green
## 491          2.0    Dark Green
## 492          0.9         Green
## 493          3.4         Green
## 494          2.5         Green
## 495          2.0    Dark Green
## 496          4.0         Green
## 497          3.5         Green
## 498          2.5         Green
## 499          2.2         Green
## 500          1.0         Green
## 501          2.5         Green
## 502          3.5         Green
## 503          2.5         Green
## 504          1.5    Dark Green
## 505          1.6         Green
## 506          4.0         Green
## 507          2.6    Gray-Green
## 508          3.0         Green
## 509          3.5         Green
## 510          3.0         Green
## 511          3.0    Gray-Green
## 512          3.8         Green
## 513          1.3         Green
## 514          1.5           Red
## 515          2.0         Green
## 516          2.0         Green
## 517          2.5         Green
## 518          2.2         Green
## 519          2.8         Green
## 520          1.7         Green
## 521          1.2         Green
## 522          0.9         Green
## 523          1.8         Green
## 524          2.0         Green
## 525          4.2         Green
## 526          2.5         Green
## 527          2.5         Green
## 528          2.0         Green
## 529          3.3         Green
## 530          2.0    Dark Green
## 531          4.2         Green
## 532          2.3         Green
## 533          1.8         Green
## 534          2.0         Green
## 535          2.5         Green
## 536          1.5  Yellow-Green
## 537          1.6  Yellow-Green
```

```
## 538          2.0         Green
## 539          2.5    Dark Green
## 540          2.5         Green
## 541          3.0         Green
## 542          3.0         Green
## 543          1.2         Green
## 544          2.5         Green
## 545          1.6         Green
## 546          2.5         Green
## 547          3.2         Green
## 548          1.1         Green
## 549          2.2         Green
## 550          2.0         Green
## 551          2.6         Green
## 552          3.4         Green
## 553          1.1         Green
## 554          2.7         Green
## 555          3.0         Green
## 556          2.5         Green
## 557          4.3    Dark Green
## 558          4.5         Green
## 559          4.5         Green
## 560          2.7         Green
## 561          4.5         Green
## 562          3.0         Green
## 563          3.8         Green
## 564          2.1         Green
## 565          2.5  Yellow-Green
## 566          1.9         Green
## 567          2.4         Green
## 568          1.3         Green
## 569          3.5         Green
## 570          4.7         Green
## 571          1.5         Green
## 572          2.1           Red
## 573          2.7         Green
## 574          1.6         Green
## 575          1.5         Green
## 576          1.2         Green
## 577          1.2         Green
## 578          1.2         Green
## 579          1.2         Green
## 580          1.2         Green
## 581          2.0         Green
## 582          2.6    Gray-Green
## 583          2.0         Green
## 584          2.5         Green
## 585          1.9         Green
## 586          3.0         Green
## 587          3.5         Green
## 588          1.5         Green
## 589          3.2         Green
## 590          2.2         Green
## 591          2.5    Dark Green
```

```
## 592          1.5         Green
## 593          2.3         Green
## 594          2.2         Green
## 595          2.4         Green
## 596          2.2         Green
## 597          1.5    Dark Green
## 598          2.3         Green
## 599          3.0         Green
## 600          3.0         Green
## 601          1.0    Dark Green
## 602          2.0         Green
## 603          3.0         Green
## 604          2.5         Green
## 605          1.5         Green
## 606          2.0         Green
## 607          2.0         Green
## 608          3.0         Green
## 609          2.6         Green
## 610          1.7         Green
## 611          2.0         Green
## 612          2.5         Green
## 613          3.5    Gray-Green
## 614          1.5         Green
## 615          2.5         Green
## 616          1.6         Green
## 617          3.6         Green
## 618          4.0         Green
## 619          1.5    Dark Green
## 620          1.5    Dark Green
## 621          1.5         Green
## 622          1.5    Dark Green
## 623          1.2         Green
## 624          3.0         Green
## 625          2.2         Green
## 626          2.2         Green
## 627          1.7         Green
## 628          2.3  Yellow-Green
## 629          2.0         Green
## 630          3.6         Green
## 631          2.0         Green
## 632          3.5         Green
## 633          3.0         Green
## 634          3.0         Green
## 635          3.0         Green
## 636          2.0         Green
## 637          2.5    Dark Green
## 638          2.0         Green
## 639          2.9         Green
## 640          2.2         Green
## 641          2.5         Green
## 642          2.0         Green
## 643          2.0         Green
## 644          2.5    Gray-Green
## 645          2.5         Green
```

```
## 646          2.5         Green
## 647          2.3         Green
## 648          2.0         Green
## 649          1.0         Green
## 650          2.5    Gray-Green
## 651          3.0         Green
## 652          2.5         Green
## 653          3.2         Green
## 654          3.5         Green
## 655          3.5         Green
## 656          2.2         Green
## 657          2.3         Green
## 658          4.2         Green
## 659          3.3         Green
## 660          4.3         Green
## 661          3.0         Green
## 662          3.0         Green
## 663          1.1    Dark Green
## 664          3.8         Green
## 665          1.4  Yellow-Green
## 666          3.2         Green
## 667          1.0    Dark Green
## 668          3.0         Green
## 669          1.8         Green
## 670          3.9         Green
## 671          2.0         Green
## 672          2.0         Green
## 673          2.8         Green
## 674          3.4    Gray-Green
## 675          3.5         Green
## 676          3.8    Dark Green
## 677          2.0         Green
## 678          2.1         Green
## 679          4.0    Dark Green
## 680          2.4         Green
## 681          2.5    Dark Green
## 682          2.1         Green
## 683          3.7         Green
## 684          3.5         Green
## 685          3.0         Green
## 686          1.6         Green
## 687          2.0         Green
## 688          3.0         Green
## 689          3.0         Green
## 690          2.0         Green
## 691          4.0         Green
## 692          3.0    Gray-Green
## 693          3.0         Green
## 694          2.5         Green
## 695          3.0         Green
## 696          1.5         Green
## 697          1.5    Dark Green
## 698          2.7         Green
## 699          2.5    Dark Green
```

```
## 700          1.7         Green
## 701          4.0         Green
## 702          2.0         Green
## 703          1.0         Green
## 704          2.5         Green
## 705          2.5         Green
## 706          2.0    Dark Green
## 707          1.2         Green
## 708          1.5         Green
## 709          2.0         Green
## 710          3.2         Green
## 711          2.0         Green
## 712          2.0         Green
## 713          2.0         Green
## 714          2.4         Green
## 715          3.5         Green
## 716          1.6         Green
## 717          1.7         Green
## 718          2.0         Green
## 719          2.5         Green
## 720          2.5         Green
## 721          2.5         Green
## 722          1.5         Green
## 723          2.5         Green
## 724          2.0         Green
## 725          3.6         Green
## 726          1.5         Green
## 727          1.2         Green
## 728          2.0         Green
## 729          2.3    Dark Green
## 730          1.7         Green
## 731          2.5    Dark Green
## 732          3.7         Green
## 733          1.8         Green
## 734          1.4         Green
## 735          2.0         Green
## 736          1.8    Gray-Green
## 737          2.0         Green
## 738          2.0         Green
## 739          1.5         Green
## 740          2.0    Dark Green
## 741          1.6         Green
## 742          1.7         Green
## 743          3.0         Green
## 744          1.4         Green
## 745          3.0         Green
## 746          2.0  Yellow-Green
## 747          2.0         Green
## 748          2.7         Green
## 749          1.5         Green
## 750          2.2         Green
## 751          3.5    White-Gray
## 752          3.0  Yellow-Green
## 753          1.8         Green
```

```
## 754          3.0         Green
## 755          1.6         Green
## 756          2.4         Green
## 757          0.9    White-Gray
## 758          4.0         Green
## 759          1.0         Green
## 760          2.6         Green
## 761          2.0         Green
## 762          1.9         Green
## 763          2.0         Green
## 764          2.1         Green
## 765          2.8         Green
## 766          2.4         Green
## 767          2.5         Green
## 768          1.5    Dark Green
## 769          4.8         Green
## 770          3.2         Green
## 771          2.0         Green
## 772          3.0         Green
## 773          2.2         Green
## 774          2.5         Green
## 775          2.5         Green
## 776          2.7         Green
## 777          1.5         Green
## 778          3.4         Green
## 779          2.0         Green
## 780          1.4         Green
## 781          2.8         Green
## 782          2.8         Green
## 783          3.0         Green
## 784          2.1         Green
## 785          3.5    Dark Green
## 786          1.8         Green
## 787          2.7         Green
## 788          2.8  Yellow-Green
## 789          2.0         Green
## 790          2.1         Green
## 791          2.0         Green
## 792          2.1         Green
## 793          1.8         Green
## 794          2.7         Green
## 795          2.6         Green
## 796          1.3         Green
## 797          2.1         Green
## 798          0.9    Dark Green
## 799          1.5         Green
## 800          2.0         Green
## 801          2.2    Dark Green
## 802          1.3         Green
## 803          1.0    Dark Green
## 804          1.8         Green
## 805          1.7         Green
## 806          1.6    Dark Green
## 807          2.4         Green
```

```
## 808          2.0         Green
## 809          1.5         Green
## 810          3.0         Green
## 811          2.0    Dark Green
## 812          1.0         Green
## 813          4.7         Green
```

```
## With the above cleaned data set, we do the ANOVA.

lm.ph.Foliage.color <- lm(plants.range ~ Foliage_Color,data=required.data.set)

anova(lm.ph.Foliage.color)
```

```
## Analysis of Variance Table
##
## Response: plants.range
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Foliage_Color   5   8.97 1.79434   2.975 0.01141 *
## Residuals     807 486.73 0.60314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## ANOVA TABLE
## From the above ANOVA table, the $H_0 : \beta_1=\beta_2=\beta_3=\beta_4=\beta_5 =0$ has been rejected



summary(lm.ph.Foliage.color)
```

```
##
## Call:
## lm(formula = plants.range ~ Foliage_Color, data = required.data.set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7665 -0.4665 -0.1444  0.5250  3.2335
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.17683    0.08576  25.382   <2e-16 ***
## Foliage_ColorGray-Green     0.46484    0.18024   2.579   0.0101 *
## Foliage_ColorGreen          0.18969    0.09082   2.089   0.0371 *
## Foliage_ColorRed           -0.31016    0.45651  -0.679   0.4971
## Foliage_ColorWhite-Gray     0.46762    0.27271   1.715   0.0868 .
## Foliage_ColorYellow-Green -0.20183    0.19368  -1.042   0.2977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7766 on 807 degrees of freedom
## Multiple R-squared:  0.0181, Adjusted R-squared:  0.01202
## F-statistic: 2.975 on 5 and 807 DF,  p-value: 0.01141
```

```
## Summary table
## From the above Summary table, we can figure out that between (Foliage-Color-Gray-Green and Plants_ra
## (Foliage-ColorGreen and Plants_range), there are significant relationship under the p-value 0.0101,
```

Problem 6

Finish this homework by pushing your changes to your repo. In general, your workflow for this should be:

1. git pull – to make sure you have the most recent repo
2. In R: do some work
3. git add – this tells git to track new files
4. git commit – make message INFORMATIVE and USEFUL
5. git push – this pushes your local changes to the repo

If you have difficulty with steps 1-5, git is not correctly or completely setup. See me for help.