Homework 3

Hwasoo Shin

2019 9 6

Problem 3

Most of all, I think Github will help me to save back-up files easily. Also, when I want to share my data or work with other people, it will be helpful to use functions in Github. Also, once you know how to use version control (especially in Git), I will be able to compare and add files through syntax.

Problem 4

*Sensory Data

First we should see how the data looks like, and clean it.

```
## Warning: package 'stringr' was built under R version 3.6.1
```

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

Item	1	2	3	4	5
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	4.3	4.9	3.3	5.3	4.4
1	4.3	4.5	4.0	5.5	3.3
1	4.1	5.3	3.4	5.7	4.7
2	6.0	5.3	4.5	5.9	4.7
2	4.9	6.3	4.2	5.5	4.9
2	6.0	5.9	4.7	6.3	4.6
3	2.4	2.5	2.3	3.1	2.4
3	3.9	3.0	2.8	2.7	1.3

Item	1	2	3	4	5
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
3	1.9	3.9	2.6	4.6	2.2
4	7.4	8.2	6.4	6.8	6.0

1-10 of 30 rows Previous **1** 2 3 Next

Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
4.3	6.0	2.4	7.4	5.7	2.2	1.2	4.2	8.0	5.0
4.3	4.9	3.9	7.1	5.8	3.0	1.3	3.0	9.0	5.4
4.1	6.0	1.9	6.4	5.8	2.1	0.9	4.8	8.9	2.8
4.9	5.3	2.5	8.2	6.3	2.4	1.5	4.8	8.6	4.8
4.5	6.3	3.0	7.9	5.7	1.8	2.4	4.5	7.7	5.0
5.3	5.9	3.9	7.1	6.0	3.3	3.1	4.8	9.2	5.2
3.3	4.5	2.3	6.4	5.4	1.7	1.2	4.5	9.0	3.9
4.0	4.2	2.8	5.9	5.4	2.1	0.8	4.7	6.7	3.4
3.4	4.7	2.6	6.9	6.1	1.1	1.1	4.7	8.1	4.1
5.3	5.9	3.1	6.8	6.1	3.4	0.9	4.6	9.4	5.5

1-10 of 15 rows Previous **1** 2 Next

Second, we can do some analysis about the data.

```
Sensory2<-Sensory[,-1]
summary(Sensory2)
```

```
##        1               2               3               4        
##  Min.   :0.900   Min.   :1.500   Min.   :0.800   Min.   :0.900  
##  1st Qu.:2.850   1st Qu.:3.450   1st Qu.:2.650   1st Qu.:3.925  
##  Median :4.550   Median :4.950   Median :4.150   Median :5.400  
##  Mean   :4.593   Mean   :5.063   Mean   :4.167   Mean   :5.193  
##  3rd Qu.:5.950   3rd Qu.:6.225   3rd Qu.:5.400   3rd Qu.:6.275  
##  Max.   :9.000   Max.   :9.200   Max.   :9.000   Max.   :9.400  
##        5        
##  Min.   :0.700  
##  1st Qu.:2.250  
##  Median :4.600  
##  Mean   :4.267  
##  3rd Qu.:5.800  
##  Max.   :8.800
```

```
#This is the summary of each variable
```

We can see that the

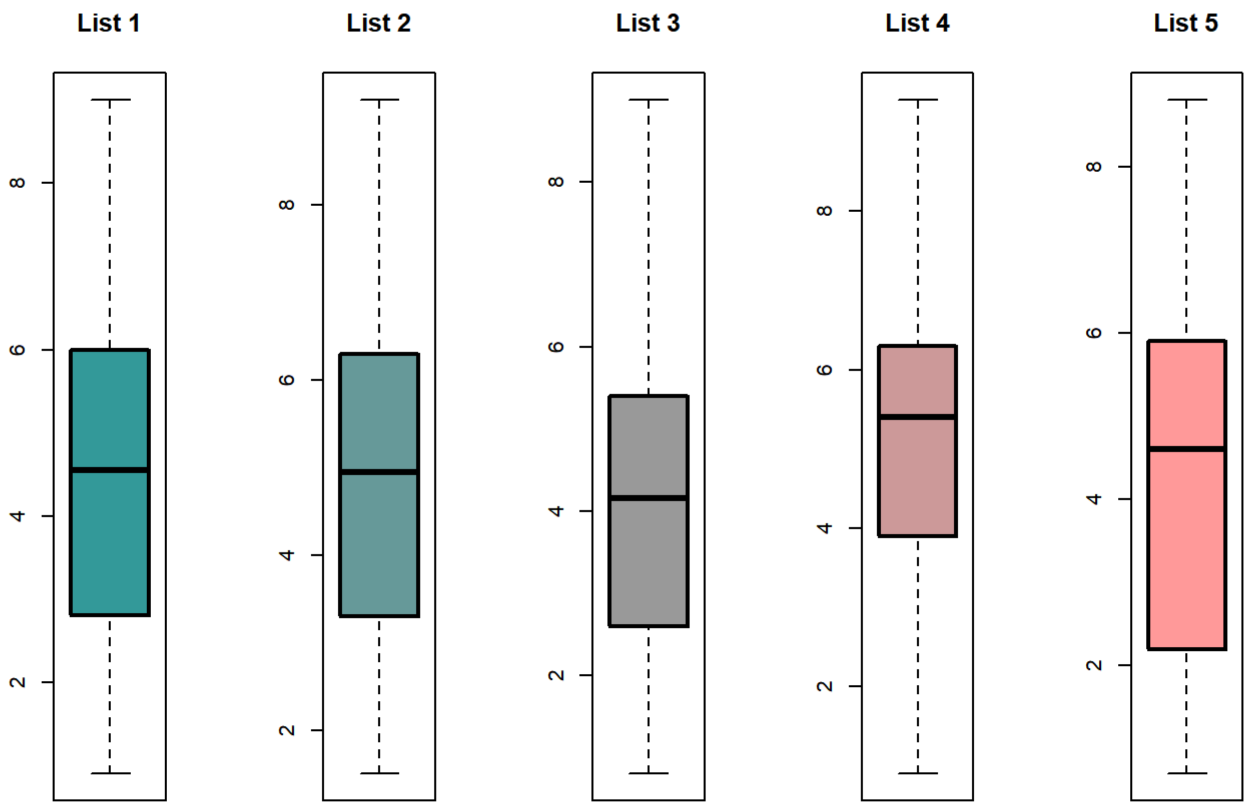
```
summary(SensoryItem)
```

```
##      Item1           Item2           Item3           Item4     
##  Min.   :3.300   Min.   :4.200   Min.   :1.300   Min.   :5.90  
##  1st Qu.:4.050   1st Qu.:4.700   1st Qu.:2.350   1st Qu.:6.40  
##  Median :4.400   Median :5.300   Median :2.600   Median :6.90  
##  Mean   :4.467   Mean   :5.313   Mean   :2.773   Mean   :6.88  
##  3rd Qu.:5.100   3rd Qu.:5.950   3rd Qu.:3.050   3rd Qu.:7.20  
##  Max.   :5.700   Max.   :6.300   Max.   :4.600   Max.   :8.20  
##      Item5          Item6           Item7           Item8      
##  Min.   :4.90   Min.   :1.100   Min.   :0.700   Min.   :3.000  
##  1st Qu.:5.70   1st Qu.:1.750   1st Qu.:1.000   1st Qu.:4.400  
##  Median :5.90   Median :2.100   Median :1.200   Median :4.600  
##  Mean   :5.92   Mean   :2.393   Mean   :1.407   Mean   :4.427  
##  3rd Qu.:6.15   3rd Qu.:3.150   3rd Qu.:1.550   3rd Qu.:4.800  
##  Max.   :7.00   Max.   :4.000   Max.   :3.100   Max.   :4.900  
##      Item9           Item10    
##  Min.   :6.700   Min.   :2.80  
##  1st Qu.:7.950   1st Qu.:3.90  
##  Median :8.800   Median :4.80  
##  Mean   :8.467   Mean   :4.52  
##  3rd Qu.:9.000   3rd Qu.:5.10  
##  Max.   :9.400   Max.   :5.50  
```

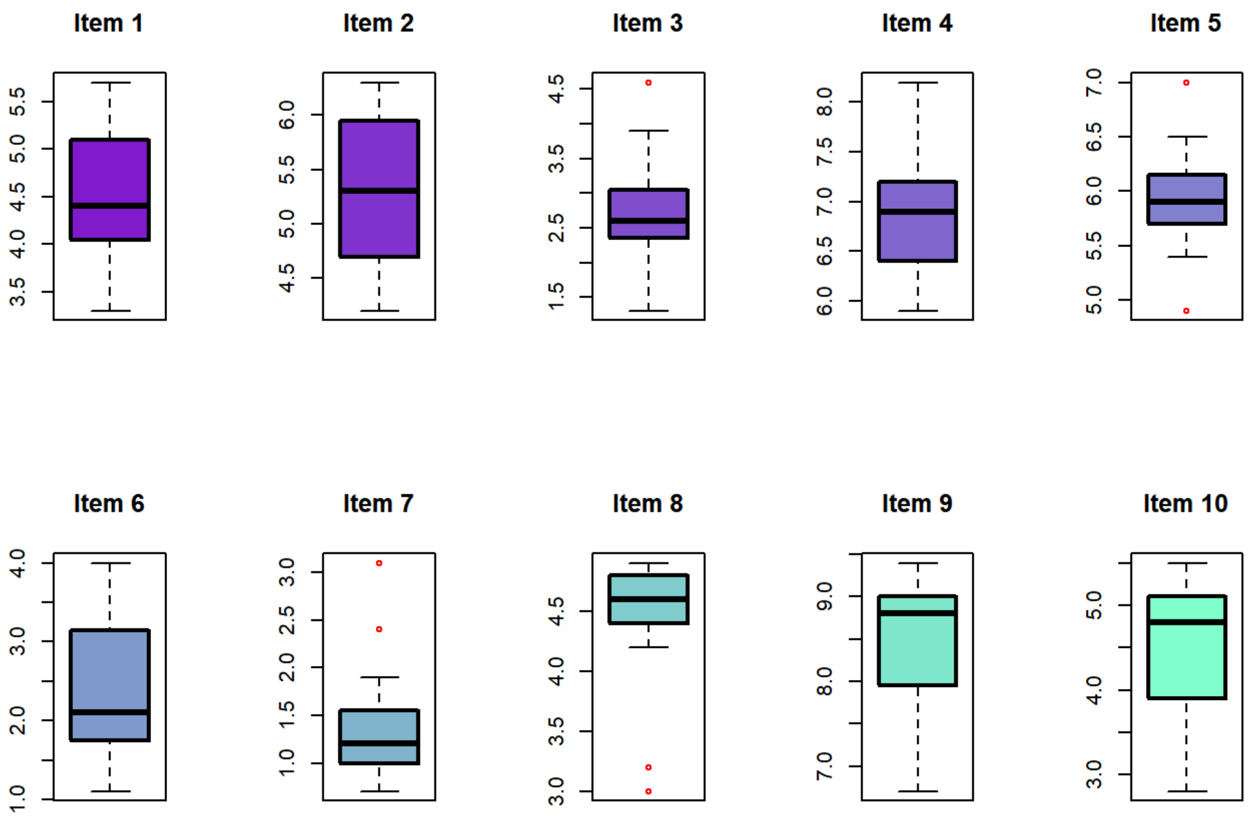
```
#And we can also find the summary of each item as well
```

We can see the distribution of each list. Although there are some differences between plots, the distributions don’t differ a lot.

```
par(mfcol=c(1,5)) #We will put 5 plots on one window
for(i in 1:5){
boxplot(Sensory2[,i],boxlwd=2,boxwex=1.5,col=rgb(0.2*i,0.6,0.6),main=paste('List',i)) #We are making plots for each variable
}
```



```
#This is the boxplot of each variable. We can see how the data is distributed
par(mfrow=c(2,5)) #We will put 10 plots for each plot of an item
for(i in 1:10){
boxplot(SensoryItem[,i],boxlwd=2,boxwex=1.5,outcol='red',col=rgb(0.5,0.1*i,0.8),main=paste('Ite
m',i)) #Making plots for each item
}
```



```
#This is the boxplot of each item. We can see how the data is distributed
```

We can see the distribution by each item. We can see that there are some differences between plots; values of Item 8 are usually bigger than other items. On the other hand, values of Item 7 are usually smaller than other items.

Long Jump Data

```
k<-readLines('LongJumpData.dat.txt')
```

```
## Warning in readLines("LongJumpData.dat.txt"): 'LongJumpData.dat.txt'에서 불
## 완전한 마지막 행이 발견되었습니다
```

```
#We will get the text file and read by lines
l<-character()
#Making an empty vector
k<-k[-1]
#We will skip the first line that we got from readling text file
for(i in 1:6){
kw<-word(k[i],1:10) #Extracting all the words in each line
kw<-kw[!is.na(kw)==TRUE] #If nothing was extracted, we won't pull that data
l<-c(l,kw) #Adding the values from previous steps to assigned vector
}
length(l) #Number of observations
```

```
## [1] 44
```

```
idx1<-seq(1,44,by=2) #Odd numbers from 1 to 44
idx2<-seq(2,44,by=2) #Even numbers from 2 to 44
Year<-l[idx1] #Assign odd number order obersvations to variable 'Year'
Long_Jump<-l[idx2] #Assign even number order obersvations to variable 'Long_Jump'
LongJumpData<-data.frame(Year,Long_Jump) #Make Year and Long_Jump variable into data frame
```

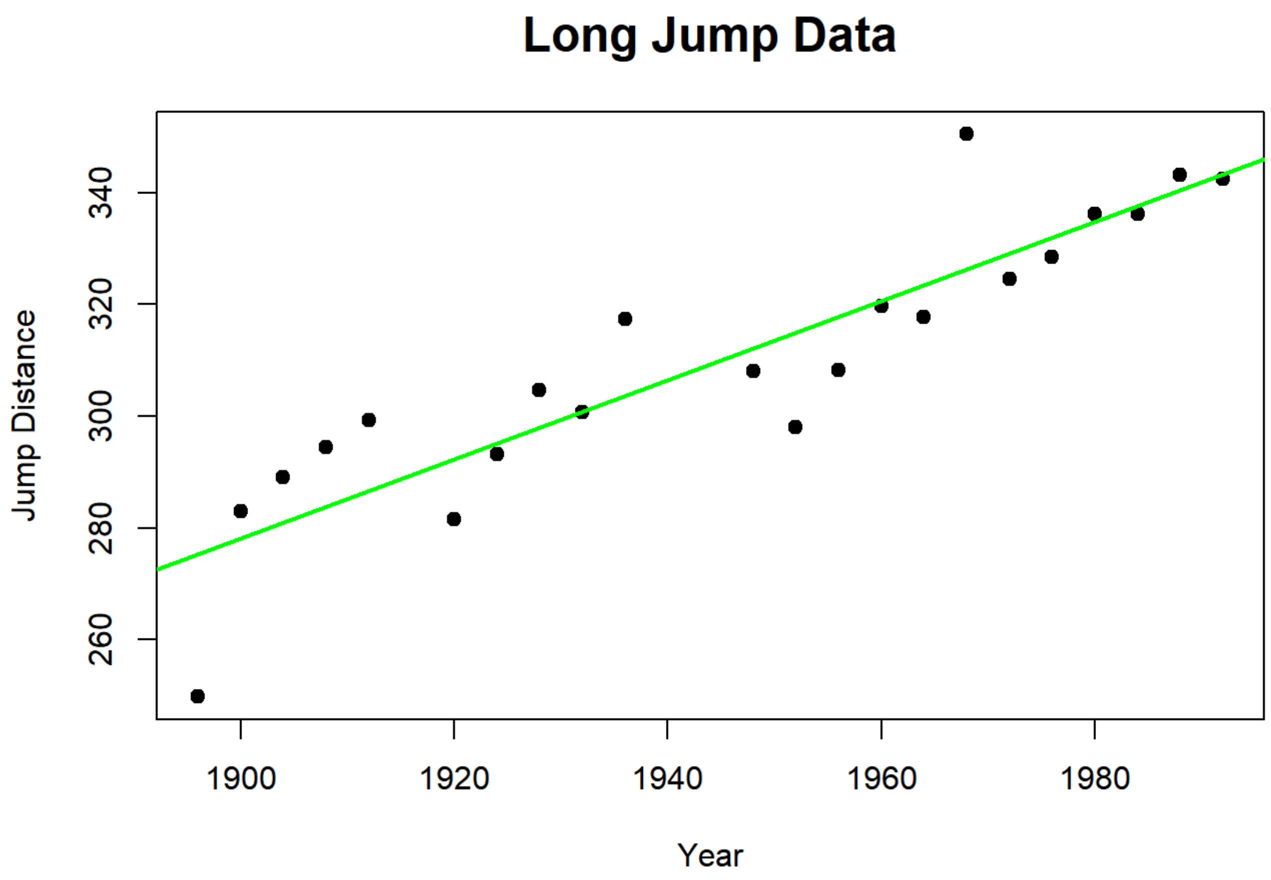
Through the steps above, we are able to import data to R

```
LongJumpData$Year<-as.numeric(as.character(LongJumpData$Year))
#Changing the type of variable from factor to numeric
LongJumpData$Year<-LongJumpData$Year+1900
#Added 1900 since the vector is centered in 1900
LongJumpData$Long_Jump<-as.numeric(as.character(LongJumpData$Long_Jump))
#Changing the type of variable from factor to numeric
summary(LongJumpData)
```

```
##       Year        Long_Jump
##  Min.   :1896   Min.   :249.8
##  1st Qu.:1921   1st Qu.:295.4
##  Median :1950   Median :308.1
##  Mean   :1945   Mean   :310.3
##  3rd Qu.:1971   3rd Qu.:327.5
##  Max.   :1992   Max.   :350.5
```

Above is the summary of Long Jump Data. We can see how two variables are distributed. We can also find how two variables are related through scatterplot and a regression line.

```
plot(LongJumpData$Year,LongJumpData$Long_Jump,xlab='Year',ylab='Jump Distance',main='Long Jump Data',
pch=19,cex.main=1.5) #Making a scatterplot. The y-variable will the the distance of jump and x-axis will be year.
abline(lm(LongJumpData$Long_Jump~LongJumpData$Year),col='green',lwd=2) #Making a regression line. lm is a function for making a regression line, and abline will draw the line using the coefficients we got from lm function.
```



we can see that the regression line is made in increasing direction, which is, as time goes by the distance of jump has increased.

Brain and Body Data

We can use the text file to read the data.

```
k<-readLines('BrainandBodyWeight.dat.txt')
#Read every line in text file.
k<-k[-1]
#Remove the first line we read, which is the names of variable
l<-numeric()
#Make an empty numeric vector
for(i in 1:22){
kw<-as.numeric(word(k[i],1:10,sep=' ')) #Extract every word in the line
kw<-kw[is.na(kw)==FALSE]
l<-c(l,kw) #Put the words extracted into a vector
}
length(l) #Number of observations
```

```
## [1] 124
```

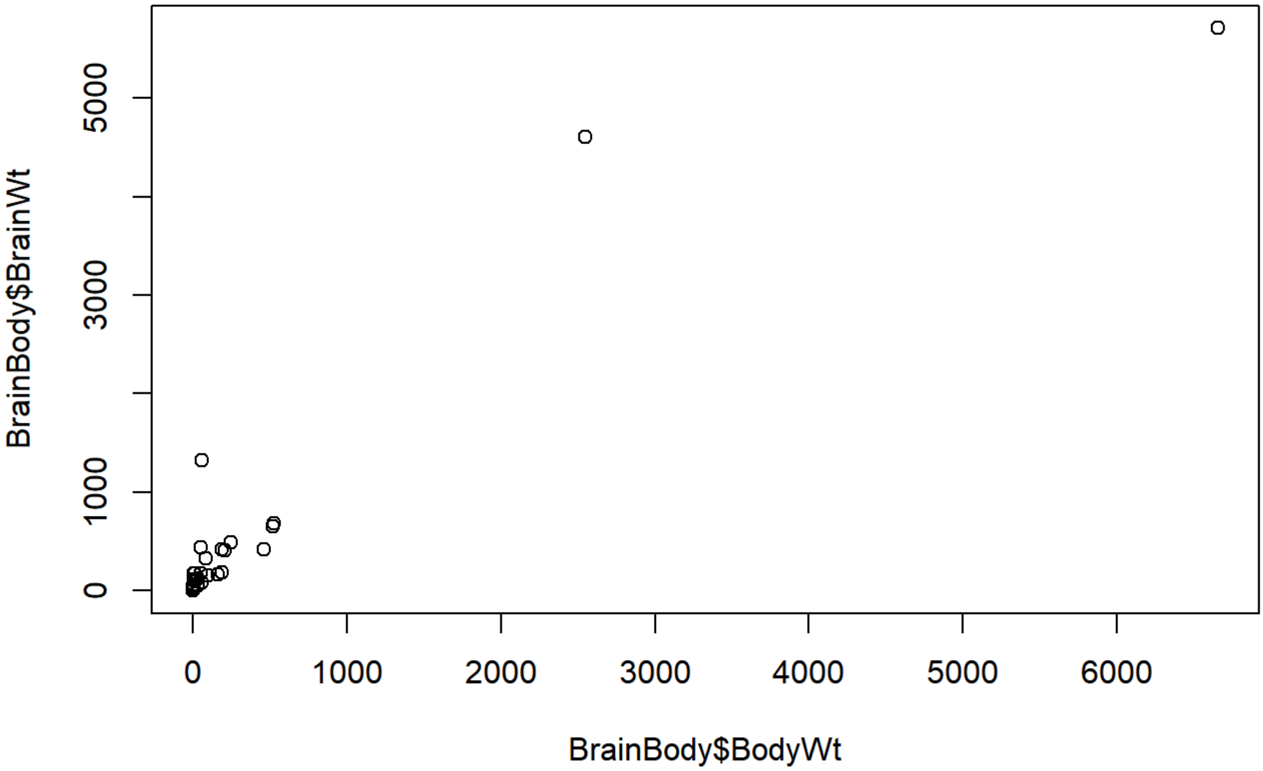
```
idx1<-seq(1,124,by=2) #Getting odd numbers from 1 to 124
idx2<-seq(2,124,by=2) #Getting even numbers from 2 to 124
BrainWt<-l[idx2] #The values in odd number order will be Brain weight
BodyWt<-l[idx1] #The values in even number order will be Body weight
BrainBody<-data.frame(BodyWt,BrainWt) #Make two variables into a data frame
```

Through the steps above, we are able to make a data frame. We can get the summary of each variable and relation through this.

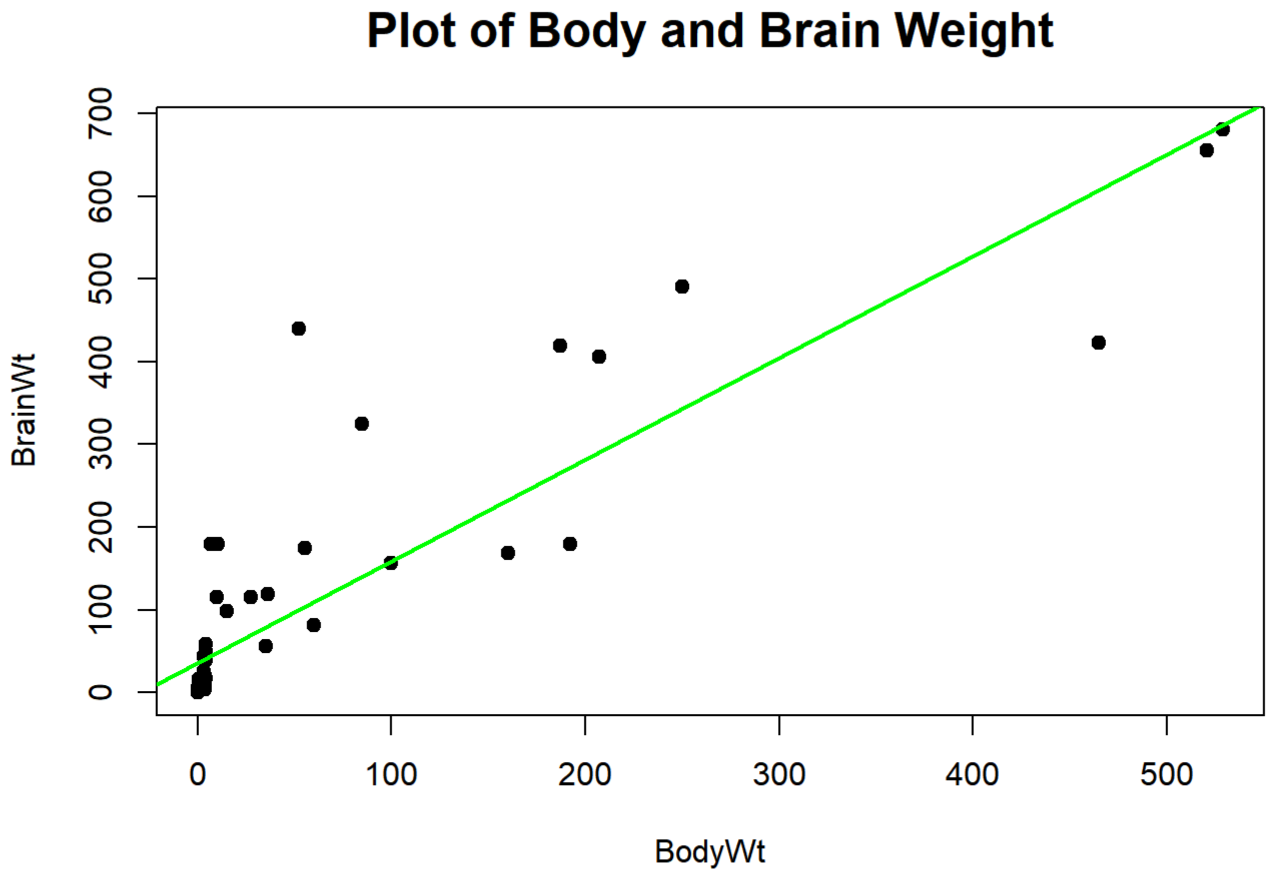
```
summary(BrainBody)
```

```
##      BodyWt             BrainWt       
##  Min.   :   0.005   Min.   :   0.10  
##  1st Qu.:   0.600   1st Qu.:   4.25  
##  Median :   3.342   Median :  17.25  
##  Mean   : 198.790   Mean   : 283.13  
##  3rd Qu.:  48.203   3rd Qu.: 166.00  
##  Max.   :6654.000   Max.   :5712.00  
```

```
#This is the summary of two variables; brain weight and body weight
plot(BrainBody$BodyWt,BrainBody$BrainWt)
```



```
#There are some extreme values. We can remove them and plot it again.
BrainBody2<-BrainBody
#Duplicate the data frame
BrainBody2<-BrainBody2[which(BrainBody$BodyWt<1000&BrainBody$BrainWt<1000),]
#The dupliated data frame will only have values that BodyWt and BrainWt variables are both less
than 1000.
plot(BrainBody2,pch=19,main='Plot of Body and Brain Weight',cex.main=1.5)
abline(lm(data=BrainBody2,BrainWt~BodyWt),col='green',lwd=2)
```



```
#The scatter plot for two variables in modified data frame. We can also draw a regression line
 over the scatterplot.
```

From the data above, we can conclude that the brain weight and body weight are postively correlated. Also, since the tangent of regression line is positive, we can learn that the brain weight will increase when body weight increases.

Tomato data

Since the data is not cleaned but has only a few observations, we will type the data to get the variables and values

```
k<-readLines('tomato.dat.txt')
#Read every line in tomato.dat text file
k
```

```
## [1] "#this needs reformatting to read into Splus"
## [2] "                      10000             20000     30000"
## [3] "Ife₩₩#1           16.1,15.3,17.5   16.6,19.2,18.5   20.8,18.0,21.0"
## [4] "PusaEarlyDwarf   8.1,8.6,10.1,    12.7,13.7,11.5   14.4,15.4,13.7 "
```

```
#Read the values. The data is messy but only has a few observations
V1<-c(16.1,15.3,17.5,8.1,8.6,10.1)
V2<-c(16.6,19.2,18.5,12.7,13.7,11.5)
V3<-c(20.8,18.0,12.0,14.4,15.4,13.7)
#Enter values to make a variable.
tomato<-data.frame(V1,V2,V3)
#Make 3 variables above into a data frame
colnames(tomato)<-c('10k','20k','30k')
#The variable names will be 10k, 20k and 30k respectively
Ife<-paste('Ife#1',1:3,sep='')
Pursa<-paste('PursaEarlyDwarf',1:3,sep='')
#We can also make row names for the data frame. Each will be Ife1, Ife2, Ife3, PursaEarlyDwarf
1,PursaEarlyDwarf2, and PursaEarlyDwarf3
rownames(tomato)<-c(Ife,Pursa)
#Put rownames for the data
tomato
```

	10k <dbl>	20k <dbl>	30k <dbl>
Ife#11	16.1	16.6	20.8
Ife#12	15.3	19.2	18.0
Ife#13	17.5	18.5	12.0
PursaEarlyDwarf1	8.1	12.7	14.4
PursaEarlyDwarf2	8.6	13.7	15.4
PursaEarlyDwarf3	10.1	11.5	13.7

6 rows

```
#This is the data frame we obtained. Since there were multiple data on one cell, we will put th
is into different cell in data frame.
```

Through these steps we are able to write the tomato data file. For analysis, we can use the following syntax.

```
summary(tomato)
```

```
##       10k              20k             30k       
##  Min.   : 8.100   Min.   :11.50   Min.   :12.00  
##  1st Qu.: 8.975   1st Qu.:12.95   1st Qu.:13.88  
##  Median :12.700   Median :15.15   Median :14.90  
##  Mean   :12.617   Mean   :15.37   Mean   :15.72  
##  3rd Qu.:15.900   3rd Qu.:18.02   3rd Qu.:17.35  
##  Max.   :17.500   Max.   :19.20   Max.   :20.80  
```

```
#We can see the summary of each variable; 10k, 20k and 30k
```

However, we can also make this data frame that has variables for each tomato brand

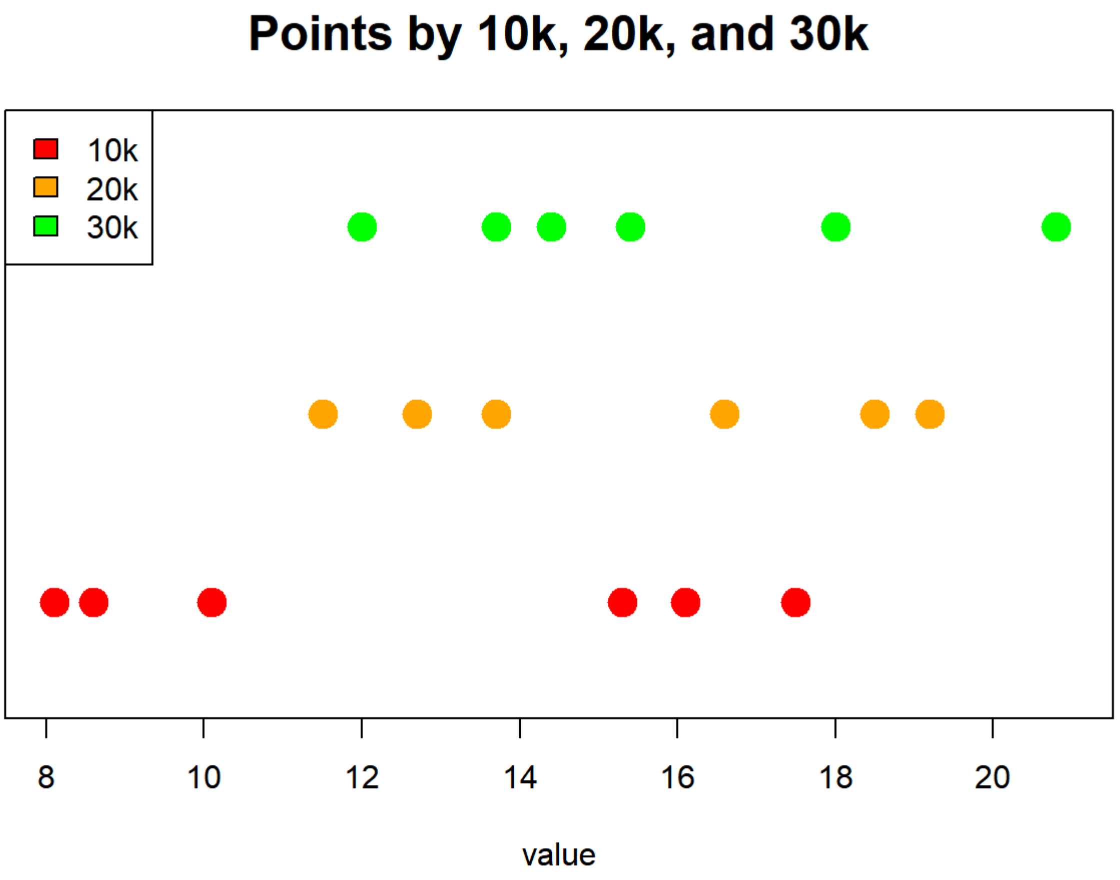
```
Ife<-as.vector(as.matrix(tomato[1:3,]))
Pursa<-as.vector(as.matrix(tomato[4:6,]))
#Assign values for tomato brands variables
summary(Ife) #Summary of tomato brand 'Ife'
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   16.10   17.50   17.11   18.50   20.80
```

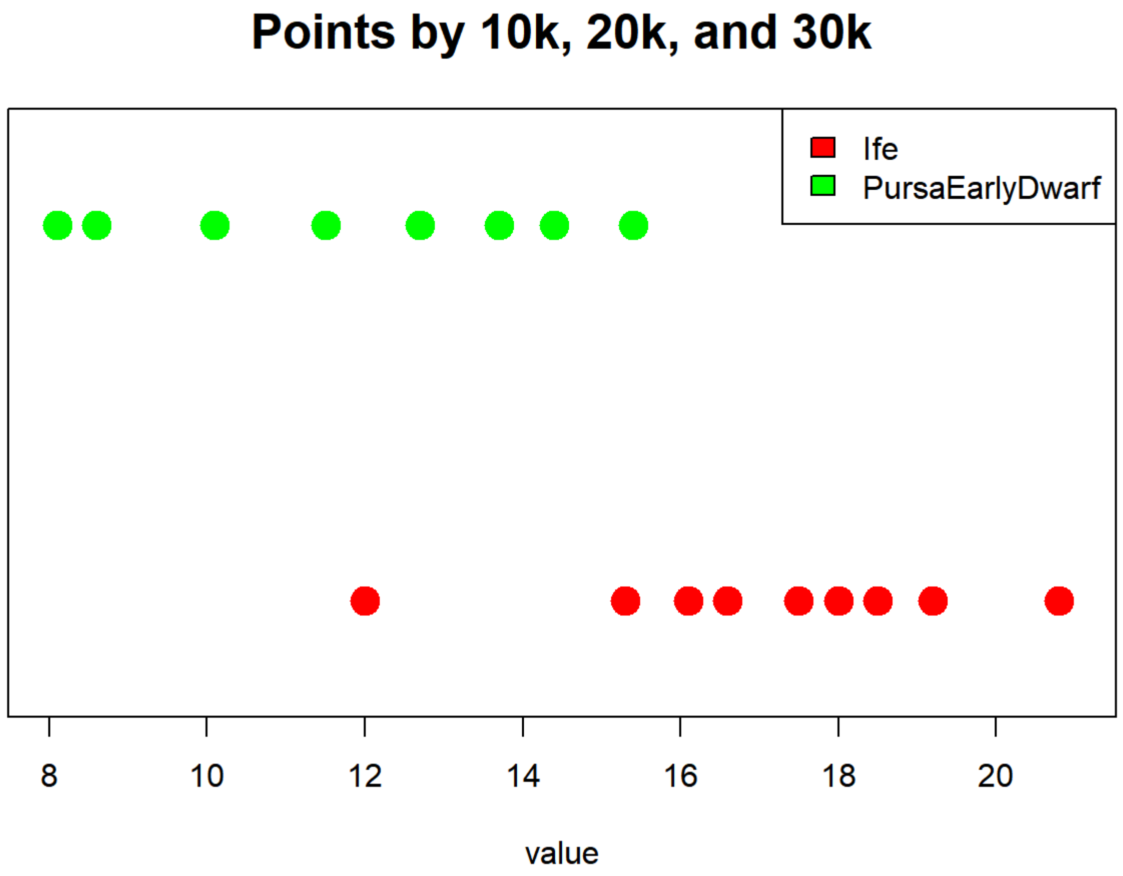
```
summary(Pursa) #Summary of tomato brand 'PursaEarlyDwarf'
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.10   10.10   12.70   12.02   13.70   15.40
```

```
plot(tomato[,1],rep(-1,6),ylim=c(-1.5,1.5),col='red',cex=2,pch=19,yaxt='n',xlim=c(8,21),
main='Points by 10k, 20k, and 30k',cex.main=1.5,ylab='',xlab='value')
points(tomato[,2],rep(0,6),ylim=c(-1.5,1.5),col='orange',cex=2,pch=19) #Make a plot for the first variable, 10k
points(tomato[,3],rep(1,6),ylim=c(-1.5,1.5),col='green',cex=2,pch=19) #Plot points of second variable on the existing plot
legend('topleft',fill=c('red','orange','green'),legend=c('10k','20k','30k')) #Plot points of thrid variable on the existing plot
```



```
#This is the plot of how the values by 3 factor 10k, 20k and 30k are distributed.
plot(Ife,rep(-1,9),ylim=c(-1.5,1.5),col='red',cex=2,pch=19,yaxt='n',xlim=c(8,21),
main='Points by 10k, 20k, and 30k',cex.main=1.5,ylab='',xlab='value') #Make a plot for Ife tomato brand
points(Pursa,rep(1,9),ylim=c(-1.5,1.5),col='green',cex=2,pch=19,yaxt='n') #Plot points from PursaEarlyDwarf tomato brand data
legend('topright',fill=c('red','green'),legend=c('Ife','PursaEarlyDwarf'))
```



```
#We can also make a plot of how the values by 2 tomato brands Ife, PursaEarlyDwarf are distributed
```

Problem 5

First we should read the raw data to look how the data looks like. To acheive this, we can try the following steps.

```
plants<-read.table('C:/Users/pc/Desktop/HWASOO/STUDY/StatPackage/plants.txt',header=T)
#We can read the text file using read.table function
summary(plants)
```

```
##                       Scientific_Name              Duration
##  Abelmoschus                 :   1   Perennial        :3031
##  Abelmoschus esculentus      :   1   Annual           : 682
##  Abies                       :   1   Annual, Perennial: 179
##  Abies balsamea              :   1   Annual, Biennial :  95
##  Abies balsamea var. balsamea:   1   Biennial         :  57
##  Abutilon                    :   1   (Other)          :  92
##  (Other)                     :5160   NA's             :1030
##            Active_Growth_Period      Foliage_Color      pH_Min
##  Spring and Summer   : 447      Dark Green  :  82   Min.   :3.000
##  Spring              : 144      Gray-Green  :  25   1st Qu.:4.500
##  Spring, Summer, Fall:  95      Green       : 692   Median :5.000
##  Summer              :  92      Red         :   4   Mean   :4.997
##  Summer and Fall     :  24      White-Gray  :   9   3rd Qu.:5.500
##  (Other)             :  30      Yellow-Green:  20   Max.   :7.000
##  NA's                :4334      NA's        :4334   NA's   :4327
##      pH_Max         Precip_Min      Precip_Max         Shade_Tolerance
##  Min.   : 5.100   Min.   : 4.00   Min.   : 16.00   Intermediate: 242
##  1st Qu.: 7.000   1st Qu.:16.75   1st Qu.: 55.00   Intolerant  : 349
##  Median : 7.300   Median :28.00   Median : 60.00   Tolerant    : 246
##  Mean   : 7.344   Mean   :25.57   Mean   : 58.73   NA's        :4329
##  3rd Qu.: 7.800   3rd Qu.:32.00   3rd Qu.: 60.00
##  Max.   :10.000   Max.   :60.00   Max.   :200.00
##  NA's   :4327     NA's   :4338    NA's   :4338
##    Temp_Min_F
##  Min.   :-79.00
##  1st Qu.:-38.00
##  Median :-33.00
##  Mean   :-22.53
##  3rd Qu.:-18.00
##  Max.   : 52.00
##  NA's   :4328
```

We can see there are many NAs in the data. In this case, we are trying to use 3 variables, which are pH_max, pH_min and Foliage_color. Therefore we will retrieve data that has no NAs in these variables to do the ANOVA test and make a scatterplot.

```
#Since we are looking for relation between pH and foliage color, we will get data which pH_Min
 and pH_Max are all available.
plants1<-plants[is.na(plants$pH_Min)==FALSE&is.na(plants$pH_Max)==FALSE,]
plants1$pHRange<-plants1$pH_Max-plants1$pH_Min
#Range of pH
```

Through these steps, we can first read the raw data and then get the data we need, which is, the data with pH variables with not NAs. We can check the modified data.

```
summary(plants1) #Summary of the modified data.
```

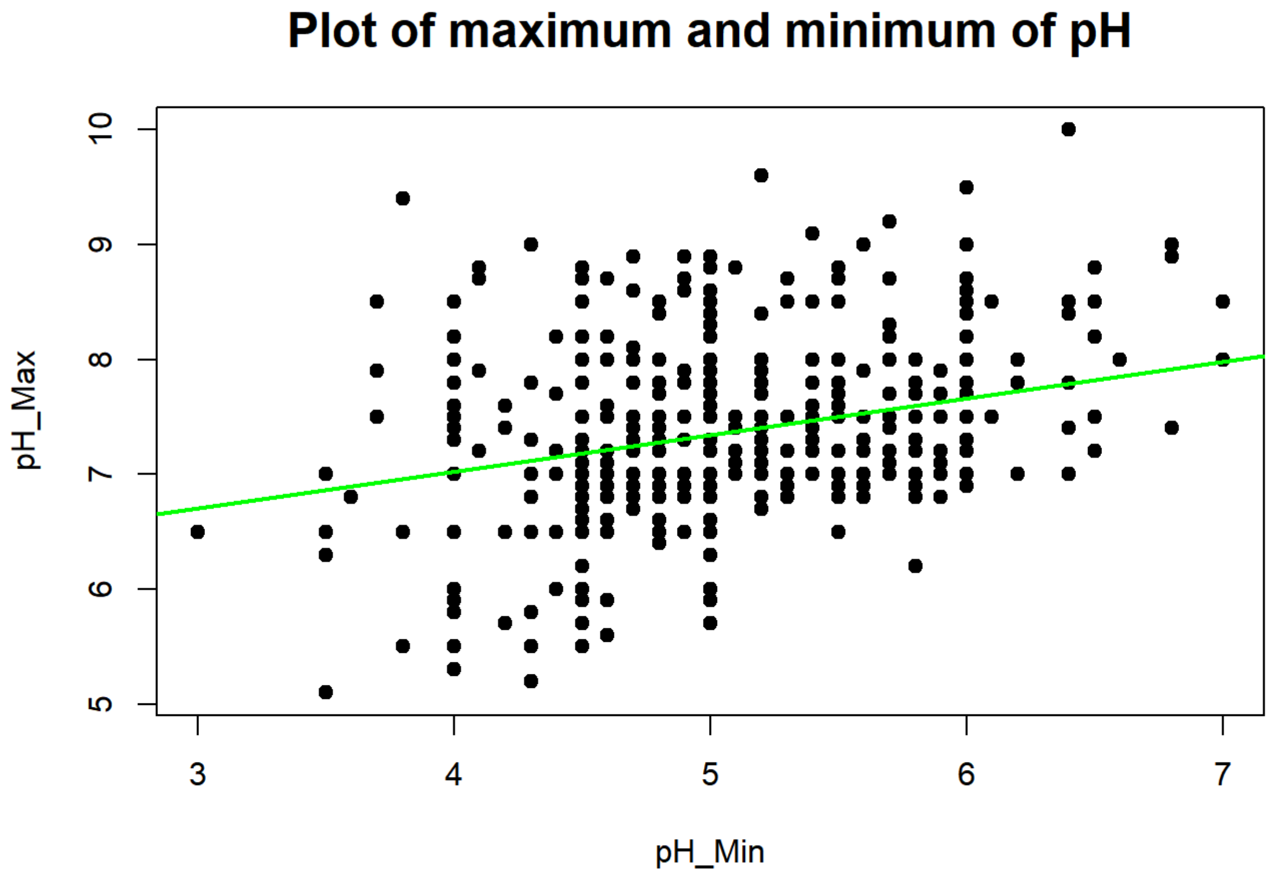
```
##            Scientific_Name                          Duration
##  Abies balsamea     :  1    Perennial                   :709
##  Acacia constricta :  1    Annual                      : 69
##  Acalypha virginica:  1    Annual, Perennial           : 36
##  Acer negundo      :  1    Annual, Biennial            :  8
##  Acer nigrum       :  1    Annual, Biennial, Perennial:  6
##  Acer pensylvanicum:  1    (Other)                     : 10
##  (Other)           :833    NA's                        :  1
##            Active_Growth_Period      Foliage_Color     pH_Min
##  Spring and Summer   :447       Dark Green  : 82   Min.   :3.000
##  Spring              :144       Gray-Green  : 25   1st Qu.:4.500
##  Spring, Summer, Fall: 95       Green       :692   Median :5.000
##  Summer              : 92       Red         :  4   Mean   :4.997
##  Summer and Fall     : 24       White-Gray  :  9   3rd Qu.:5.500
##  (Other)             : 30       Yellow-Green: 20   Max.   :7.000
##  NA's                :  7       NA's        :  7
##      pH_Max         Precip_Min      Precip_Max         Shade_Tolerance
##  Min.   : 5.100   Min.   : 4.00   Min.   : 16.00   Intermediate:242
##  1st Qu.: 7.000   1st Qu.:16.75   1st Qu.: 55.00   Intolerant  :349
##  Median : 7.300   Median :28.00   Median : 60.00   Tolerant    :246
##  Mean   : 7.344   Mean   :25.57   Mean   : 58.73   NA's        :  2
##  3rd Qu.: 7.800   3rd Qu.:32.00   3rd Qu.: 60.00
##  Max.   :10.000   Max.   :60.00   Max.   :200.00
##                   NA's   :11      NA's   :11
##    Temp_Min_F        pHRange
##  Min.   :-79.00   Min.   :0.400
##  1st Qu.:-38.00   1st Qu.:1.900
##  Median :-33.00   Median :2.200
##  Mean   :-22.53   Mean   :2.347
##  3rd Qu.:-18.00   3rd Qu.:2.900
##  Max.   : 52.00   Max.   :5.600
##  NA's   :1
```

Notice that there are no more NA values in pH variables and now the pH range variable is added to the data frame.We can also make a plot to check the relationship between minimum and maximum pH.

```
plot(plants1$pH_Min,plants1$pH_Max,pch=19,xlab='pH_Min',ylab='pH_Max',
main='Plot of maximum and minimum of pH',cex.main=1.5)
lm(data=plants1,pH_Max~pH_Min)
```

```
##
## Call:
## lm(formula = pH_Max ~ pH_Min, data = plants1)
##
## Coefficients:
## (Intercept)       pH_Min
##       5.755        0.318
```

```
#The pH_Min is the independent variable and pH_Max is the target variable
#The first value is the intercept, and second value is the tangent of the line
abline(lm(data=plants1,pH_Max~pH_Min),lwd=2,col='green')
```



```
summary(aov(data=plants1,pHRange~Foliage_Color))
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Foliage_Color   5   10.3   2.053   3.322 0.00561 **
## Residuals     826  510.5   0.618
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 7 observations deleted due to missingness
```

```
#Summary of ANOVA.
```

We can see that the degree of freedom of Foliage Color is 5, which means there are 6 classes in Foliage_Color. To use ANOVA, some assumptions are required; Variance among classes are the same. Since the p-value for this ANOVA test is smaller than 0.05, we can conclude that there are at least two classes of Foliage_Color that have different means of pH range under significance level $/alpha$=0.05.