**Data Science**

**Practical Journal**

Submitted BY

**Carol Richa Dsouza**

**Seat number : 31011121013**

Third Year Computer Science Honors

2023-2024

**DEPARTMENT OF COMPUTER SCIENCE HONORS**

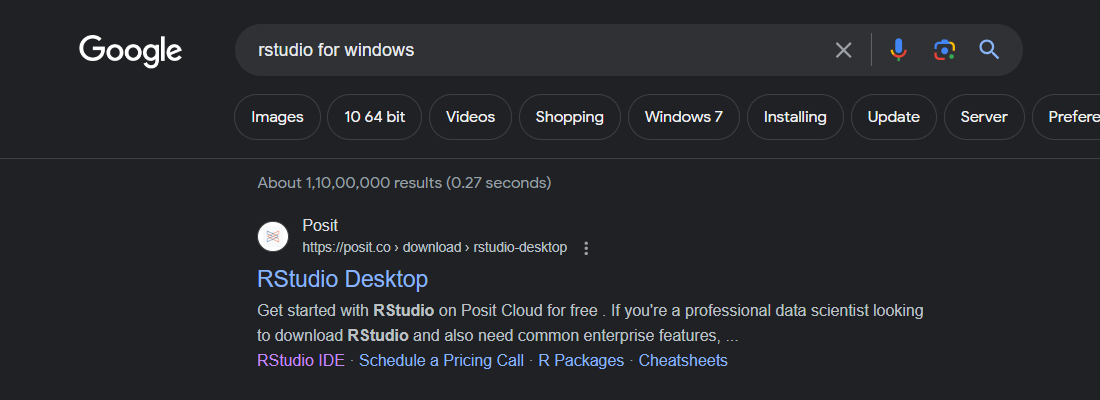
**Somaiya Vidyavihar University**

**Index**

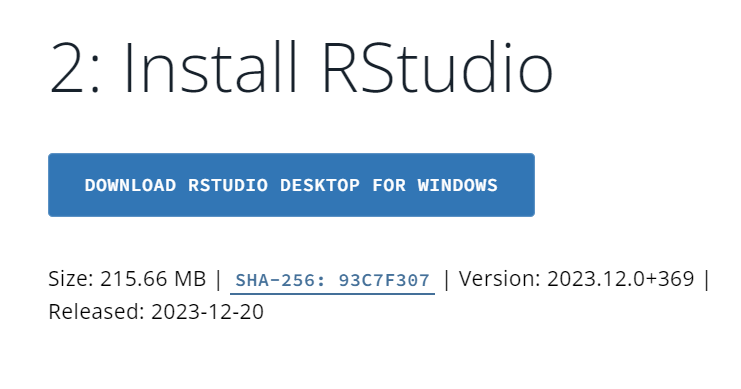
| **Sr No** | **Title** |
| --- | --- |
| **1** | Installation of R Studio and basic codes |
| **2** | Time Series Analysis |
| **3** | Time Series Frequency Analysis  i. Air Passengers  ii. Cars |
| **4** | Analysis of Variance  A. Warpbreaks dataset  B. Plant Growth dataset |
| **5** | Simple, Multiple Linear Regression |
| **6** | Hypothesis |
| **7** | Decision Tree |
| **8** | Logistic Regression |
| **9** | K-means Clustering |

**Practical 1   
Installation of R Studio and basic codes**

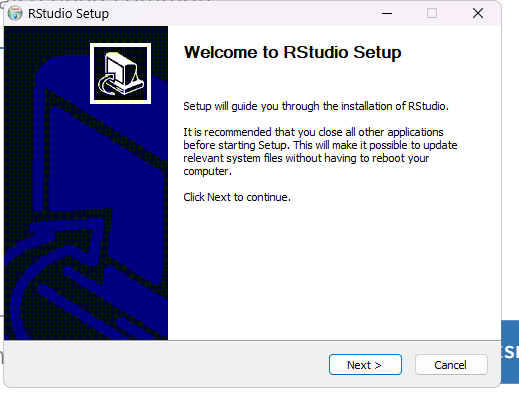
**Step 1 :** Open Google. Type “RStudio for Windows” and click the first link.



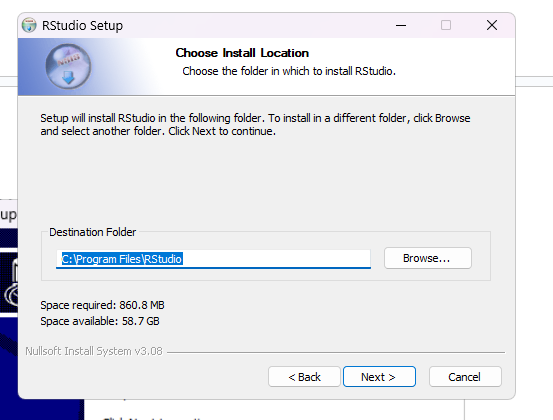
**Step 2 :** Click on RStudio Desktop for Windows



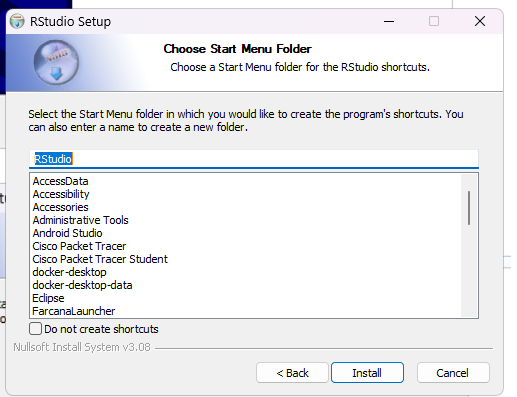
**Step 3 :** Setup the RStudio following the below steps. Click on “Next”



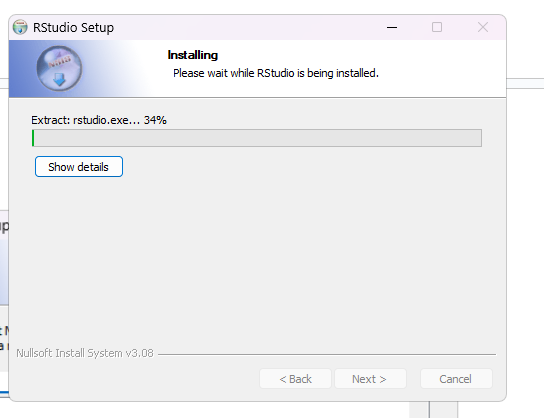
**Step 4 :** Choose the folder in which you want to install RStudio by clicking on Browse. Then click on “Next”



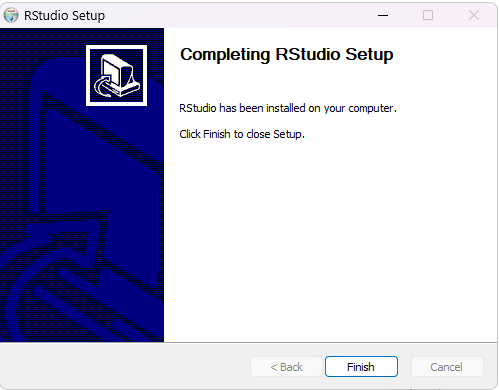
**Step 5 :** Select the Start Menu folder in which you want to create the program’s shortcuts and then click on “Install”



RStudio is installing

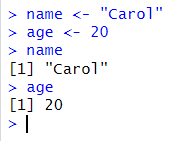


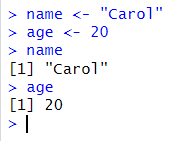
The RStudio setup has been completed. Click on “Finish”



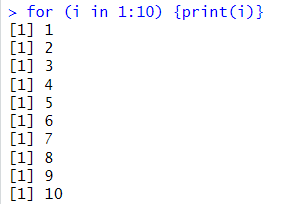
Basic Codes :

i. Variable declaration :

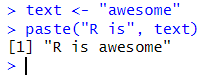




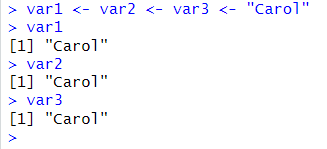
ii. Loops



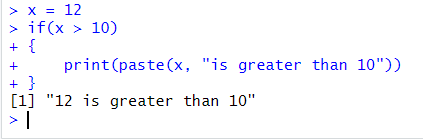
iii. paste



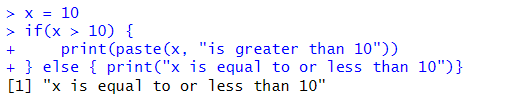
iv. Declaration of multiple variables



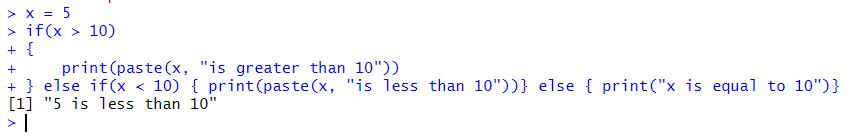
v. if statement



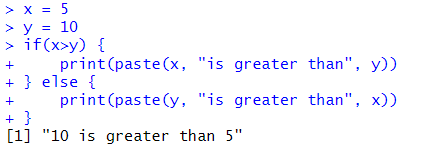
vi. if-else statement



vii. if-else if-else statement



viii. Compare 2 numbers



**Vector**

A vector is simply a list of items that are of the same type.

To combine the list of items to a vector, use the c() function and separate the items by a comma.

i. Vector of strings :



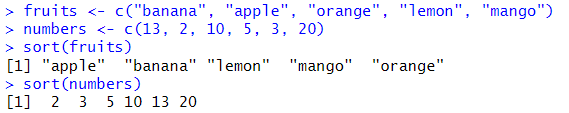
ii. Vector with numerical values in a sequence



iii. Find the length of the vector



iv. Sort vector strings and numbers



v. Access the first item in a vector



vi. To access multiple elements, refer the different index positions with the c() function



**Practical 2   
Time Series Analysis**

**1. Rainfall**

**Syntax :**

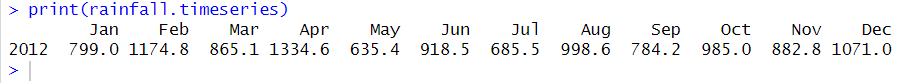
timeseries.object.name <- ts(data, start, end, frequency)

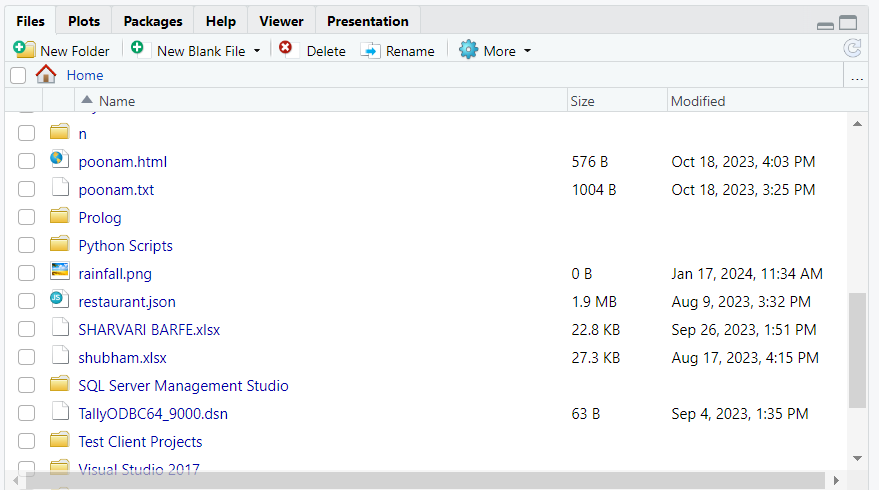
**Code :**

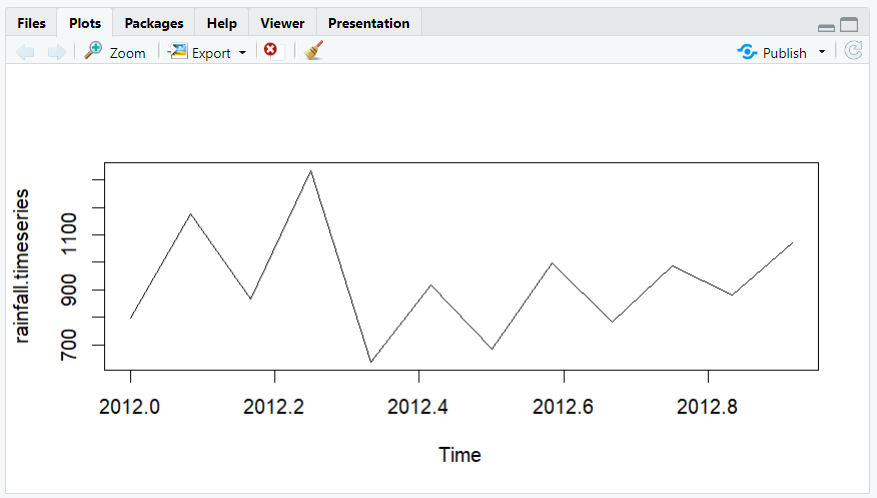
*> rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)*

*> rainfall.timeseries <- ts(rainfall, start=c(2012,1),frequency = 12)*

*> print(rainfall.timeseries)*

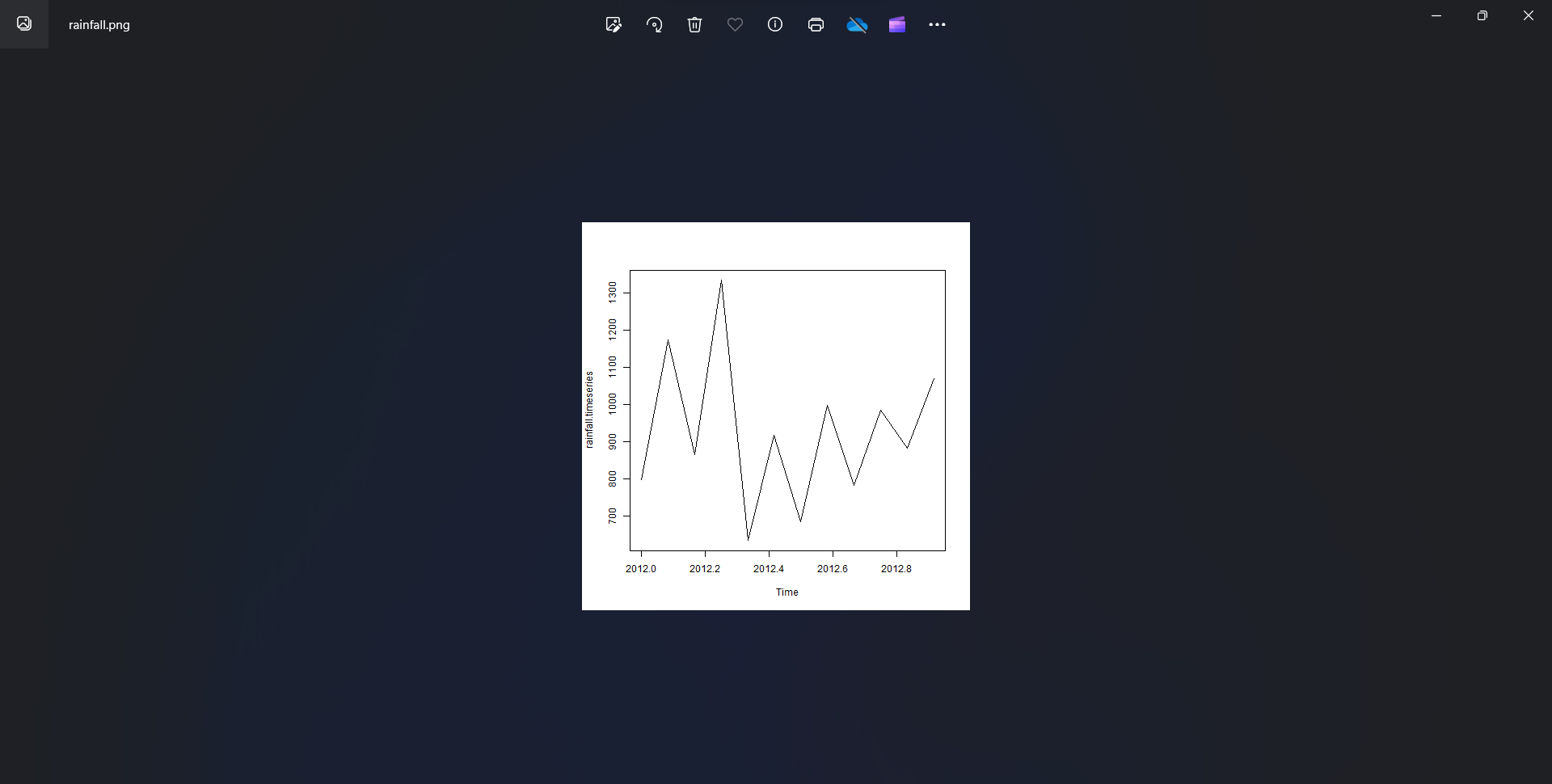


*> png(file = "rainfall.png")*

*> plot(rainfall.timeseries)*

*> dev.off()*





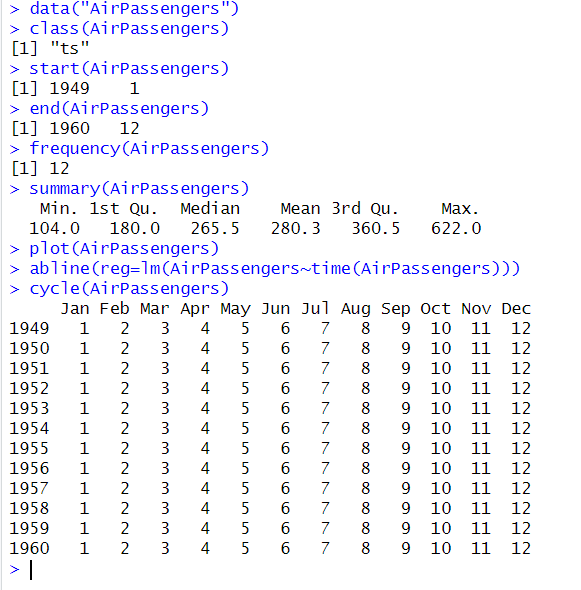
**Practical 3   
Time Series Frequency Analysis**

**1. AirPassengers**

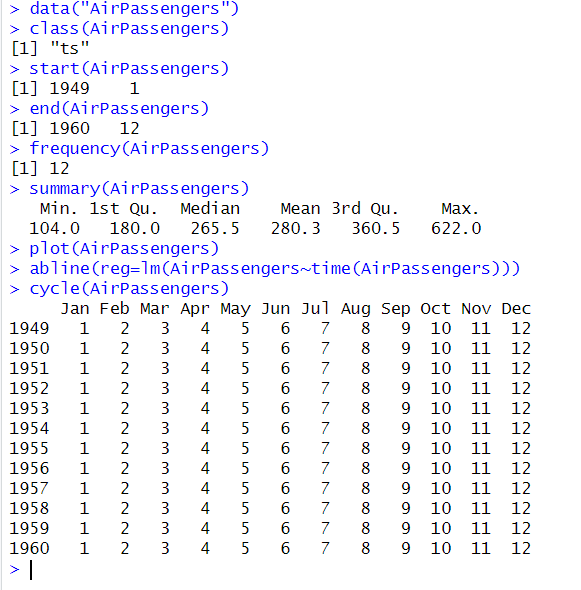
**Code :**

*> data(“AirPassengers”)*

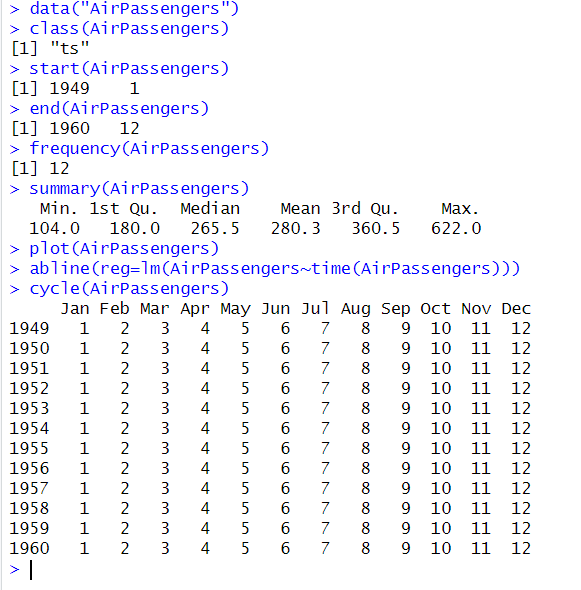
*> class(AirPassengers)*

**

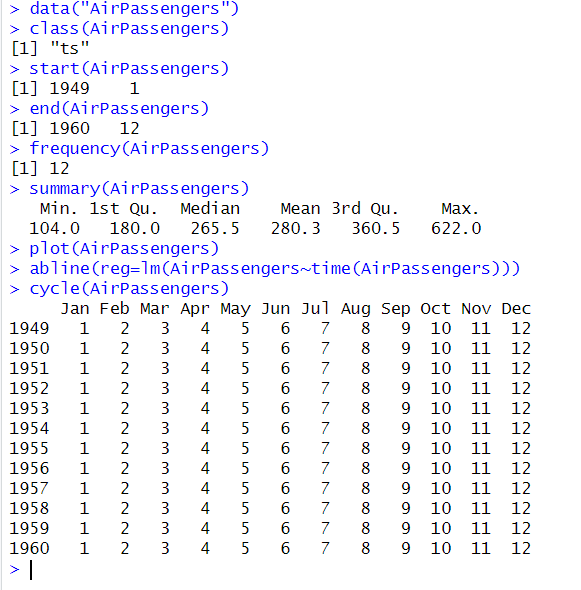
*> start(AirPassengers)*

**

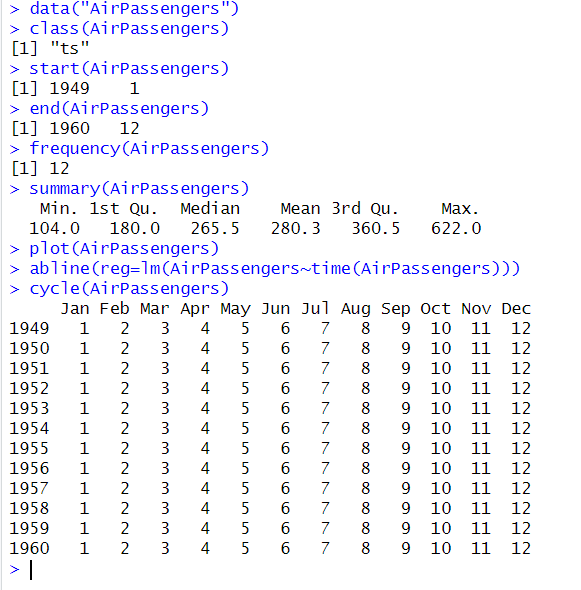
*> end(AirPassengers)*

**

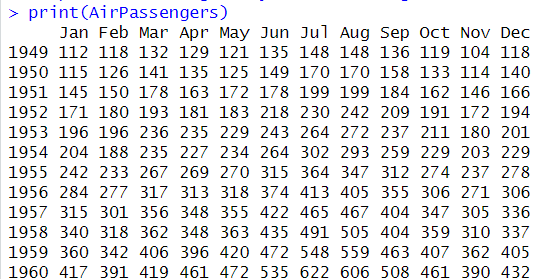
*> frequency(AirPassengers)*

**

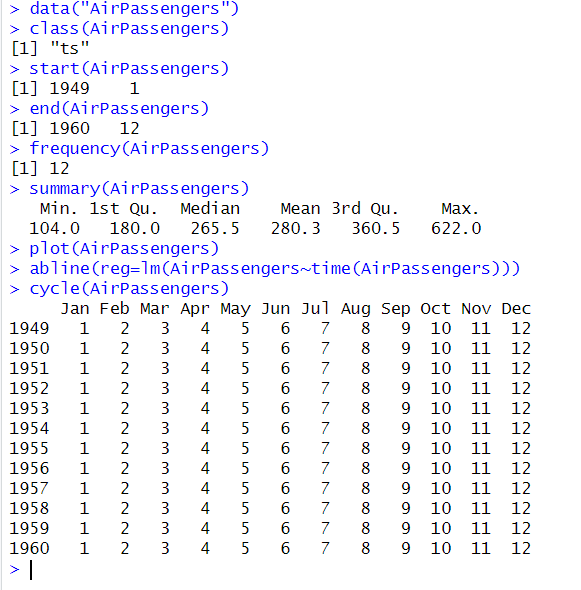
*> summary(AirPassengers)*

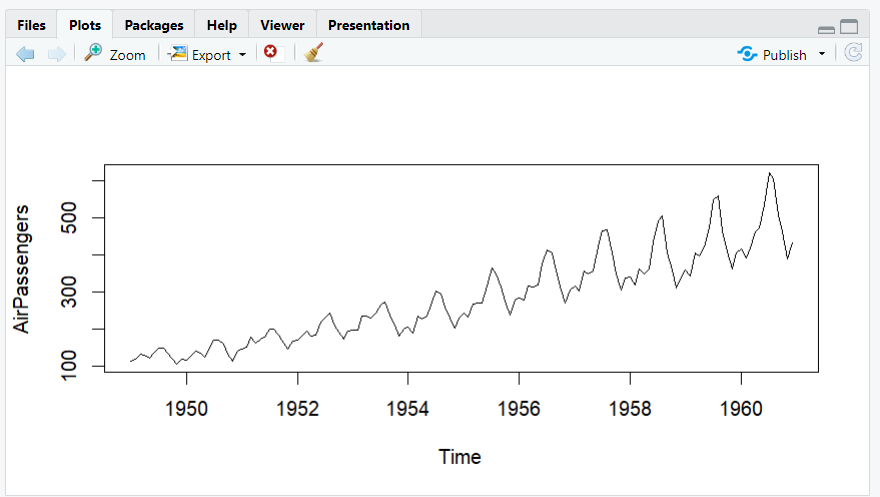
**

*> print(AirPassengers)*

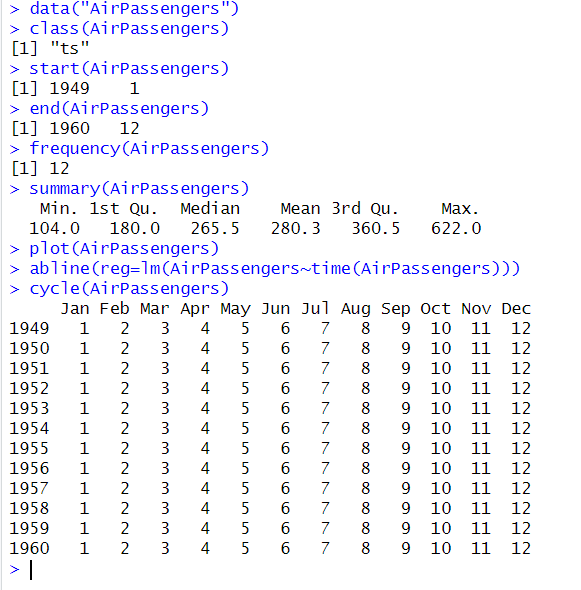
**

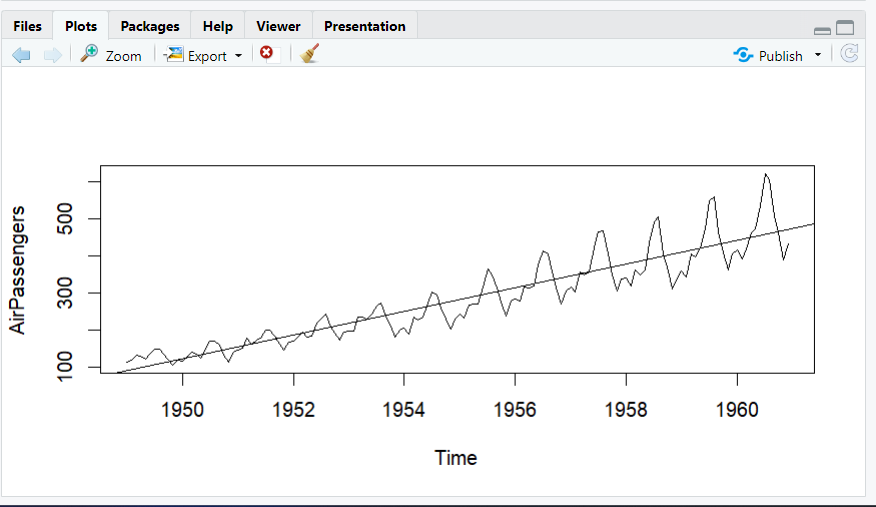
*> plot(AirPassengers)*

**

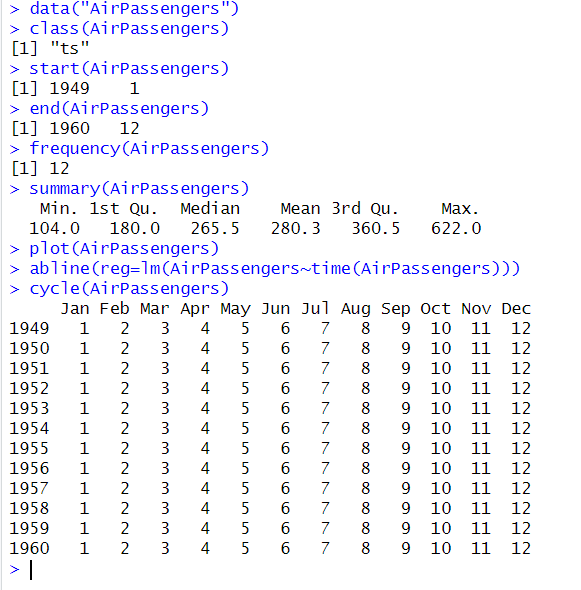
**

*> abline(reg=lm(AirPassengers~time(AirPassengers)))*

**

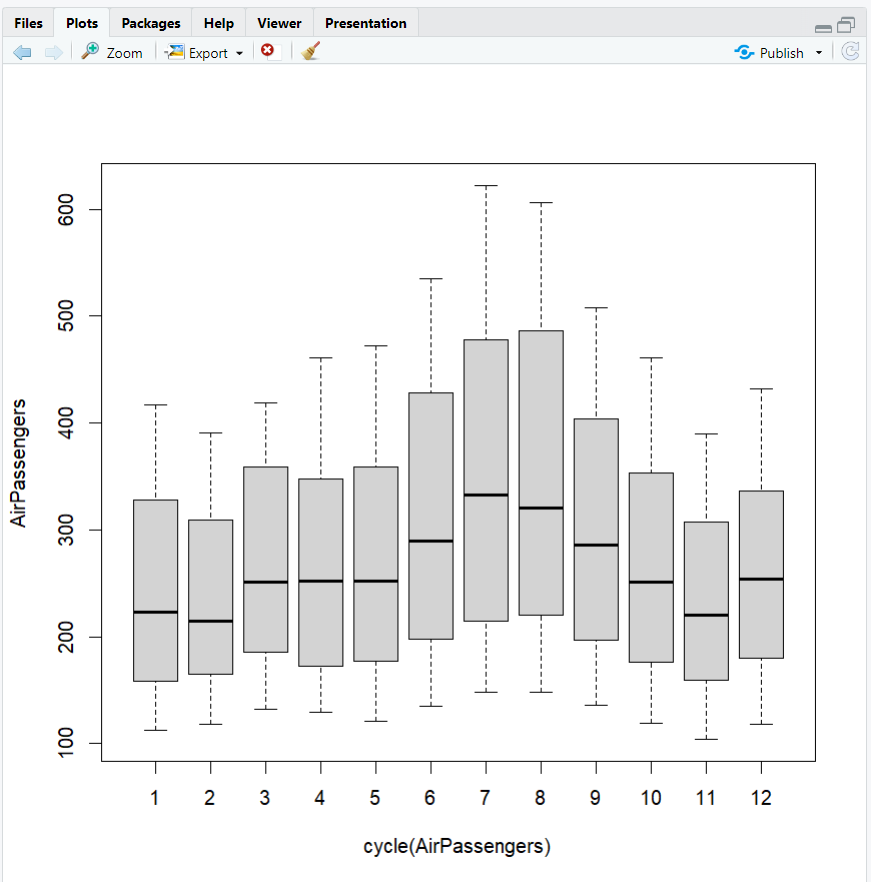
******

*> cycle(AirPassengers)*

**

*> boxplot(AirPassengers~cycle(AirPassengers))*



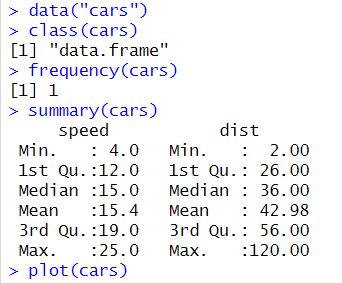


**2. Cars**

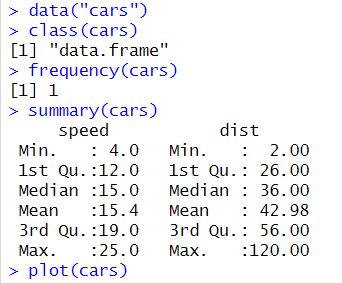
**Code :**

*> data("cars")*

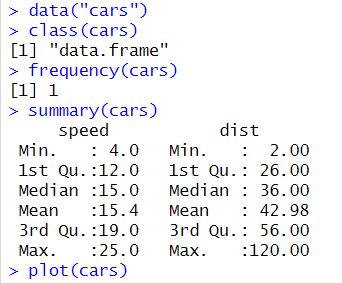
*> class(cars)*

**

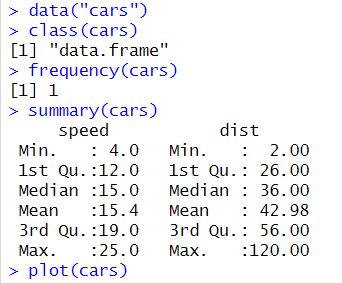
*> frequency(cars)*

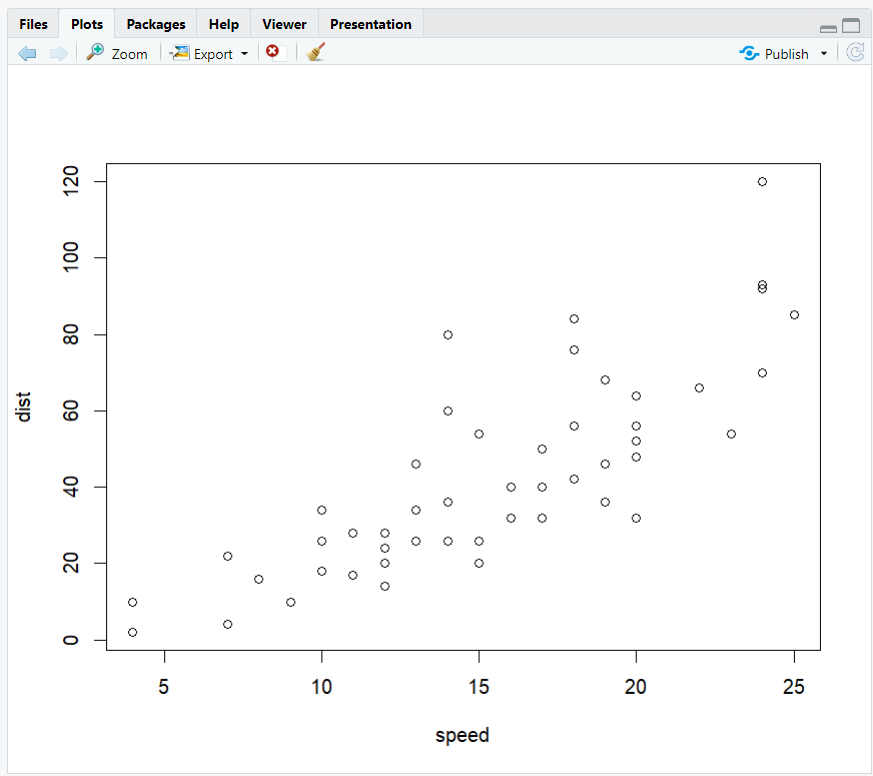
**

*> summary(cars)*

**

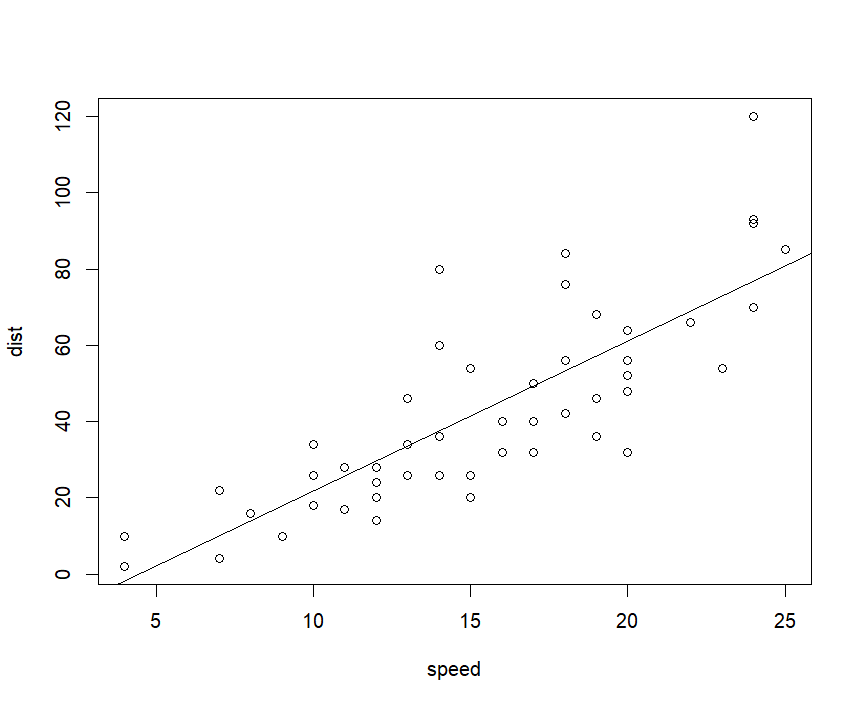
*> plot(cars)*

**

**

*> abline(reg <- lm(dist~speed, data = cars))*

****

****

**Practical 4**

**Analysis of Variance**

ANOVA, which stands for Analysis of Variance, is a statistical test used to analyze the difference between the means of more than two groups. A one-way ANOVA uses one independent variable, while a two-way ANOVA uses two independent variables.

ANOVA test the following hypothesis :

**Null hypothesis :** The means of all the (population) groups are equal

**Alternative hypothesis :** At least one mean is different

In notational form :

H0 : μ1 = μ2 = …. = μk

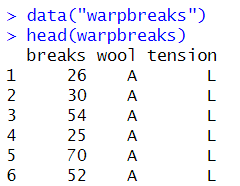
Ha : At least one mean is different

**A. Warpbreaks dataset**

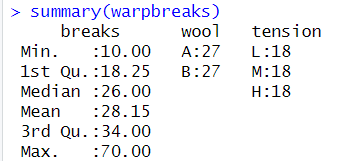
**Code :**

*> data("warpbreaks")*

*> head(warpbreaks)*

**

*> summary(warpbreaks)*

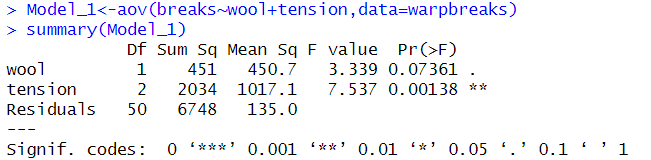


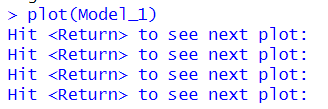
**Model 1 code :**

*> Model\_1<-aov(breaks~wool+tension,data=warpbreaks)*

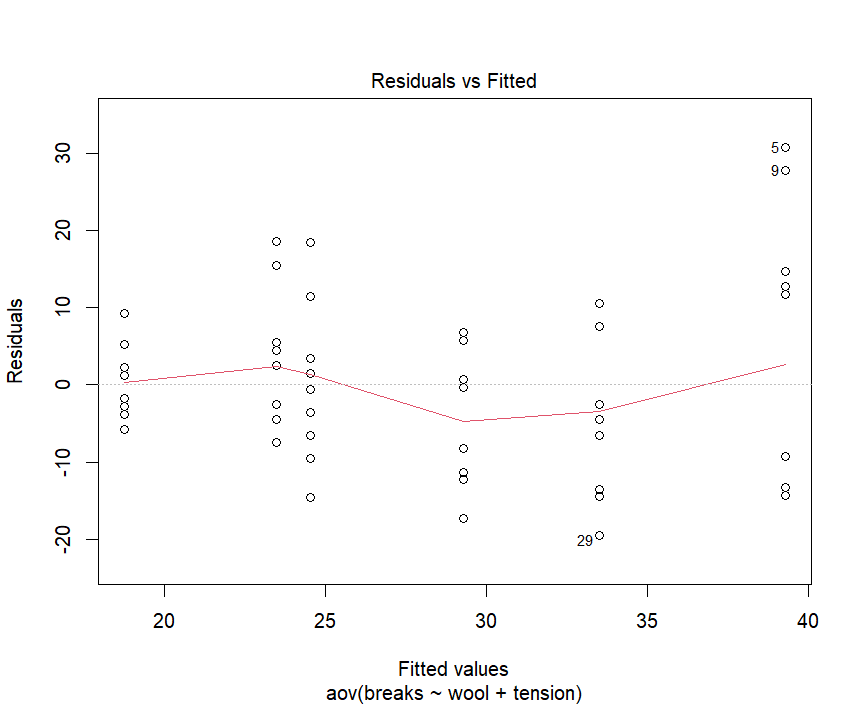
*> summary(Model\_1)*

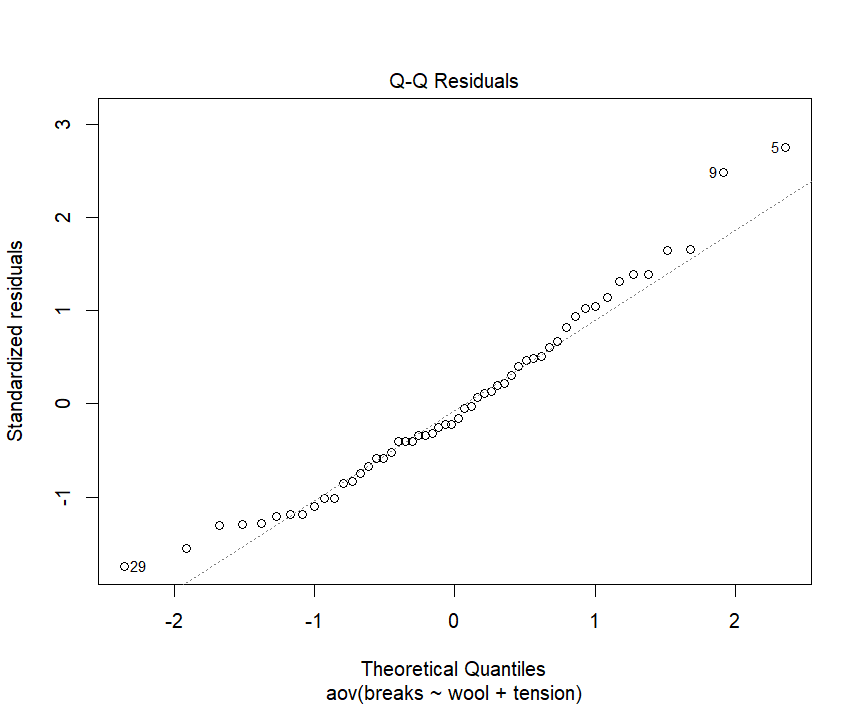
*> plot(Model\_1)*

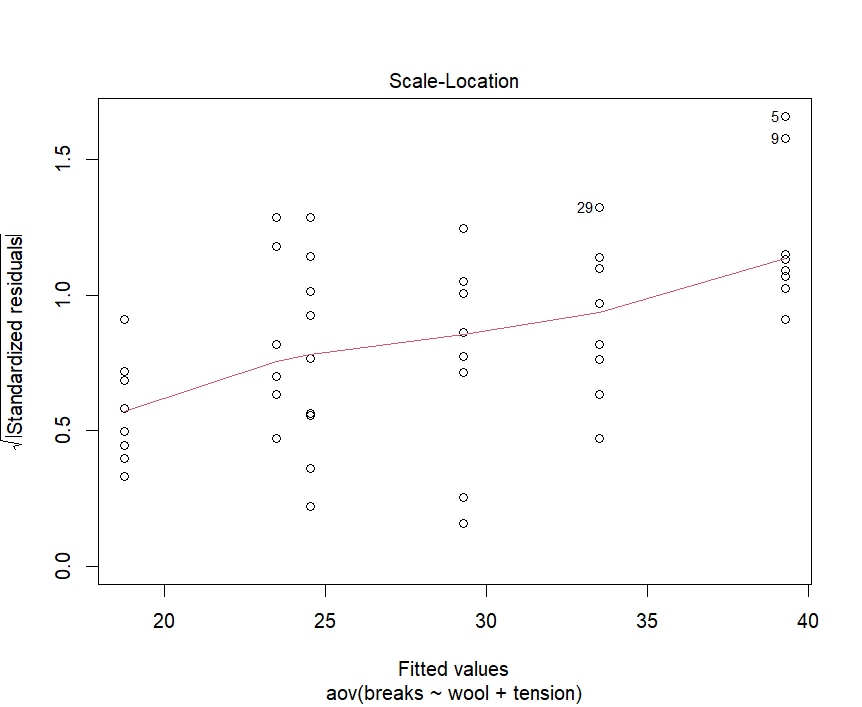


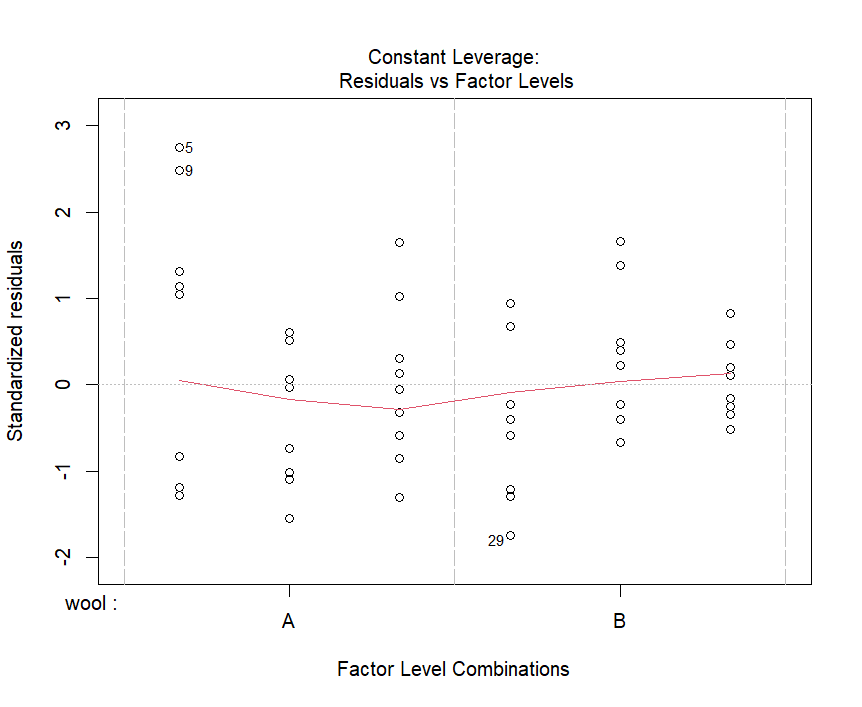


**Plots :**







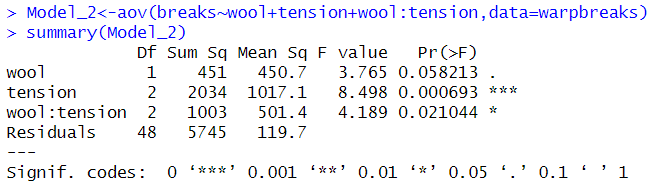


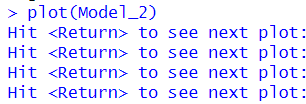
**Model 2 code :**

*> Model\_2<-aov(breaks~wool+tension+wool:tension,data=warpbreaks)*

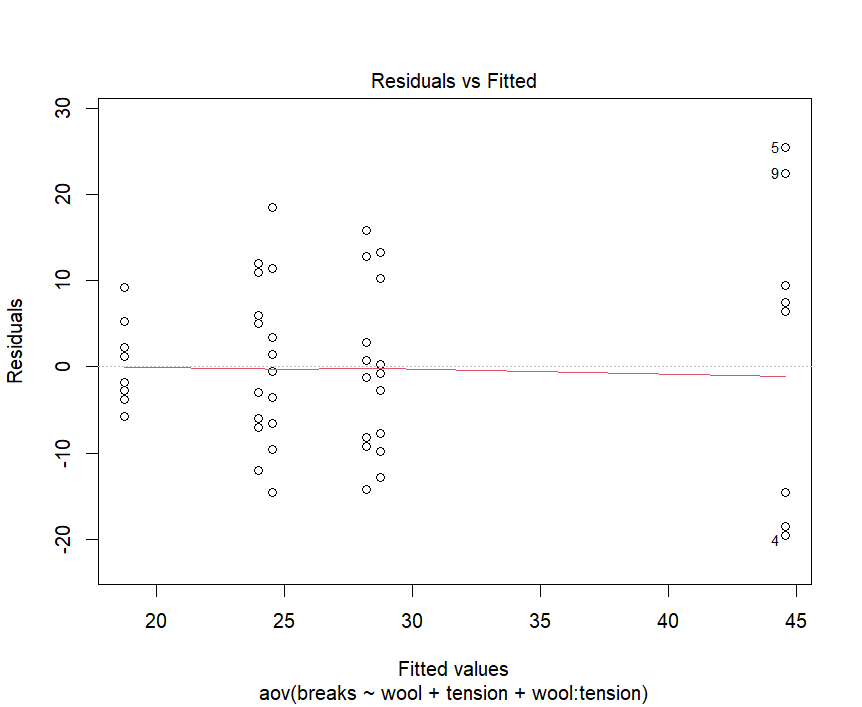
*> summary(Model\_2)*

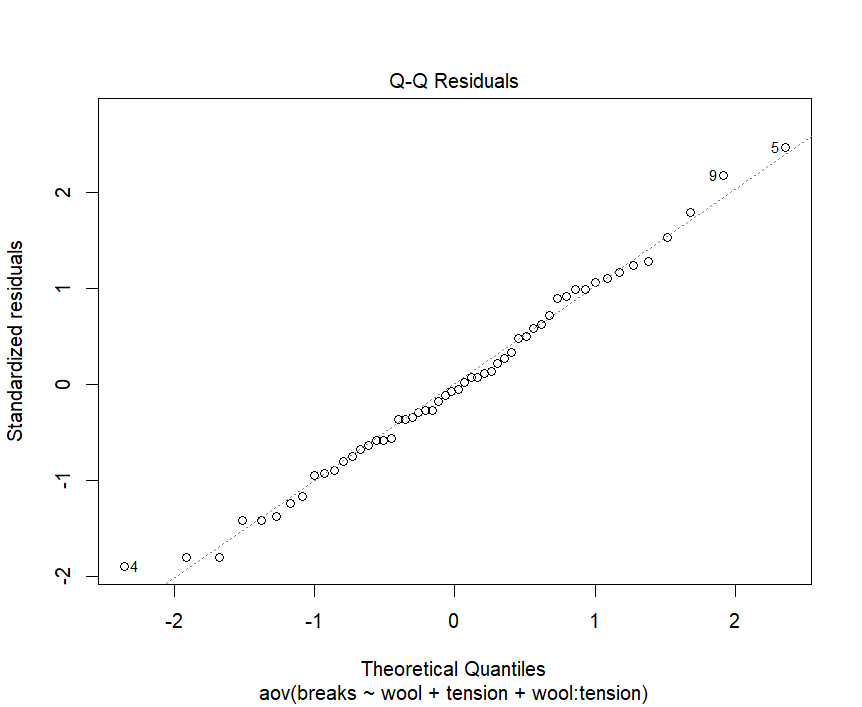
*> plot(Model\_2)*

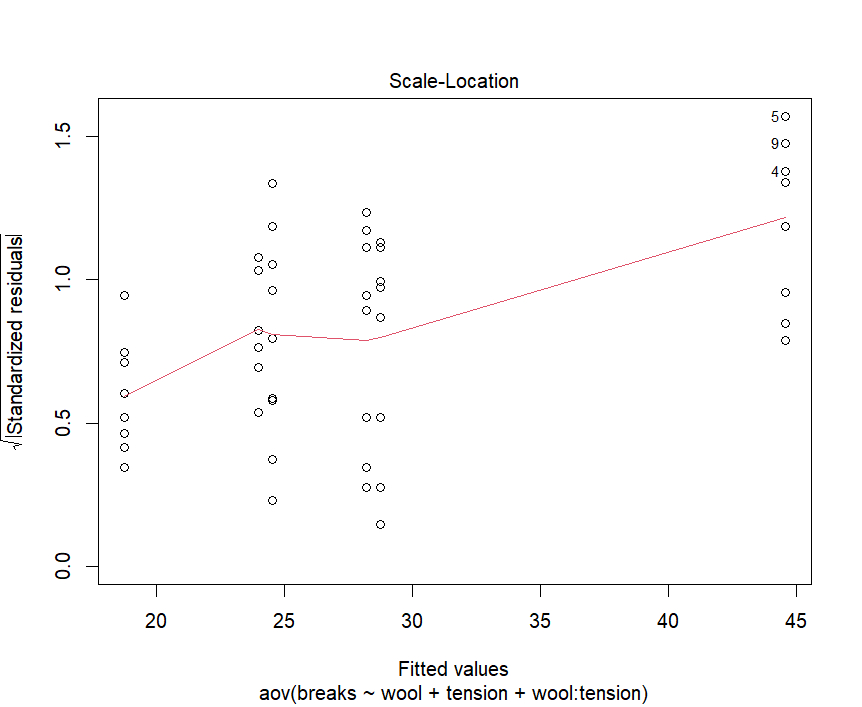
****

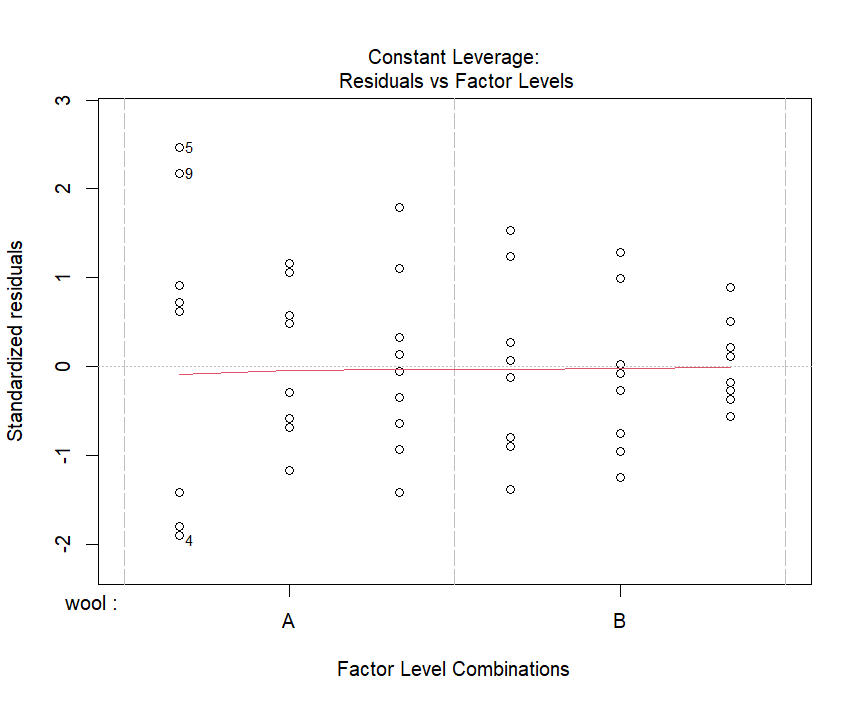
****

**Plots :**

****

****

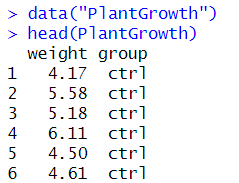
****

****

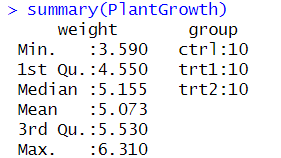
**B. Plant Growth dataset**

*> data("PlantGrowth")*

*> head(PlantGrowth)*

******

*> summary(PlantGrowth)*

******

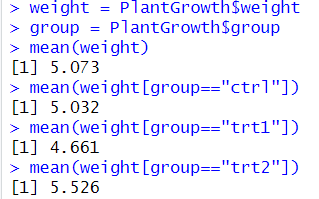
*> levels(PlantGrowth$group)*

******

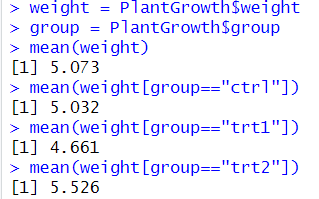
*> weight = PlantGrowth$weight*

*> group = PlantGrowth$group*

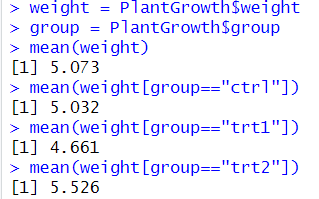
*> mean(weight)*

******

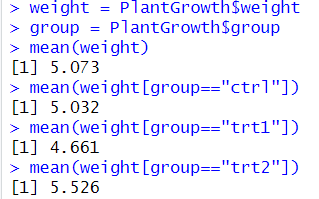
*> mean(weight[group=="ctrl"])*

******

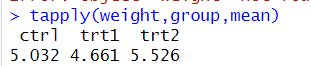
*> mean(weight[group=="trt1"])*

******

*> mean(weight[group=="trt2"])*

******

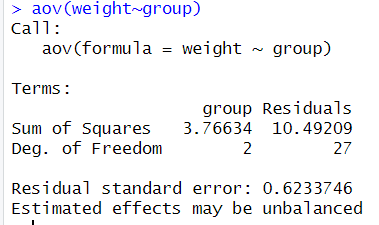
*> tapply(weight,group,mean)*

******

*> tapply(weight,group,length)*

******

*> aov(weight~group)*

******

**Practical 5**

**Simple, Multiple Linear Regression**

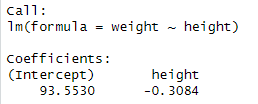
**Code:**

*> height<-c(102,117,105,141,135,115,138,144,137,100,131,119,115,121,113)*

*> weight<-c(61,46,62,54,60,69,51,50,46,64,48,56,64,48,59)*

*> student<-lm(weight~height)*

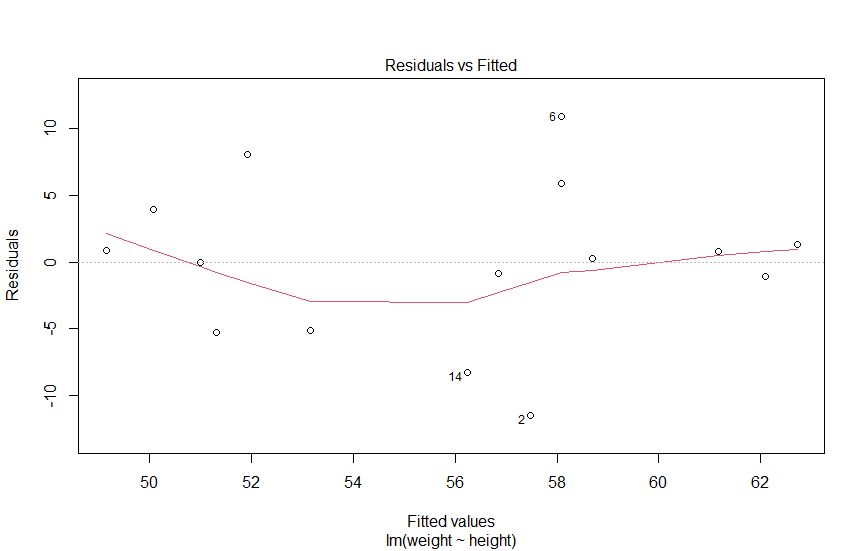
*> student*

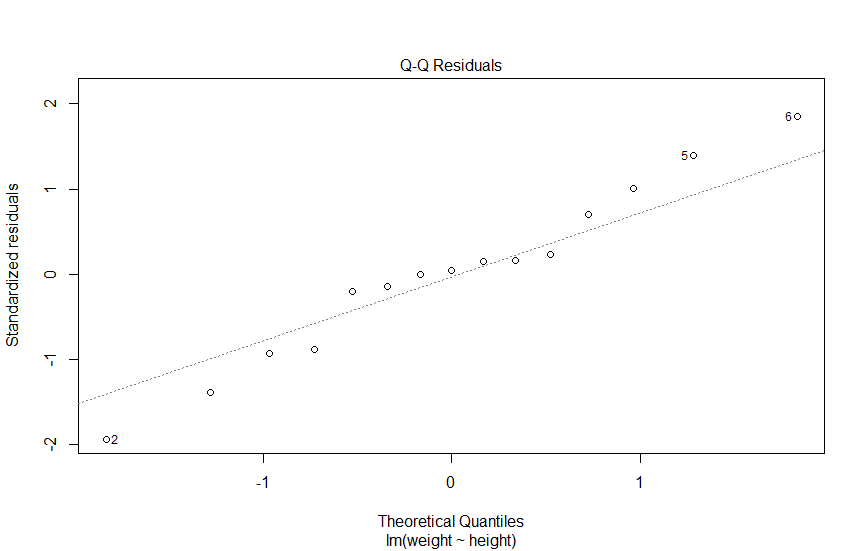
**

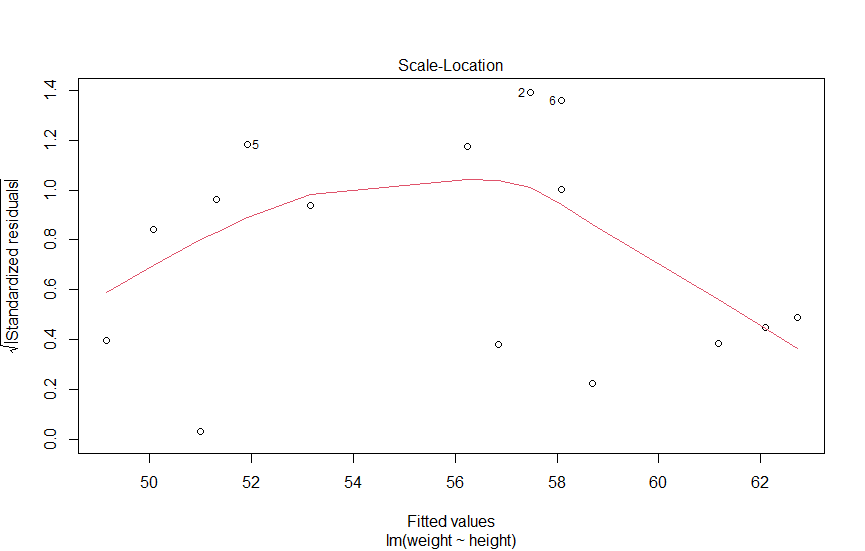
*> predict(student,data.frame(height=199),interval="confidence")*

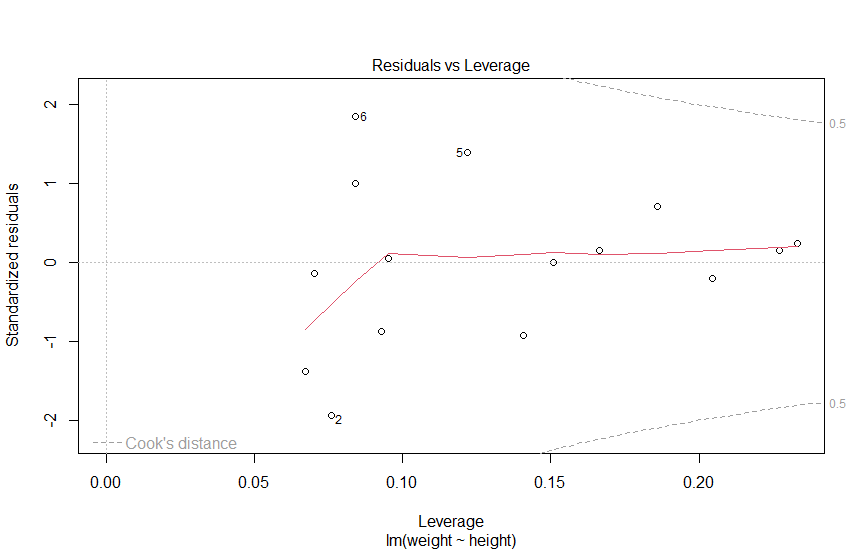


*> plot(student)*





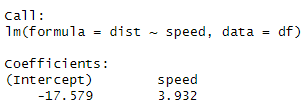




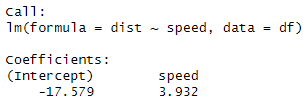
*> df<-datasets::cars*

*> my\_linear\_model<-lm(dist~speed,data=df)*

*> my\_linear\_model*



*> lm(formula=dist~speed,data=df)*



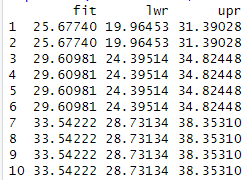
*> variable\_speed<-data.frame(speed=c(11,11,12,12,12,12,13,13,13,13))*

*> linear\_model<-lm(dist~speed,data=df)*

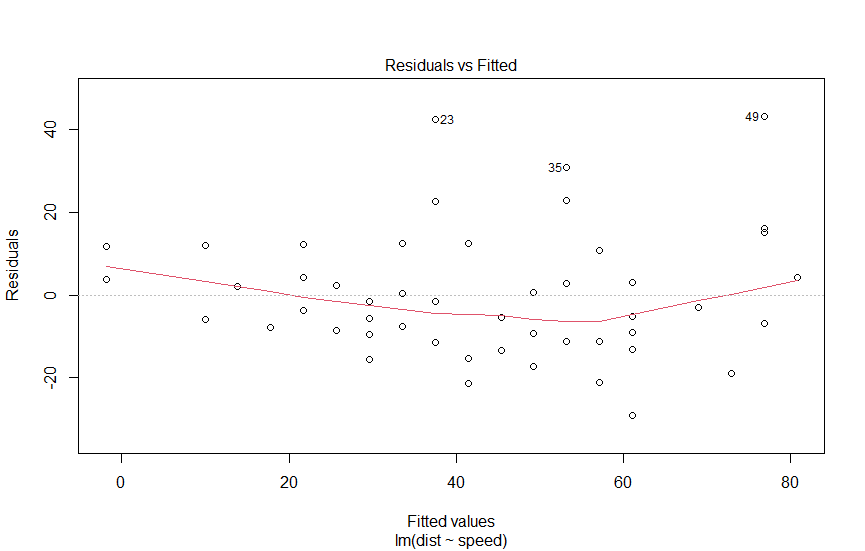
*> predict(linear\_model,newdata = variable\_speed)*

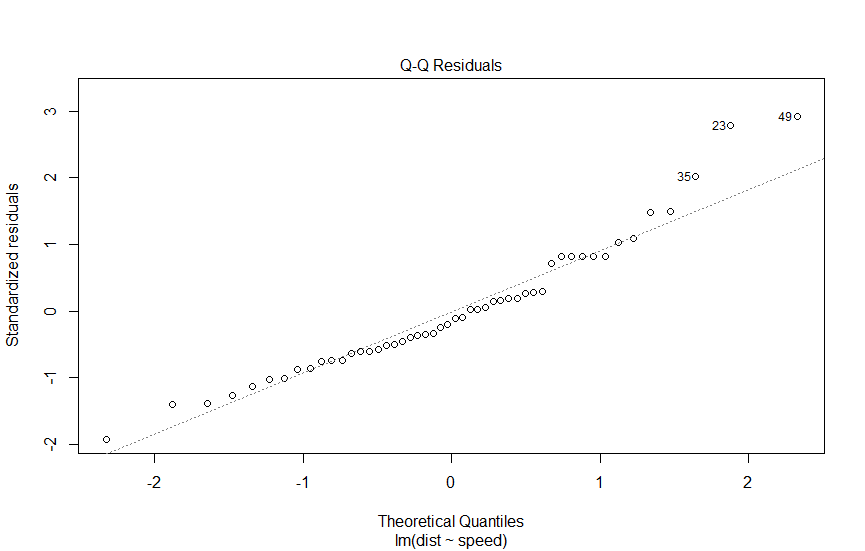


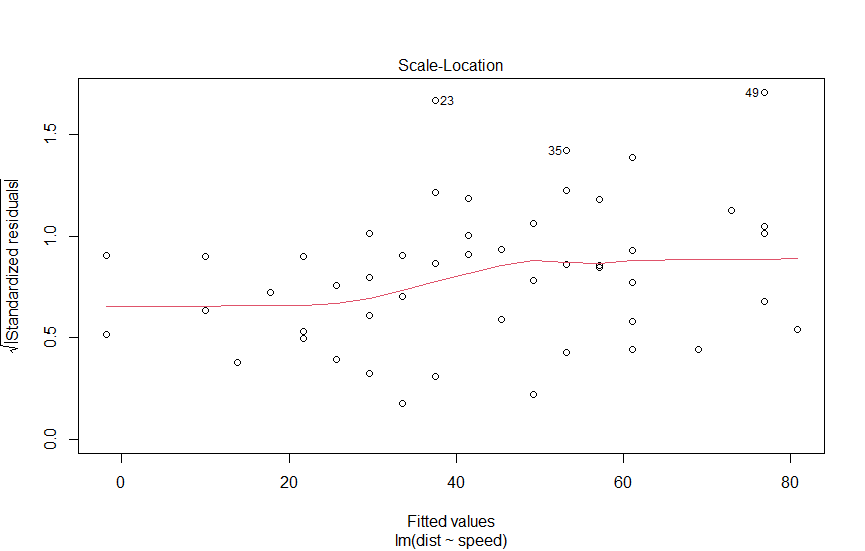
*> predict(linear\_model,newdata = variable\_speed, interval='confidence')*

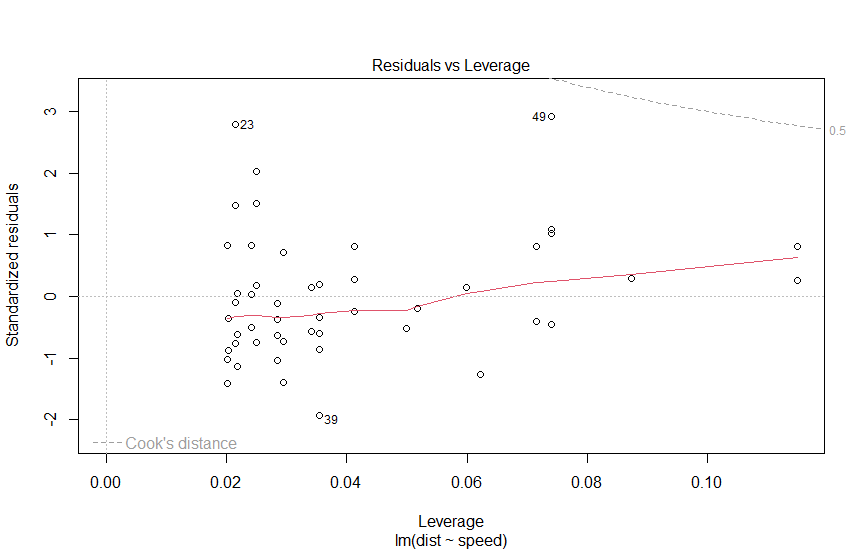


*> plot(linear\_model)*









**Practical 6**

**Hypothesis**

Hypothesis testing is a statistical method used to determine if there is enough evidence in a sample data to draw conclusions about a population.

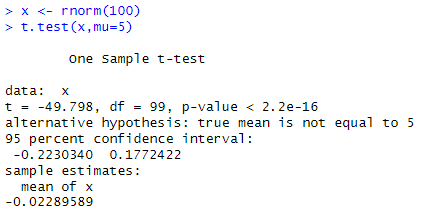
It involves formulating two competing hypotheses, the null hypothesis (H0) and the alternative hypothesis (Ha), and then collecting data to assess the evidence.

**1. One Sample T-test :**

One Sample T-test is used to compare a sample to a known value or a hypothesized value.

*> x <- rnorm(100)*

*> t.test(x, mu=5)*



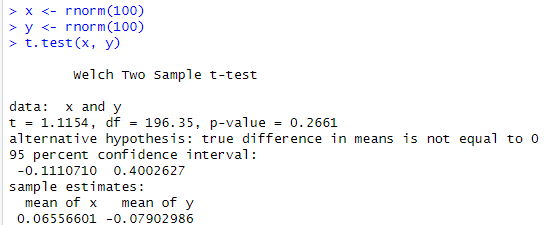
**2. Two Sample T-test**

Two Sample T-test compares two independent samples to assess if there is a significant difference between their means or distributions.

*> x <- rnorm(100)*

*> y <- rnorm(100)*

*> t.test(x, y)*

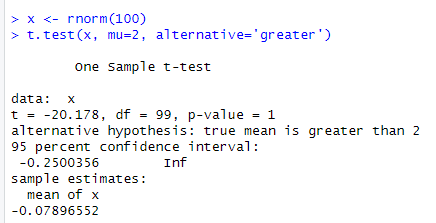
****

**3. Directional Hypothesis Testing**

In the Directional Hypothesis, the null hypothesis is rejected if the test score is too large (for right-tailed and too small for left-tailed). Thus, the rejection region for such a test consists of one part, which is right from the center.

*> x <- rnorm(100)*

*> t.test(x, mu=2, alternative=’greater’)*

****

**4. Directional hypothesis**

*> dataf <- seq(1,20,by=1)*

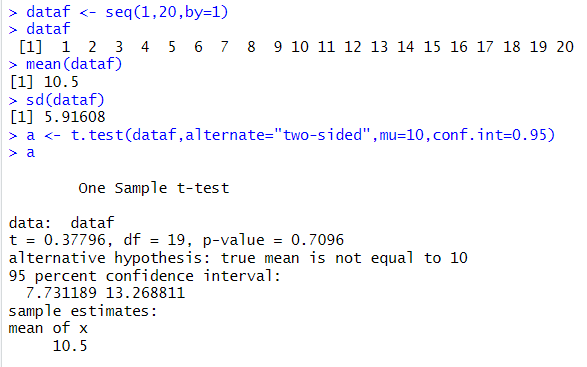
*> dataf*

*> mean(dataf)*

*> sd(dataf)*

*> a <- t.test(dataf,alternate="two-sided",mu=10,conf.int=0.95)*

*> a*

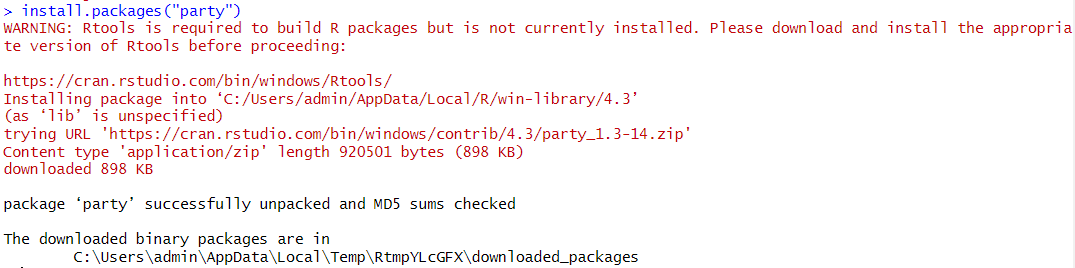
**

**Practical 7**

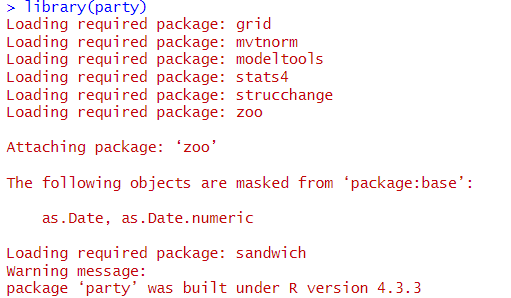
**Decision Tree**

**Code :**

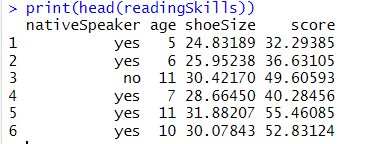
*> install.packages(“party”)*

****

*> library(party)*



*> print(head(readingSkills))*

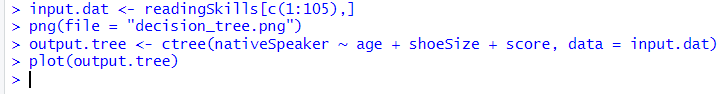


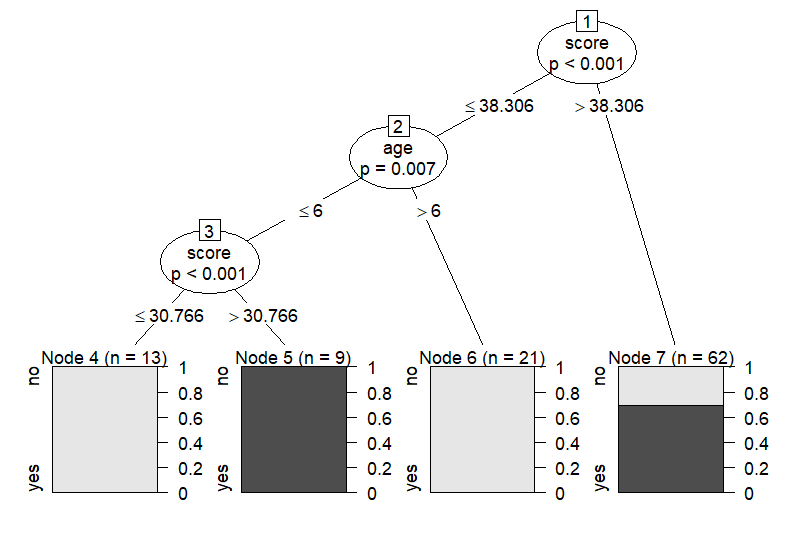
*> input.dat <- readingSkills[c(1:105),]*

*> png(file = "decision\_tree.png")*

*> output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score, data = input.dat)*

*> plot(output.tree)*





**Practical 8**

**Logistic regression**

**Syntax :**

The basic syntax for glm() function in logistic regression is :

glm(formula,data,family)

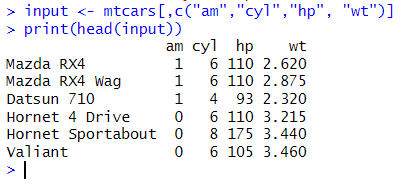
where,

* formula is the symbol presenting the relationship between the variables.
* data is the data set giving the values of these variables.
* Family is an R object to specify the details of the model. It’s value is binomial for logistic regression

**Code :**

*> input <- mtcars[,c("am","cyl","hp", "wt")]*

*>print(head(input))*

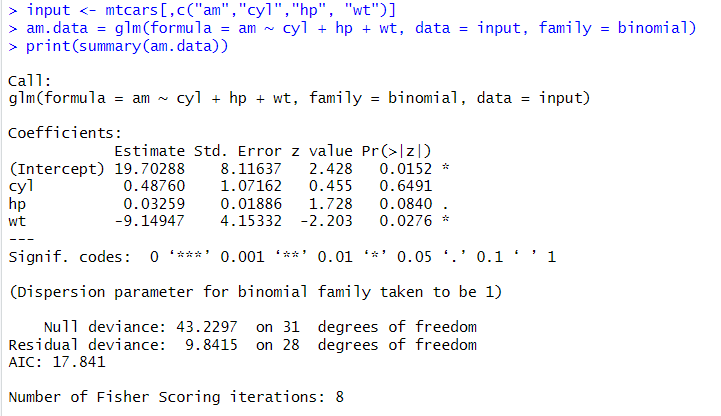


Creating the regression model :

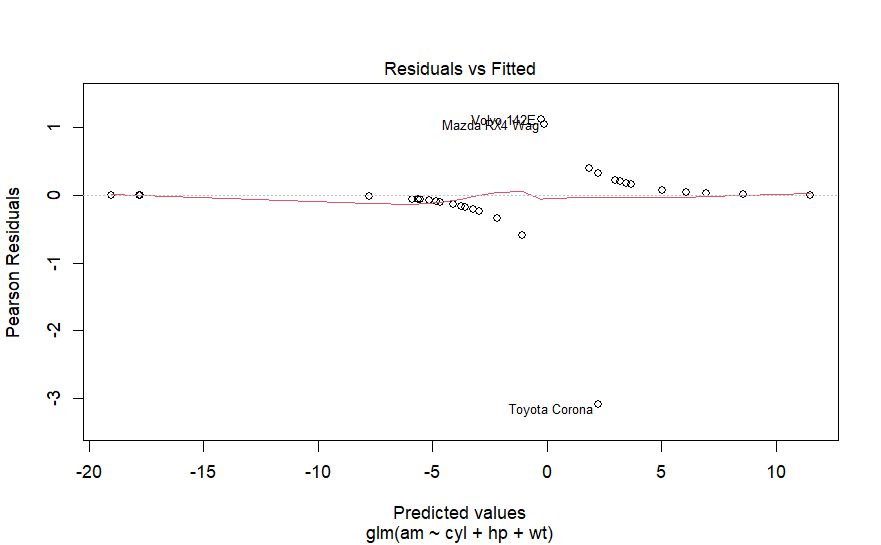
*> input <- mtcars[,c("am","cyl","hp", "wt")]*

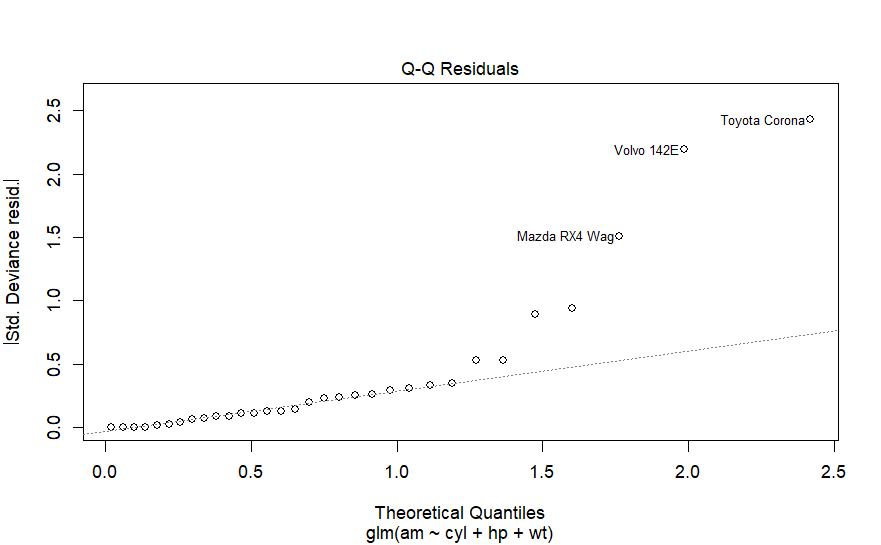
*> am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)*

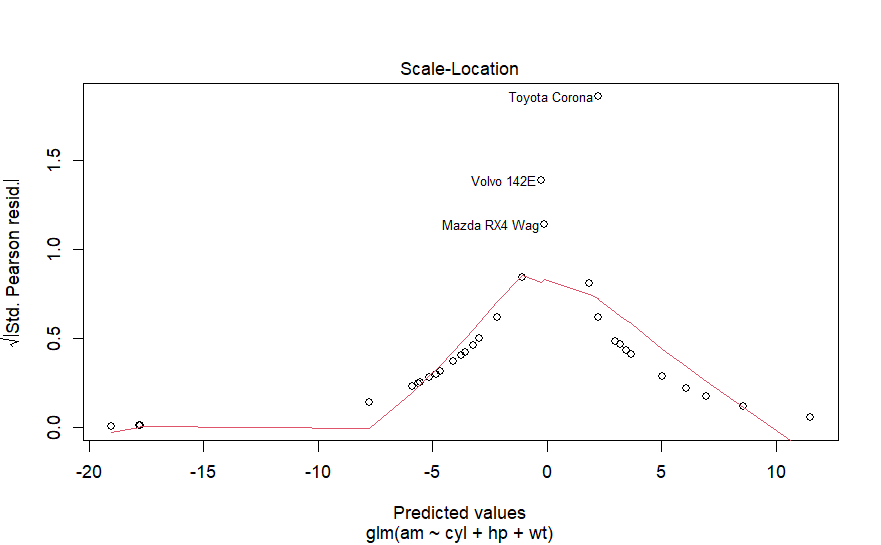
*> print(summary(am.data))*

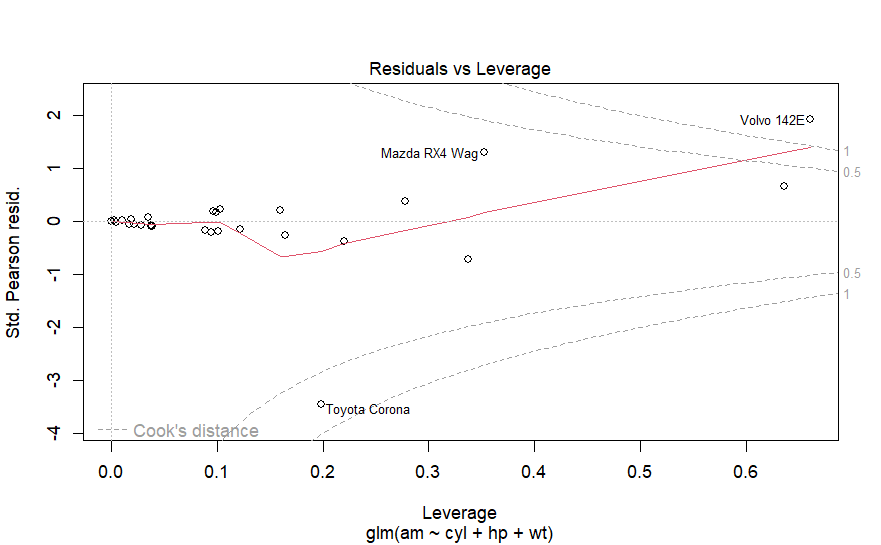


*> plot(am.data)*









**Practical 9**

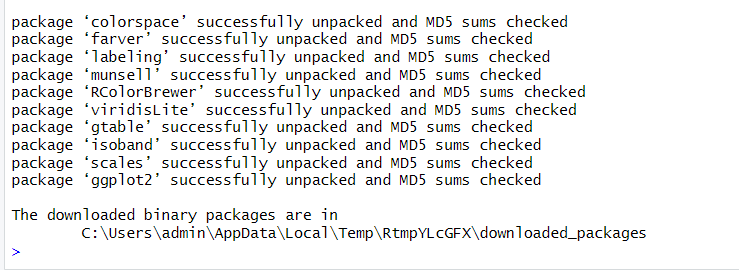
**K-means Clustering**

K-means clustering is used with unlabeled data, but in this case, we have a labeled dataset so we have to use the iris data without the Species column. In this way, the algorithm will cluster the data and we will be able to compare the predicted results with the original results, getting the accuracy of the model.

**Code :**

*> install.packages(“ggplot2”)*

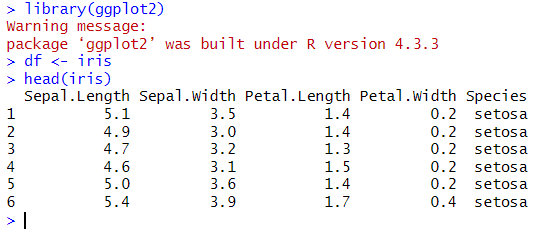




*> library(ggplot2)*

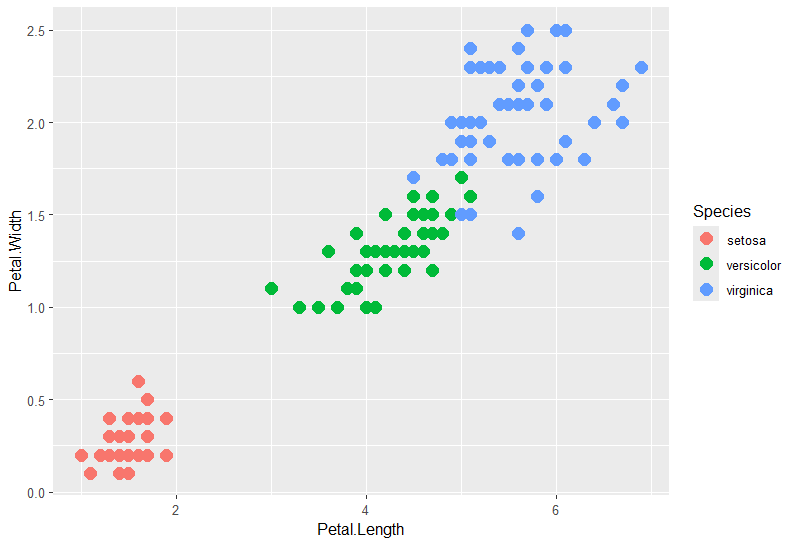
*> df <- iris*

*> head(iris)*



*> ggplot(df,aes(Petal.Length,Petal.Width))+geom\_point(aes(col=Species),size=4)*

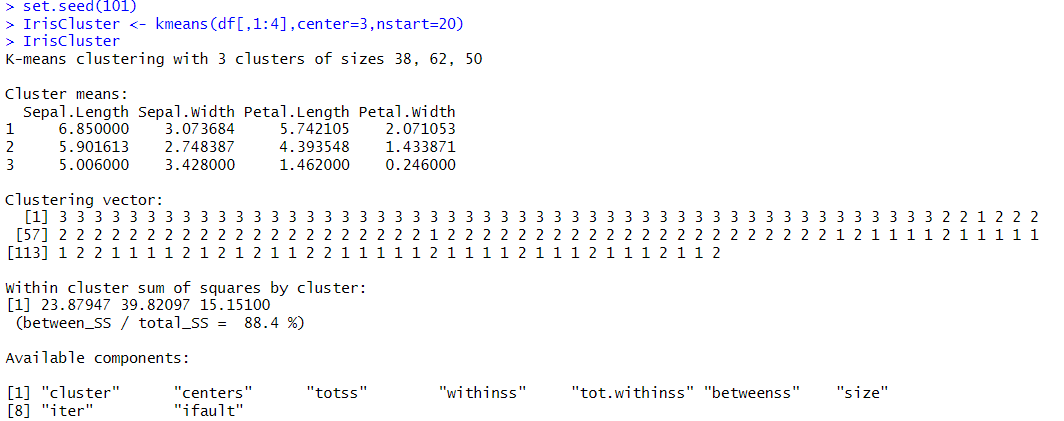




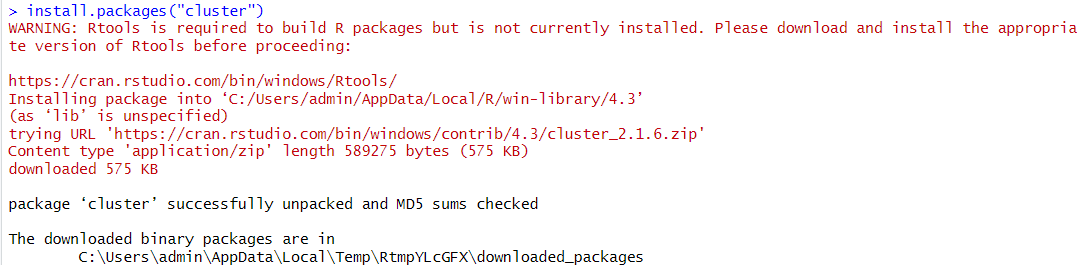
*> set.seed(101)*

*> IrisCluster <- kmeans(df[,1:4],center=3,nstart=20)*

*> IrisCluster*



*> install.packages("cluster")*



*> library(cluster)*

*> clusplot(iris,IrisCluster$cluster, color = T, shade = T, labels = 0, lines = 0)*

