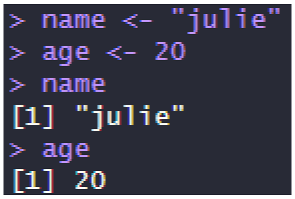
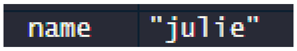
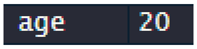
Simple Variable Declaration



```
> name <- "julie"
> age <- 20
> name
[1] "julie"
> age
[1] 20
```

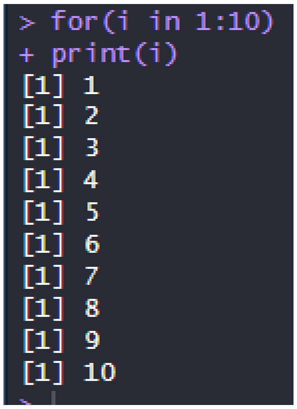


name	"julie"



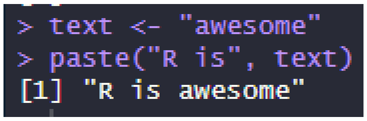
age	20

For Loop



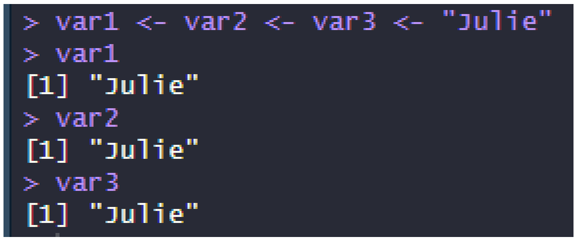
```
> for(i in 1:10)
+ print(i)
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

Paste

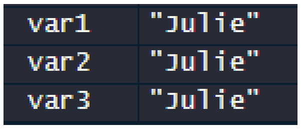


```
> text <- "awesome"
> paste("R is", text)
[1] "R is awesome"
```

Declaration of multiple variables

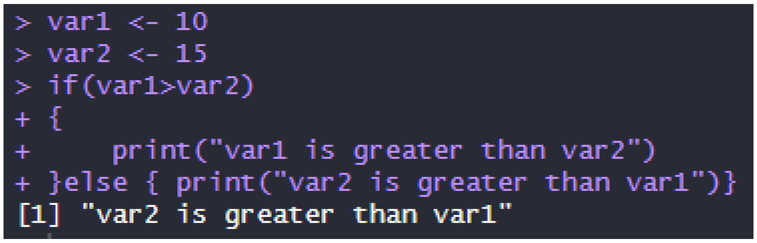


```
> var1 <- var2 <- var3 <- "Julie"
> var1
[1] "Julie"
> var2
[1] "Julie"
> var3
[1] "Julie"
```



var1	"Julie"
var2	"Julie"
var3	"Julie"

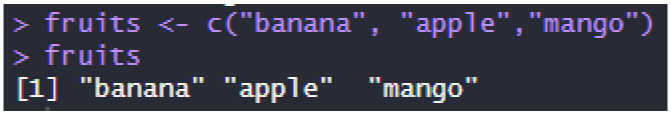
Take 2 variables & compare them using if-else statement



```
> var1 <- 10
> var2 <- 15
> if(var1>var2)
+ {
+     print("var1 is greater than var2")
+ }else { print("var2 is greater than var1")}
[1] "var2 is greater than var1"
```

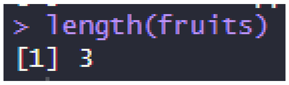
VECTOR Function

Vector of strings



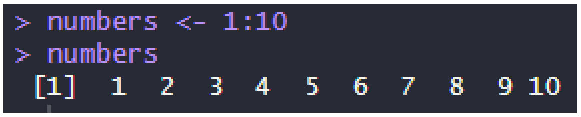
```
> fruits <- c("banana", "apple","mango")
> fruits
[1] "banana" "apple"  "mango"
```

Length of string



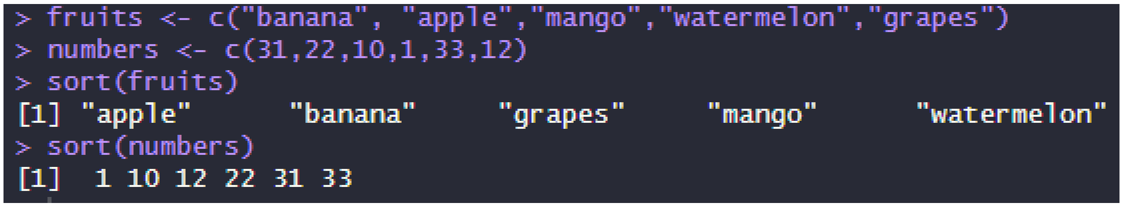
```
> length(fruits)
[1] 3
```

Vector with numeral values in a sequence



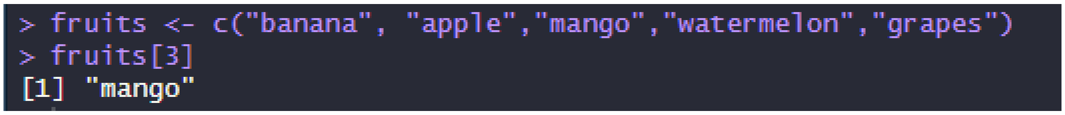
```
> numbers <- 1:10
> numbers
 [1]  1  2  3  4  5  6  7  8  9 10
```

Sorting the strings



```
> fruits <- c("banana", "apple","mango","watermelon","grapes")
> numbers <- c(31,22,10,1,33,12)
> sort(fruits)
[1] "apple"      "banana"     "grapes"     "mango"      "watermelon"
> sort(numbers)
[1]  1 10 12 22 31 33
```

Accessing elements using indexing

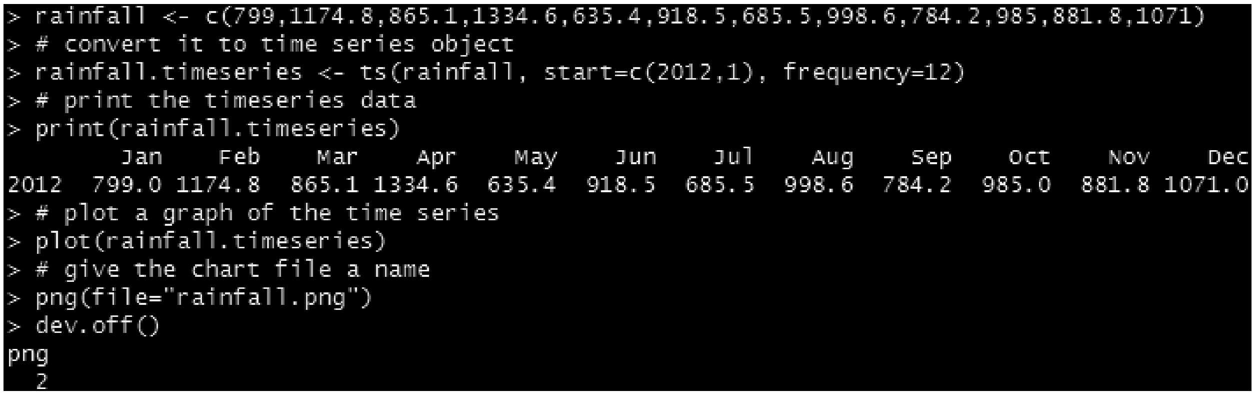


```
> fruits <- c("banana", "apple","mango","watermelon","grapes")
> fruits[3]
[1] "mango"
```

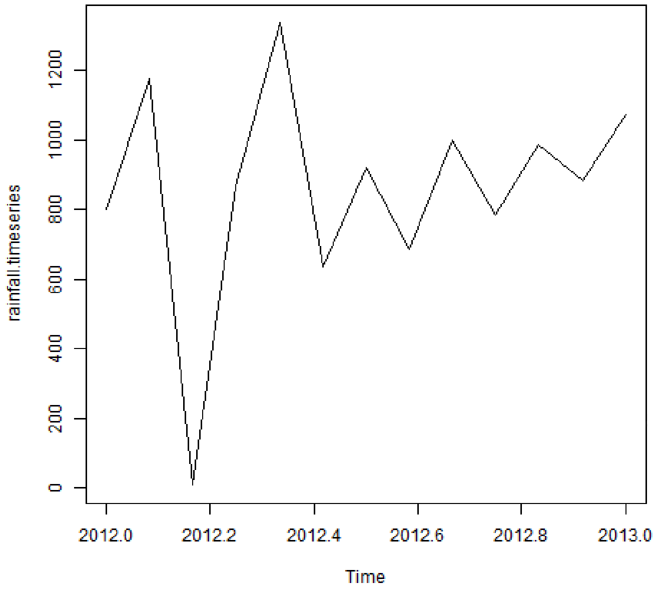
Practical 2

Time Series Analysis

```
rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,881.8,1071)
# convert it to time series object
rainfall.timeseries <- ts(rainfall, start=c(2012,1), frequency=12)
# print the timeseries data
print(rainfall.timeseries)
# plot a graph of the time series
plot(rainfall.timeseries)
# give the chart file a name
png(file="rainfall.png")
dev.off()
```



```
> rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,881.8,1071)
> # convert it to time series object
> rainfall.timeseries <- ts(rainfall, start=c(2012,1), frequency=12)
> # print the timeseries data
> print(rainfall.timeseries)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
2012  799.0 1174.8  865.1 1334.6  635.4  918.5  685.5  998.6  784.2  985.0  881.8 1071.0
> # plot a graph of the time series
> plot(rainfall.timeseries)
> # give the chart file a name
> png(file="rainfall.png")
> dev.off()
png 
  2 
```


rainfall.timeseries
0
200
400
600
800
1000
1200
2012.0
2012.2
2012.4
2012.6
2012.8
2013.0
Time

Practical 3

Time Series Frequency Analysis

data() - Loads specified data sets, or list the available data sets.

```
data(AirPassengers)
```

class() - returns the values of the class attribute of an R object.

```
class(AirPassengers)
```

start() and end() - Extract and encode the times the first and last observations were taken.

```
start(AirPassengers)
end(AirPassengers)
```

frequency() - returns the number of samples per unit time.

```
frequency(AirPassengers)
```

summary() - generic function used to produce result summaries of the results of various model fitting functions.

```
summary(AirPassengers)
```

plot() - Generic function for plotting of R objects.

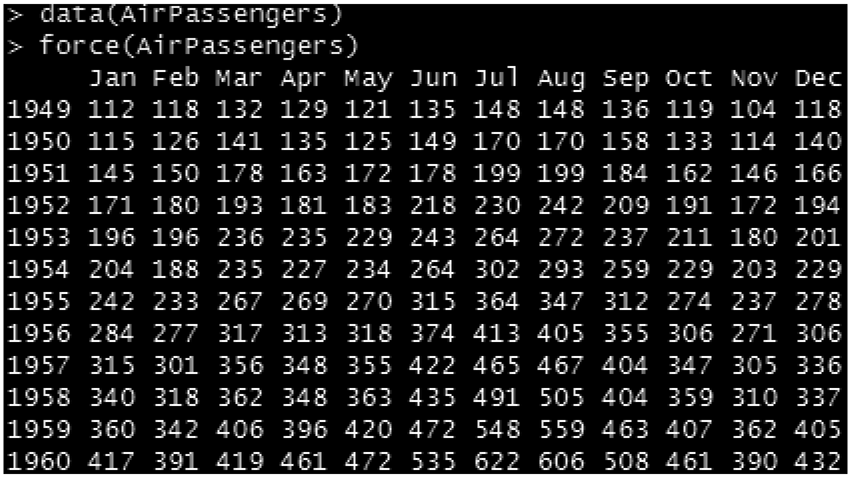
```
plot(AirPassengers)
```

abline() - adds one or more straight lines through the current plot.
lm() - used to fit linear models.

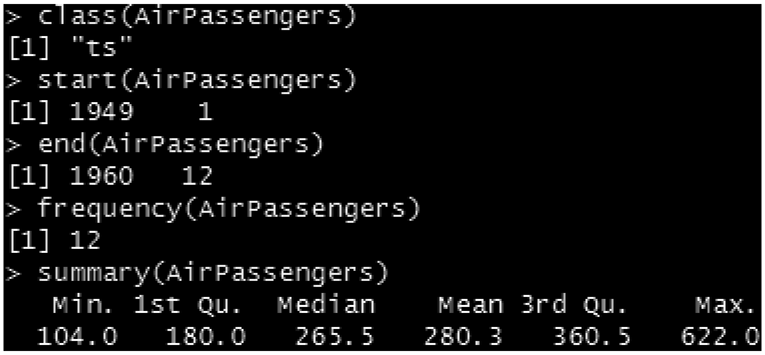
```
abline(reg=lm(AirPassengers~time(AirPassengers)))
```

boxplot() - produce box-and-whisker plot of the given grouped values.

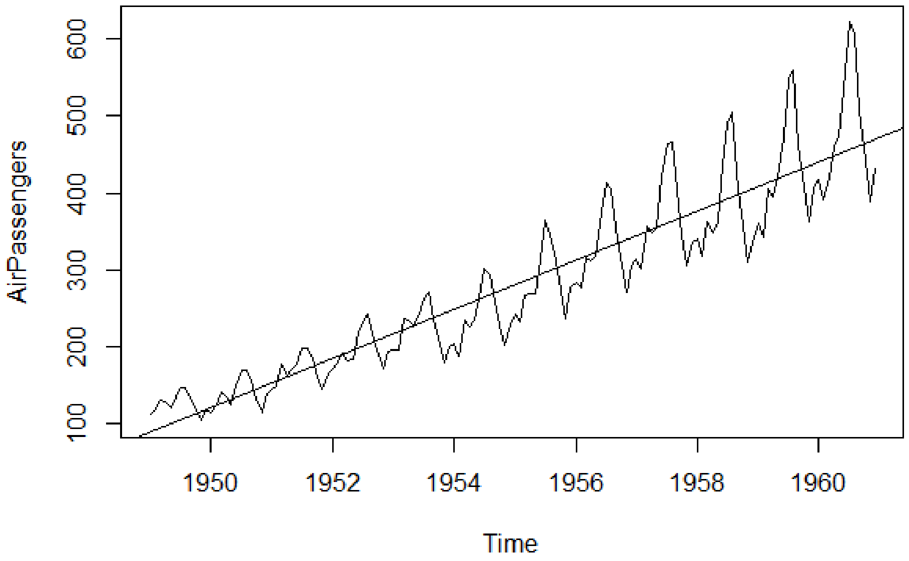
```
boxplot(AirPassengers~cycle(AirPassengers))
```

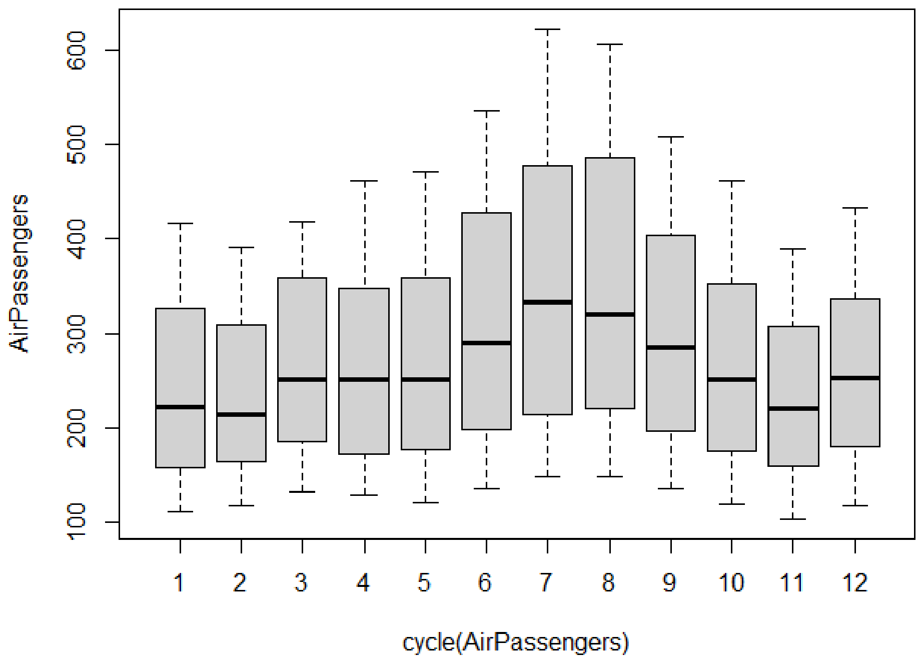


```
> data(AirPassengers)
> force(AirPassengers)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

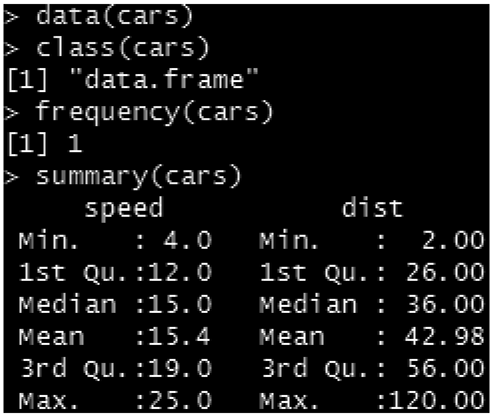


```
> class(AirPassengers)
[1] "ts"
> start(AirPassengers)
[1] 1949    1
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
```

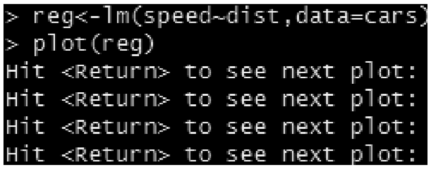




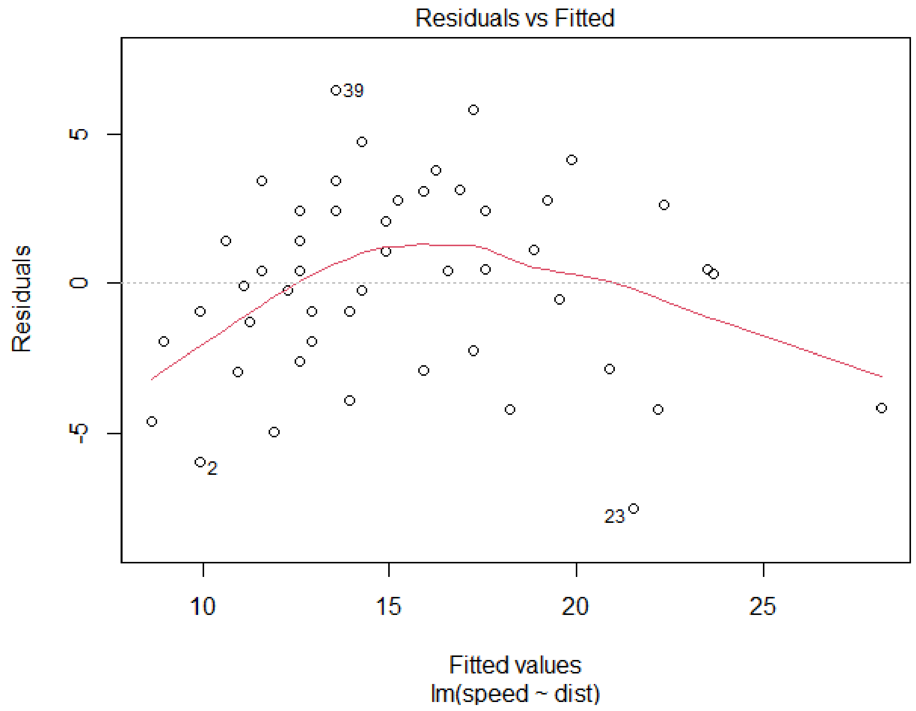
```
data(cars)
class(cars)
frequency(cars)
summary(cars)
reg<-lm(speed~dist,data=cars)
plot(reg)
```

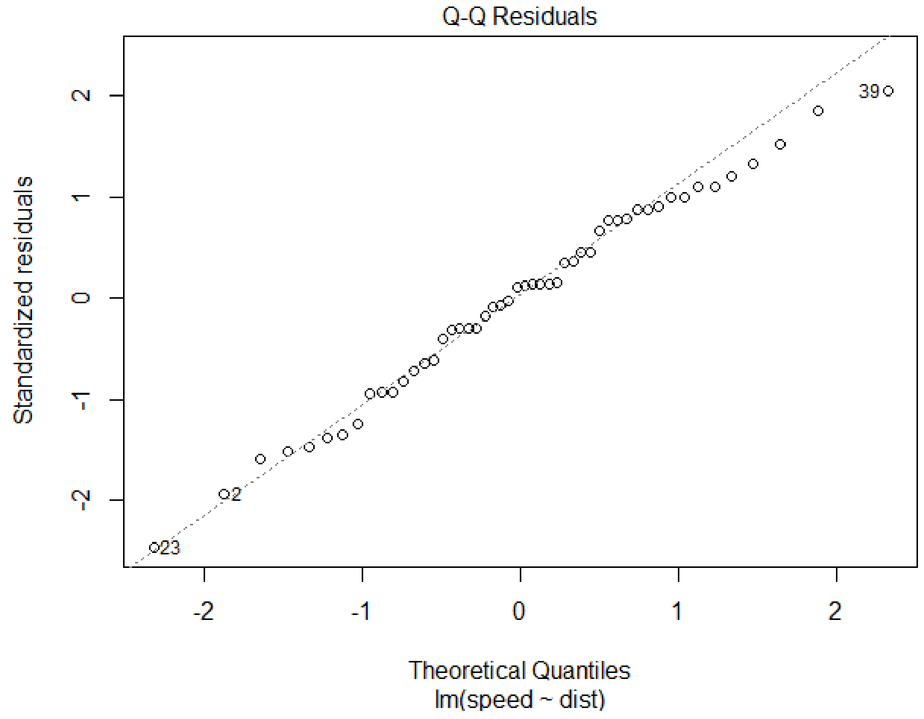


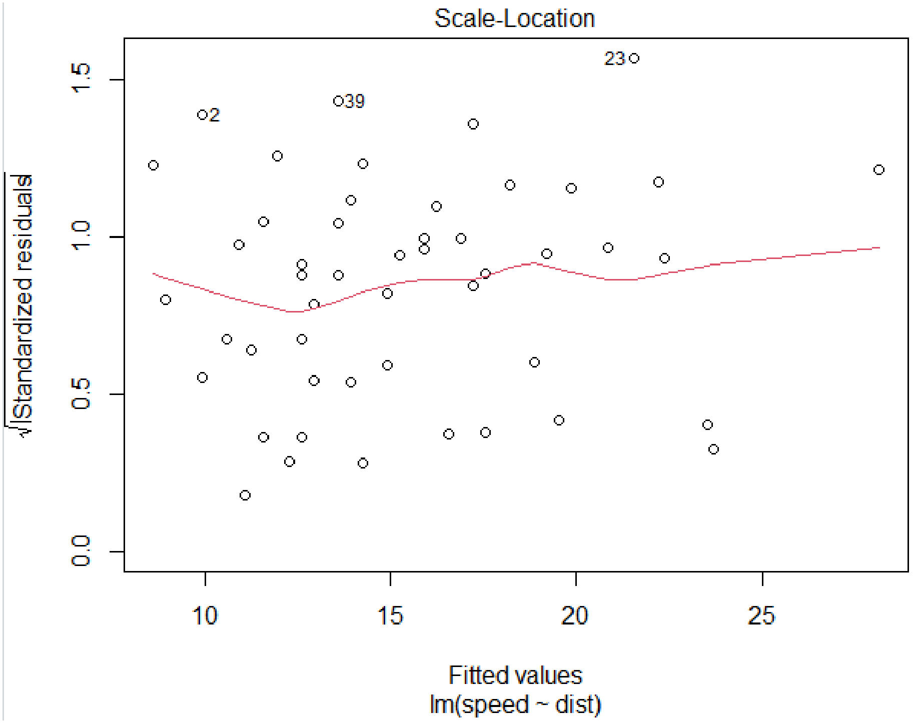
```
> data(cars)
> class(cars)
[1] "data.frame"
> frequency(cars)
[1] 1
> summary(cars)
     speed           dist       
 Min.   : 4.0   Min.   :  2.00  
 1st Qu.:12.0   1st Qu.: 26.00  
 Median :15.0   Median : 36.00  
 Mean   :15.4   Mean   : 42.98  
 3rd Qu.:19.0   3rd Qu.: 56.00  
 Max.   :25.0   Max.   :120.00  
```

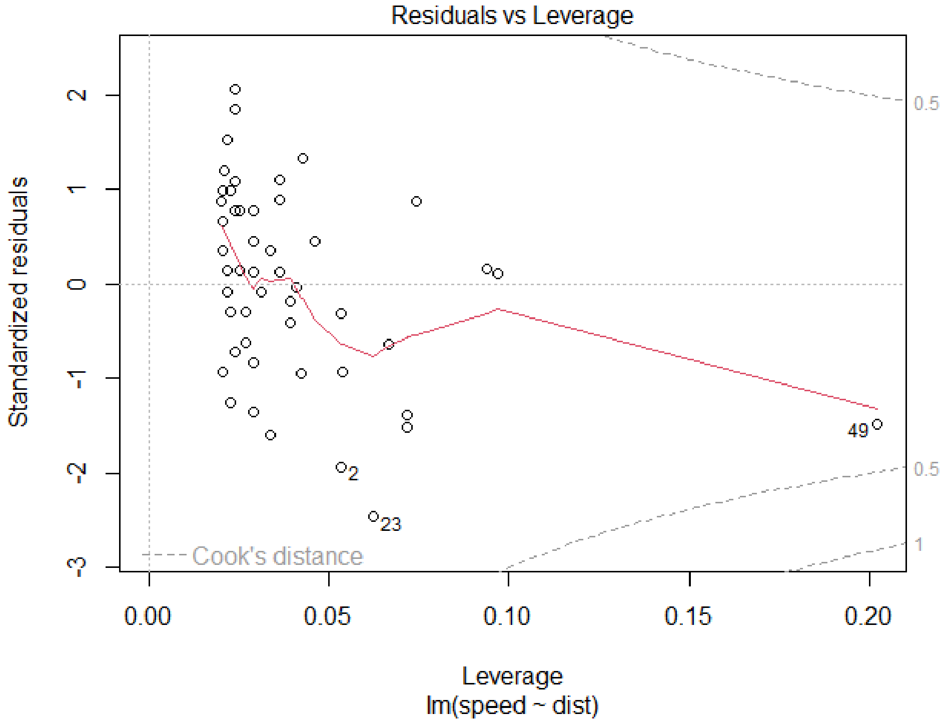


```
> reg<-lm(speed~dist,data=cars)
> plot(reg)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```


Residuals vs Fitted
Residuals
5
0
-5
39
2
23
10
15
20
25
Fitted values
lm(speed ~ dist)

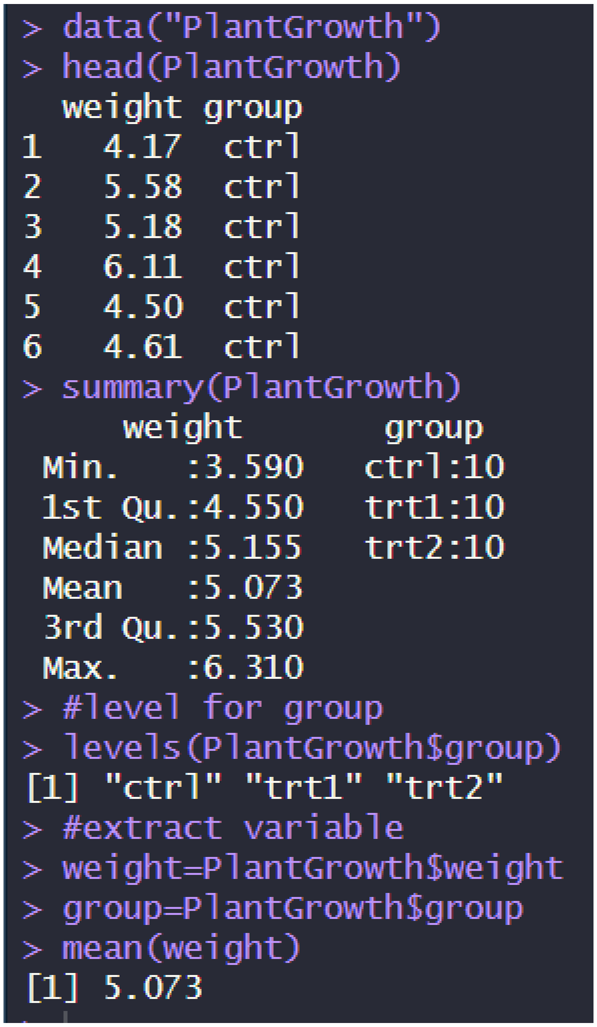

Q-Q Residuals
Standardized residuals
2
1
0
-1
-2
39
2
23
-2
-1
0
1
2
Theoretical Quantiles
lm(speed ~ dist)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
15
20
25
Fitted values
lm(speed ~ dist)
23
39
2

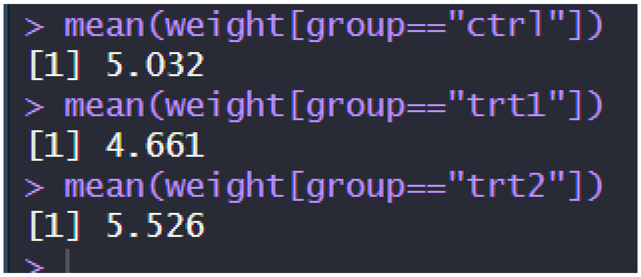

Residuals vs Leverage
Standardized residuals
-3
-2
-1
0
1
2
0.00
0.05
0.10
0.15
0.20
Leverage
lm(speed ~ dist)
Cook's distance
0.5
0.5
1
49
2
23

Practical 4

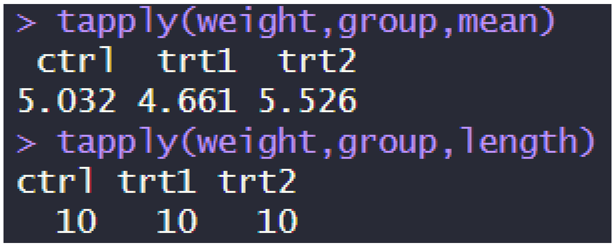
Analysis of Variates



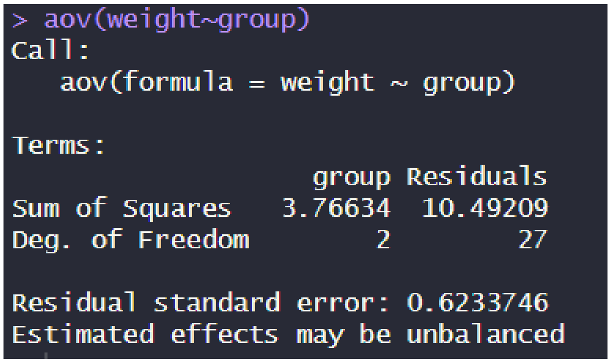
```
> data("PlantGrowth")
> head(PlantGrowth)
  weight group
1   4.17  ctrl
2   5.58  ctrl
3   5.18  ctrl
4   6.11  ctrl
5   4.50  ctrl
6   4.61  ctrl
> summary(PlantGrowth)
     weight       group   
 Min.   :3.590   ctrl:10  
 1st Qu.:4.550   trt1:10  
 Median :5.155   trt2:10  
 Mean   :5.073            
 3rd Qu.:5.530            
 Max.   :6.310            
> #level for group
> levels(PlantGrowth$group)
[1] "ctrl" "trt1" "trt2"
> #extract variable
> weight=PlantGrowth$weight
> group=PlantGrowth$group
> mean(weight)
[1] 5.073
```



```
> mean(weight[group=="ctrl"])
[1] 5.032
> mean(weight[group=="trt1"])
[1] 4.661
> mean(weight[group=="trt2"])
[1] 5.526
>
```



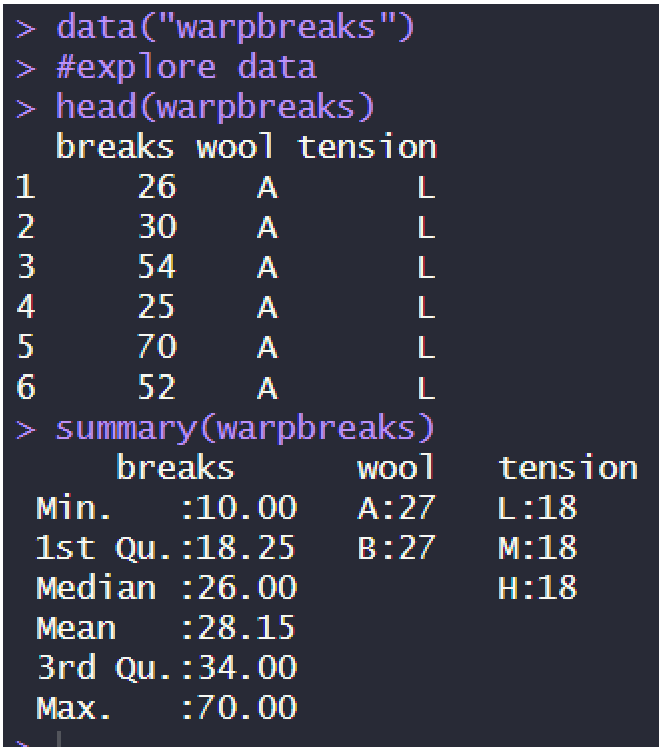
```
> tapply(weight,group,mean)
 ctrl  trt1  trt2 
5.032 4.661 5.526 
> tapply(weight,group,length)
ctrl trt1 trt2 
  10   10   10 
```



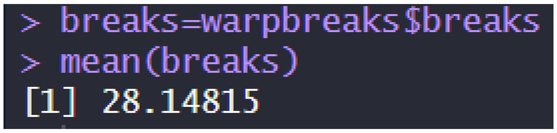
```
> aov(weight~group)
Call:
   aov(formula = weight ~ group)

Terms:
                   group Residuals
Sum of Squares   3.76634  10.49209
Deg. of Freedom        2        27

Residual standard error: 0.6233746
Estimated effects may be unbalanced
```

```
data("PlantGrowth")
head(PlantGrowth)
```

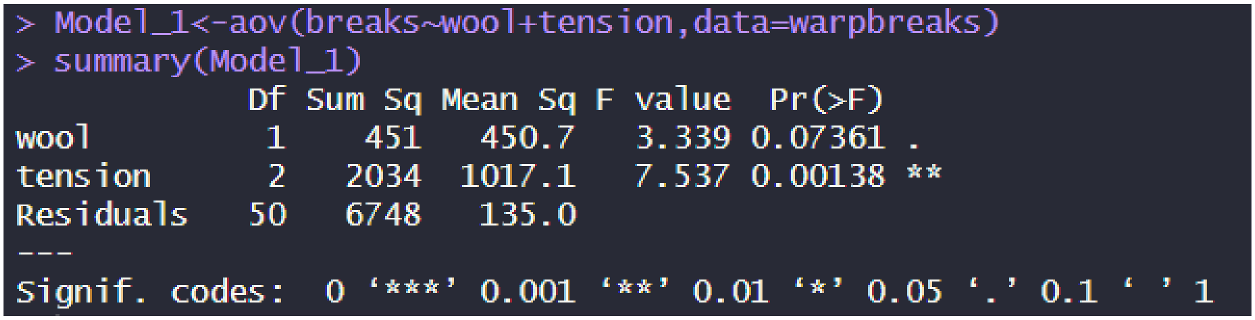
```
summary(PlantGrowth)
#level for group
levels(PlantGrowth$group)
#extract variable
weight=PlantGrowth$weight
group=PlantGrowth$group
mean(weight)

mean(weight[group=="ctrl"])
mean(weight[group=="trt1"])
mean(weight[group=="trt2"])

tapply(weight,group,mean)
tapply(weight,group,length)
aov(weight~group)
```



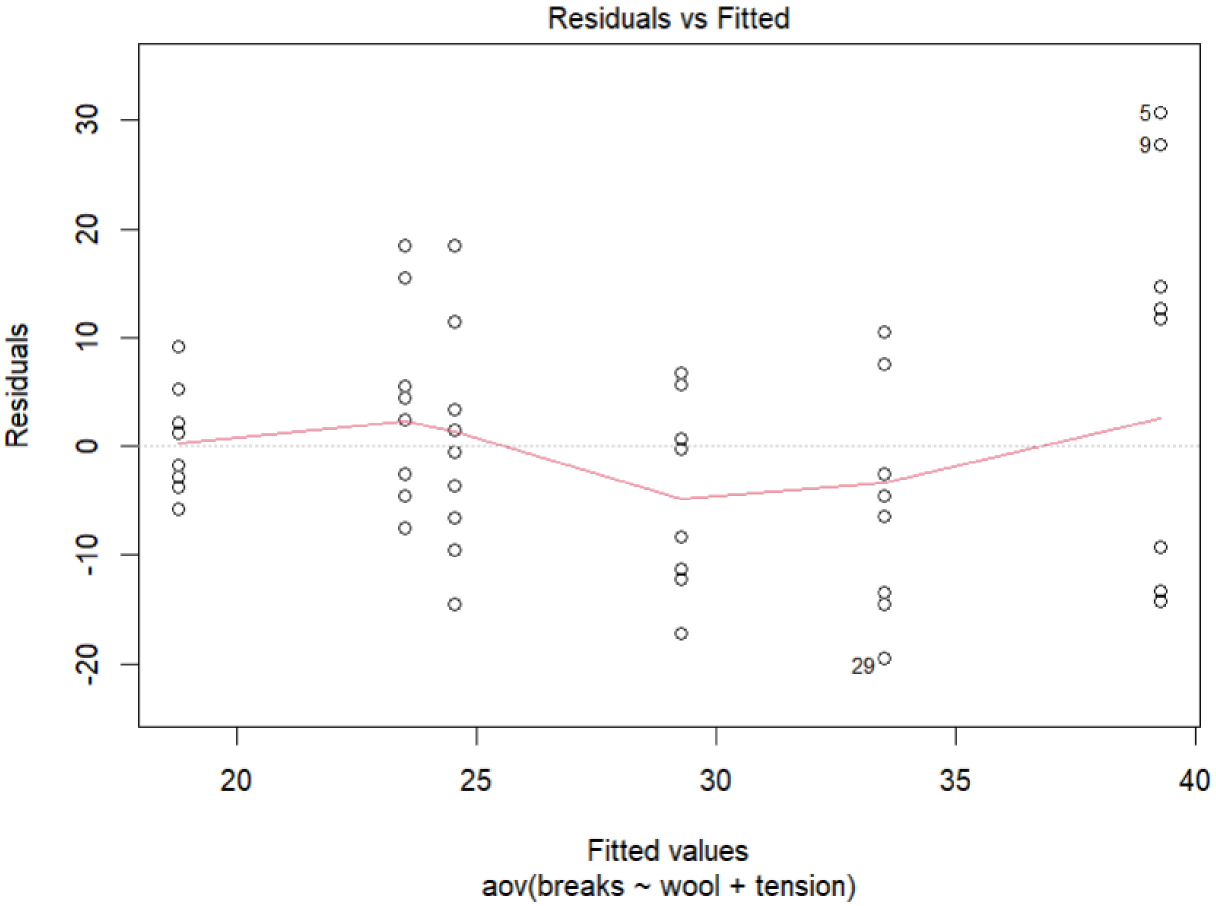
```
> data("warpbreaks")
> #explore data
> head(warpbreaks)
  breaks wool tension
1     26    A       L
2     30    A       L
3     54    A       L
4     25    A       L
5     70    A       L
6     52    A       L
> summary(warpbreaks)
     breaks      wool   tension
 Min.   :10.00   A:27   L:18
 1st Qu.:18.25   B:27   M:18
 Median :26.00          H:18
 Mean   :28.15
 3rd Qu.:34.00
 Max.   :70.00
```

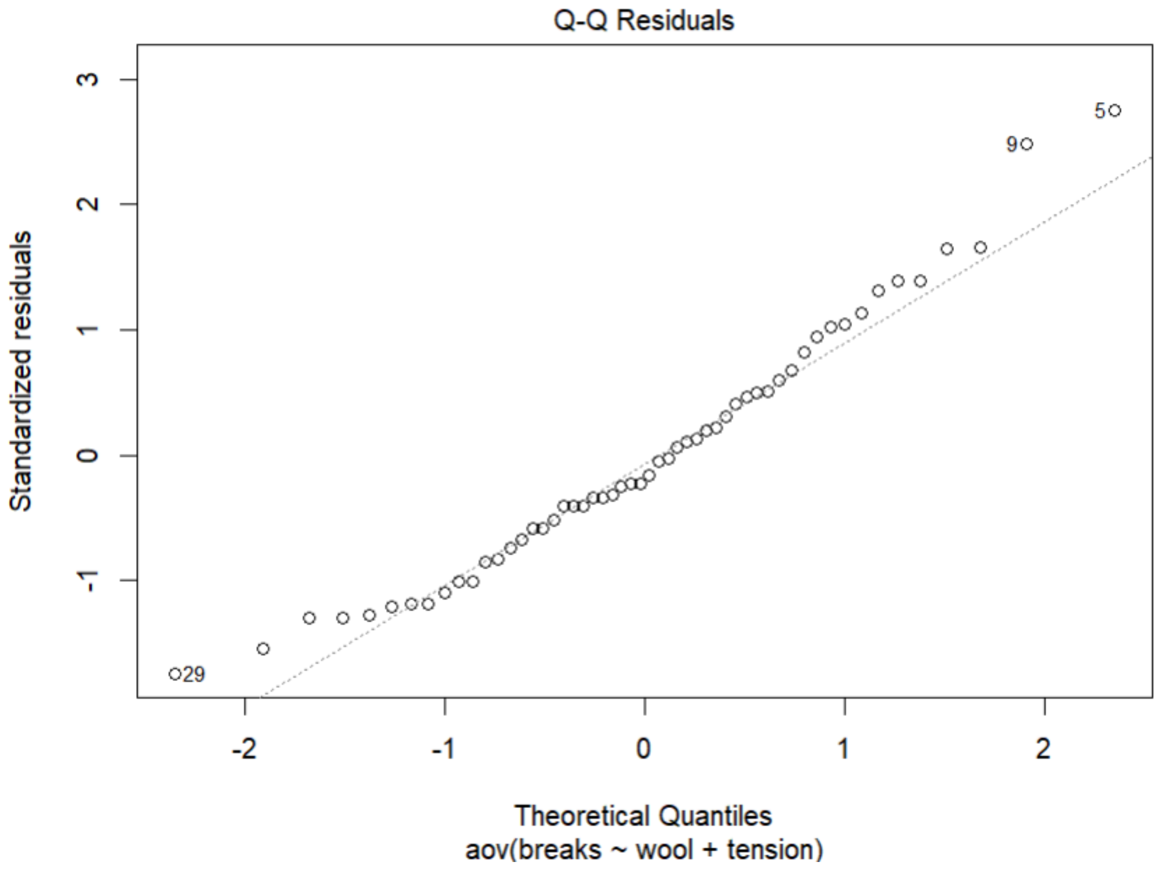


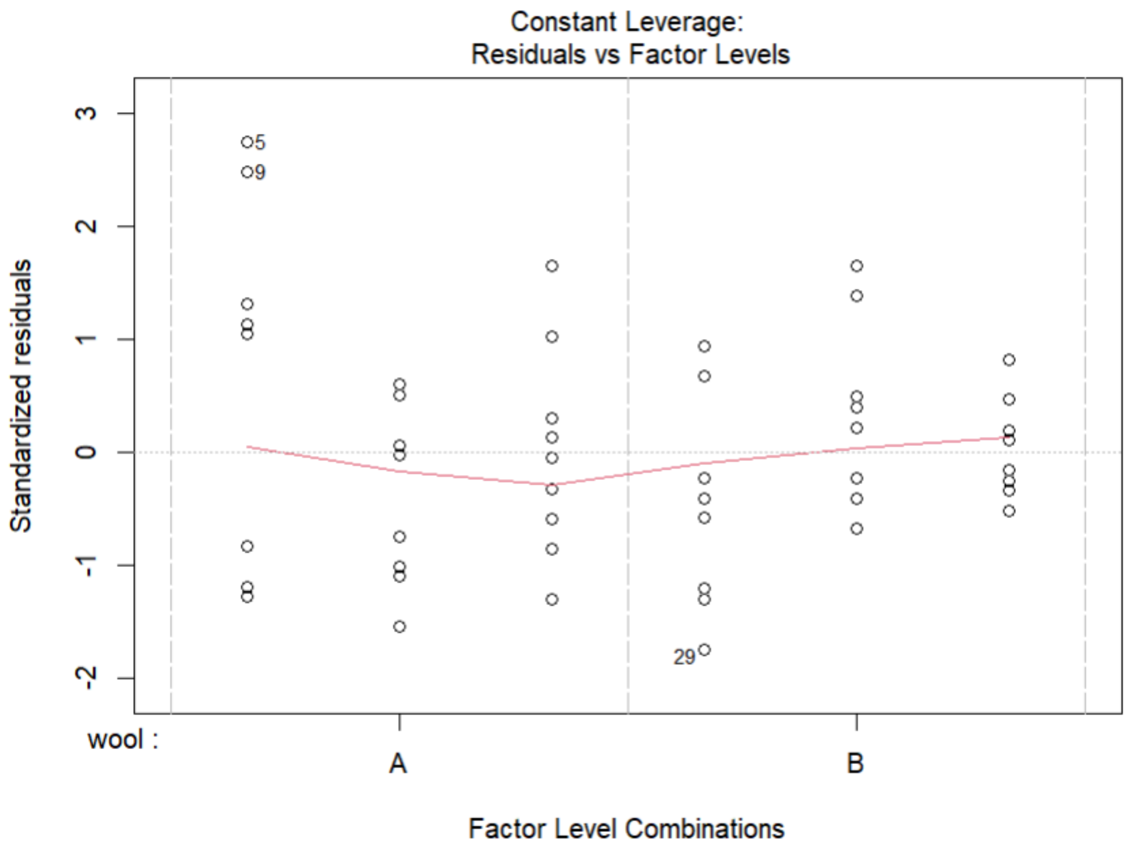
```
> breaks=warpbreaks$breaks
> mean(breaks)
[1] 28.14815
```



```
> Model_1<-aov(breaks~wool+tension,data=warpbreaks)
> summary(Model_1)
            Df Sum Sq Mean Sq F value  Pr(>F)   
wool         1    451   450.7   3.339 0.07361 . 
tension      2   2034  1017.1   7.537 0.00138 **
Residuals   50   6748   135.0                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```




Q-Q Residuals
Standardized residuals
Theoretical Quantiles
aov(breaks ~ wool + tension)
3
2
1
0
-1
-2
-1
0
1
2
29
9
5

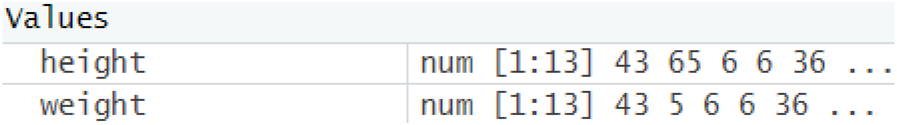


```
data("warpbreaks")
#explore data
head(warpbreaks)
summary(warpbreaks)
#extract variable
breaks=warpbreaks$breaks
mean(breaks)
Model_1<-aov(breaks~wool+tension,data=warpbreaks)
summary(Model_1)
plot(Model_1)
```

Practical 5

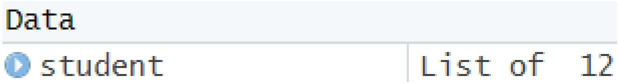
Linear Regression

```
height <- c(43,65,6,6,36,36,56,43556,43,7564,7,4764,75)
weight <- c(43,5,6,6,36,465,65,7,65467,547,647,45,7)
```



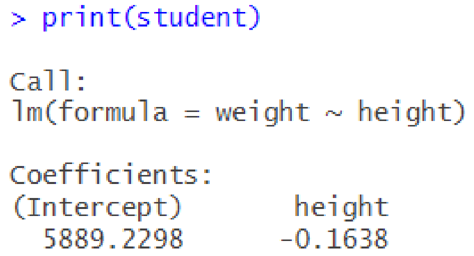
```
Values
  height        num [1:13] 43 65 6 6 36 ...
  weight        num [1:13] 43 5 6 6 36 ...
```

```
student <- lm(weight~height)
```



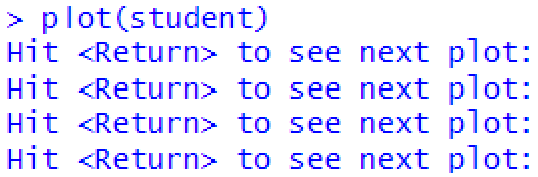
```
Data
  student       List of  12
```

```
print(student)
```

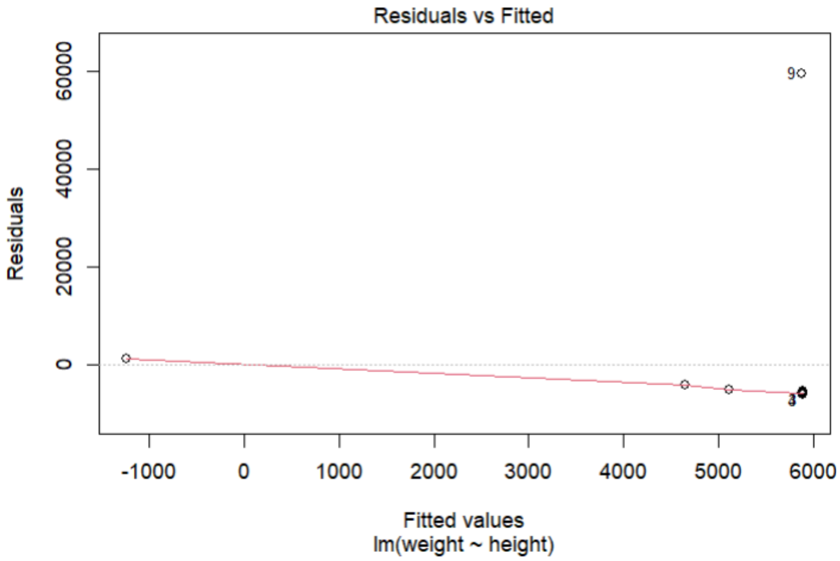


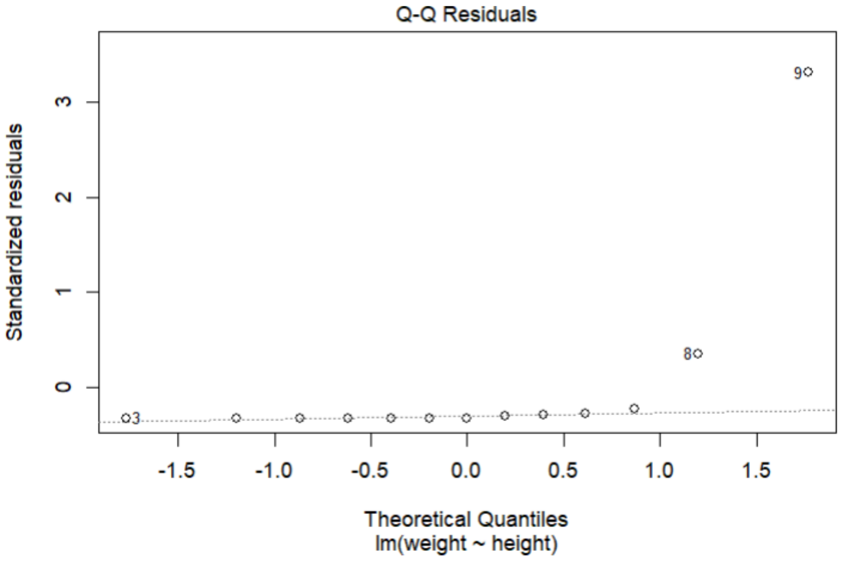
```
> print(student)

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
  5889.2298      -0.1638
```

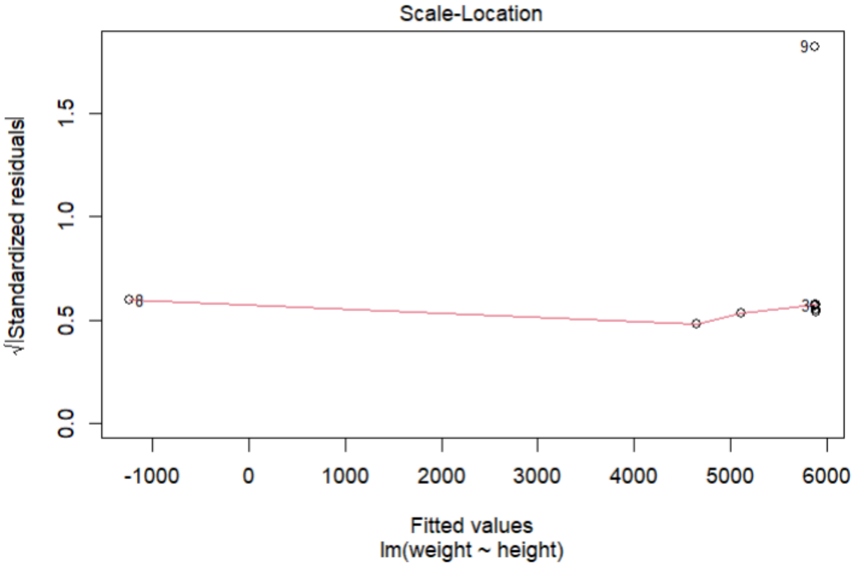
```
plot(student)
```

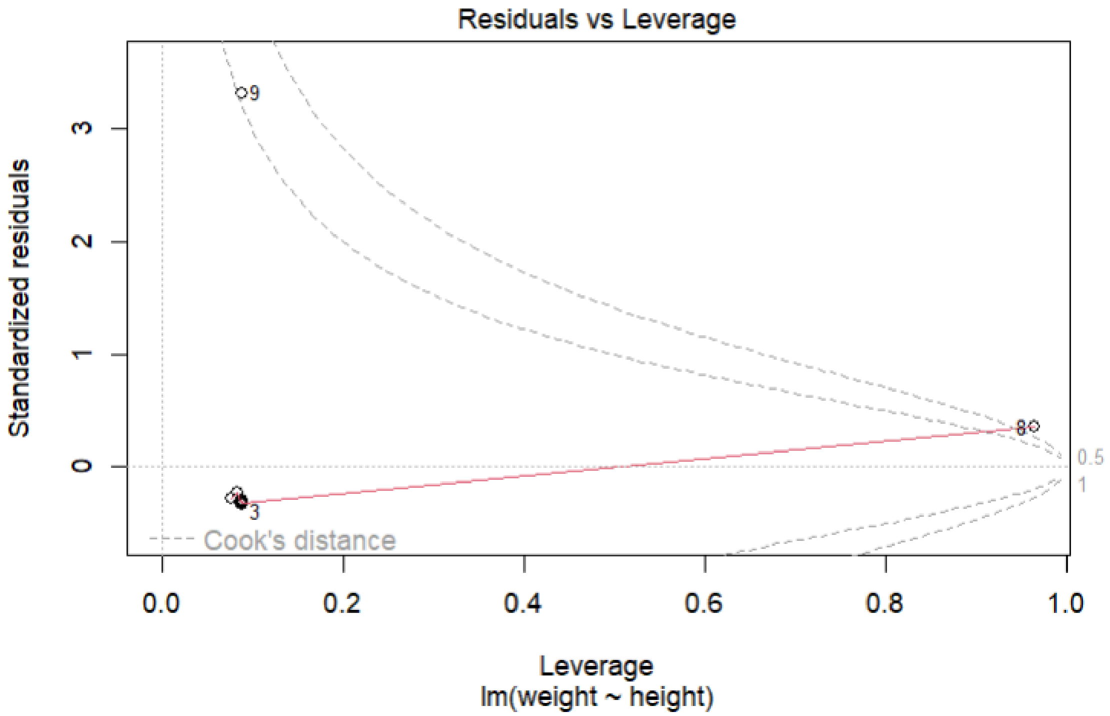


```
> plot(student)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

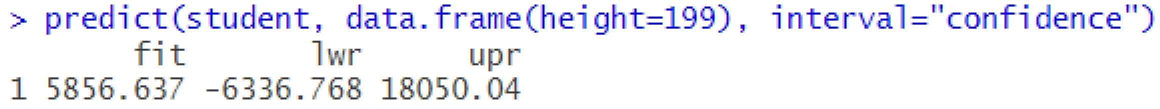

Residuals vs Fitted
Residuals
60000
40000
20000
0
-1000
0
1000
2000
3000
4000
5000
6000
9
3
Fitted values
lm(weight ~ height)


Q-Q Residuals
Standardized residuals
3
2
1
0
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
9
8
3
Theoretical Quantiles
lm(weight ~ height)


Scale-Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
-1000
0
1000
2000
3000
4000
5000
6000
9
8
3
Fitted values
lm(weight ~ height)

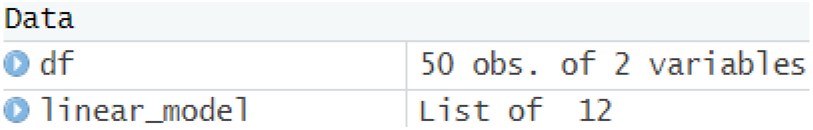


```
predict(student, data.frame(height=199), interval="confidence")
```



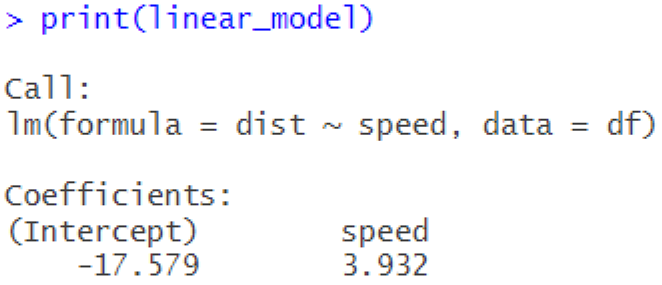
```
> predict(student, data.frame(height=199), interval="confidence")
       fit       lwr      upr
1 5856.637 -6336.768 18050.04
```

```
df <- datasets::cars
linear_model <- lm(dist~speed, data=df)
```



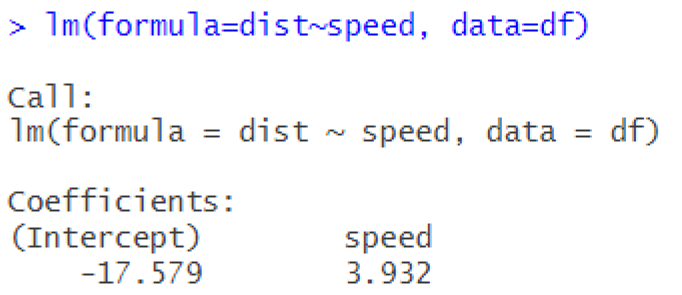
Data	
df	50 obs. of 2 variables
linear_model	List of 12

```
print(linear_model)
```



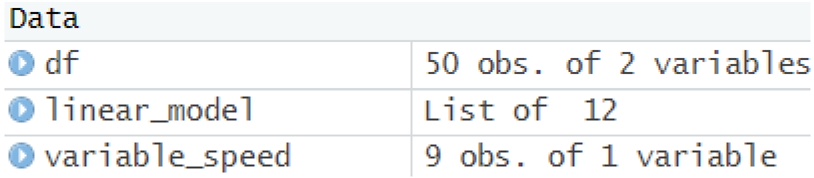
```
> print(linear_model)

Call:
lm(formula = dist ~ speed, data = df)

Coefficients:
(Intercept)        speed
    -17.579        3.932
```

```
lm(formula=dist~speed, data=df)
```



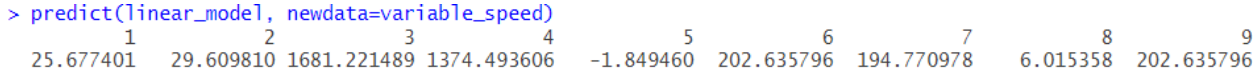
```
> lm(formula=dist~speed, data=df)

Call:
lm(formula = dist ~ speed, data = df)

Coefficients:
(Intercept)        speed
    -17.579        3.932
```

```
variable_speed <-
data.frame(speed=c(11,12,432,354,4,56,54,6,56))
linear_model <- lm(dist~speed, data=df)
```



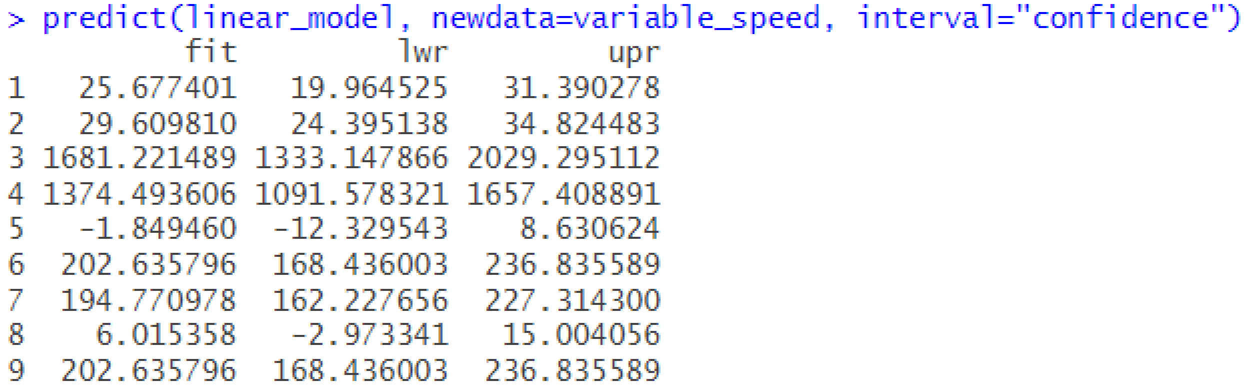
Data	
df	50 obs. of 2 variables
linear_model	List of 12
variable_speed	9 obs. of 1 variable

```
predict(linear_model, newdata=variable_speed)
```



```
> predict(linear_model, newdata=variable_speed)
          1           2           3           4           5           6           7           8           9
  25.677401   29.609810 1681.221489 1374.493606   -1.849460  202.635796  194.770978    6.015358  202.635796
```

```
predict(linear_model, newdata=variable_speed,
interval="confidence")
```



```
> predict(linear_model, newdata=variable_speed, interval="confidence")
          fit         lwr         upr
1   25.677401   19.964525   31.390278
2   29.609810   24.395138   34.824483
3 1681.221489 1333.147866 2029.295112
4 1374.493606 1091.578321 1657.408891
5   -1.849460  -12.329543    8.630624
6  202.635796  168.436003  236.835589
7  194.770978  162.227656  227.314300
8    6.015358   -2.973341   15.004056
9  202.635796  168.436003  236.835589
```

Practical 6

Hypothesis Testing

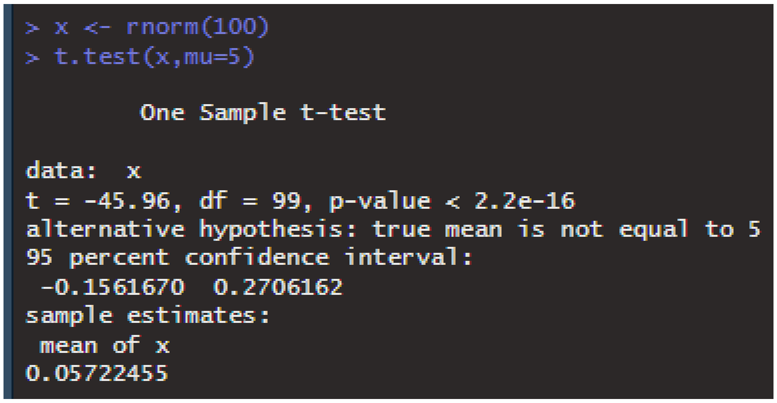
Functions:
rnorm() - to generate random numbers from a normal distribution.
t.test() - to test the statistical difference between a sample mean and a known or assumed mean.

1] One sample testing

- Syntax: t.test(x, mu)
- x: numeric vector
- mu: mean variable value

Unset

```
x <- rnorm(100)
t.test(x,mu=5)
```



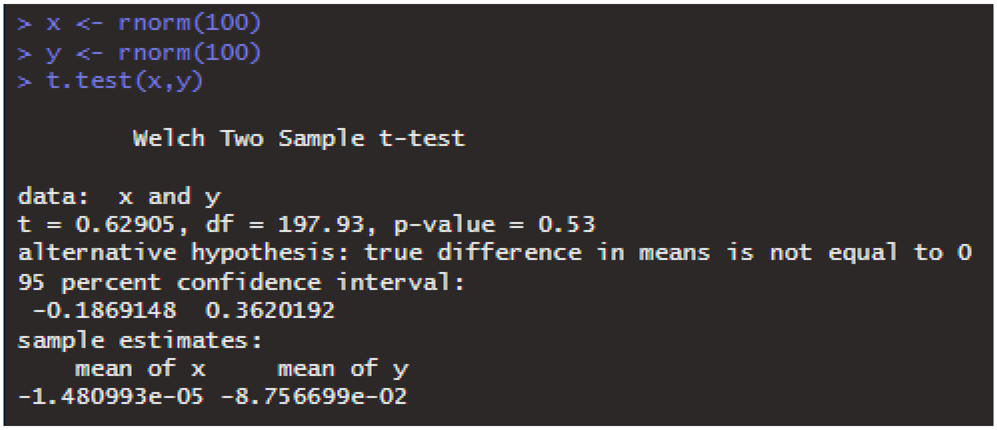
```
> x <- rnorm(100)
> t.test(x,mu=5)

	One Sample t-test

data:  x
t = -45.96, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 -0.1561670  0.2706162
sample estimates:
 mean of x
0.05722455
```

2] Two sample testing

- Syntax: t.test(x, y)
- x and y: numeric vector

Unset

```
x <- rnorm(100)
y <- rnorm(100)
t.test(x,y)
```

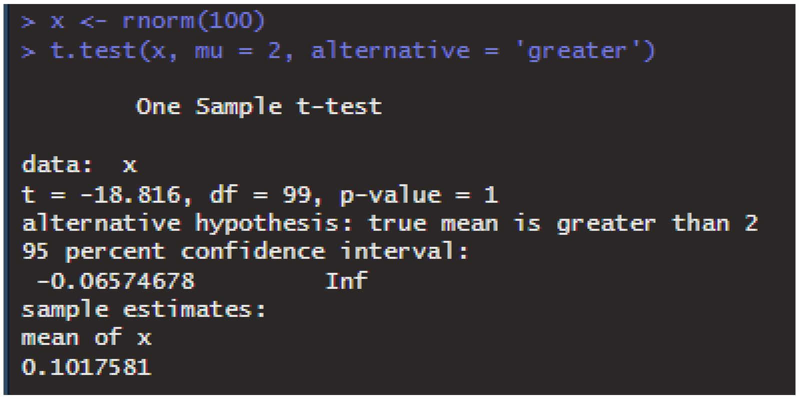


```
> x <- rnorm(100)
> y <- rnorm(100)
> t.test(x,y)

        Welch Two Sample t-test

data:  x and y
t = 0.62905, df = 197.93, p-value = 0.53
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.1869148  0.3620192
sample estimates:
    mean of x     mean of y
-1.480993e-05 -8.756699e-02
```

2] Directional hypothesis

Unset

```
x <- rnorm(100)
t.test(x, mu = 2, alternative = 'greater')
```



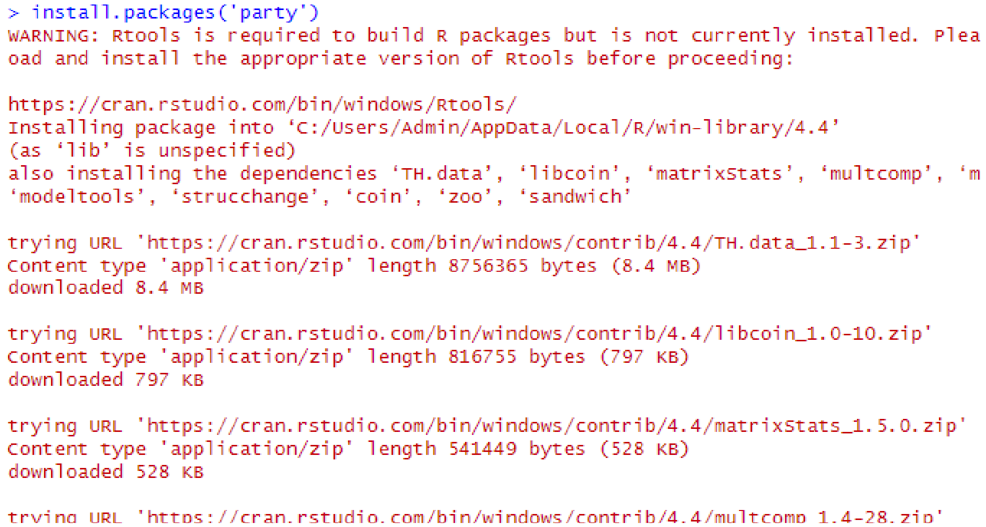
```
> x <- rnorm(100)
> t.test(x, mu = 2, alternative = 'greater')

        One Sample t-test

data:  x
t = -18.816, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 2
95 percent confidence interval:
 -0.06574678         Inf
sample estimates:
mean of x
0.1017581
```

Practical 7

Decision Tree

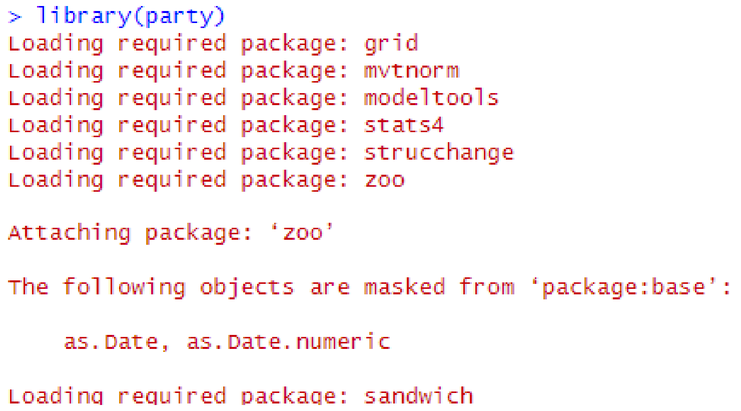
Functions:
install.packages() - to install packages from official R repositories.
library() - load packages.
ctree() - to create conditional inference trees, which are a type of decision tree.

```
install.packages('party')
```



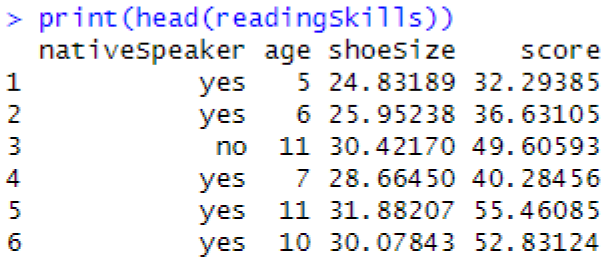
```
> install.packages('party')
WARNING: Rtools is required to build R packages but is not currently installed. Plea
oad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Admin/AppData/Local/R/win-library/4.4’
(as ‘lib’ is unspecified)
also installing the dependencies ‘TH.data’, ‘libcoin’, ‘matrixStats’, ‘multcomp’, ‘m
‘modeltools’, ‘strucchange’, ‘coin’, ‘zoo’, ‘sandwich’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/TH.data_1.1-3.zip'
Content type 'application/zip' length 8756365 bytes (8.4 MB)
downloaded 8.4 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/libcoin_1.0-10.zip'
Content type 'application/zip' length 816755 bytes (797 KB)
downloaded 797 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/matrixStats_1.5.0.zip'
Content type 'application/zip' length 541449 bytes (528 KB)
downloaded 528 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/multcomp_1.4-28.zip'
```

```
library(party)
```



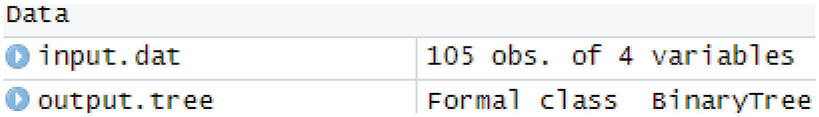
```
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: sandwich
```

```
print(head(readingSkills))
```

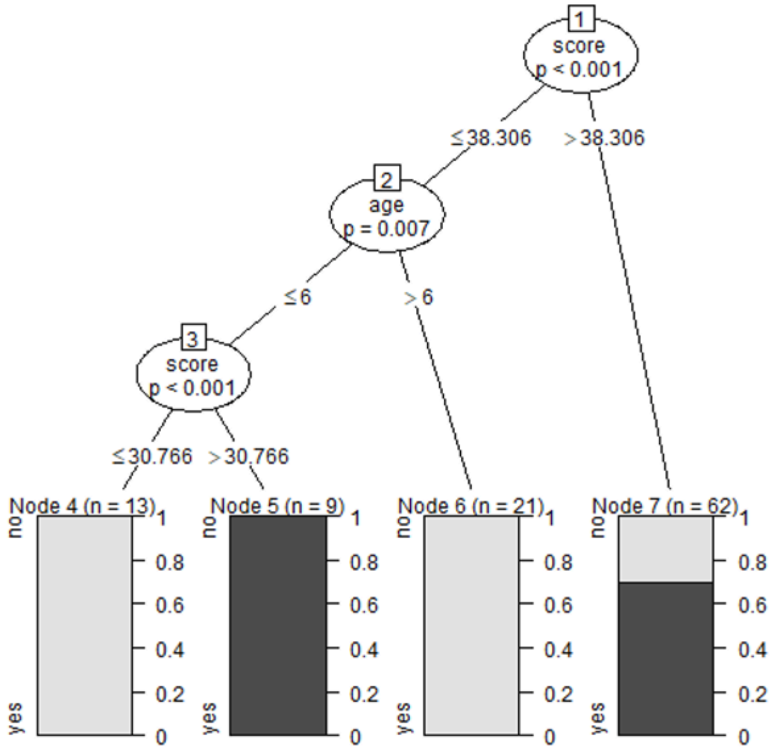


```
> print(head(readingSkills))
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
```

```
#create input dataframe
input.dat <- readingSkills[c(1:105),]
#give the chart file a name
png(file="decision_tree.png")
#create the tree
output.tree <- ctree(nativeSpeaker~age + shoeSize + score,
data=input.dat)
#plot the tree
plot(output.tree)
dev.off()
```



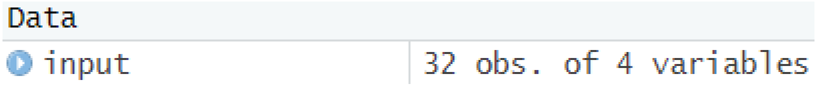
Data	
input.dat	105 obs. of 4 variables
output.tree	Formal class BinaryTree



Practical 8

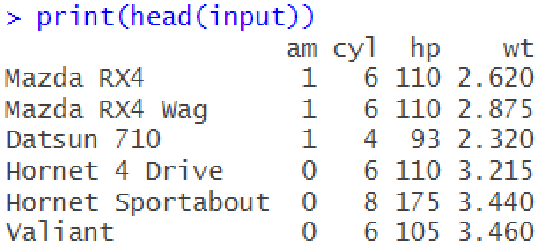
Logistic Regression

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



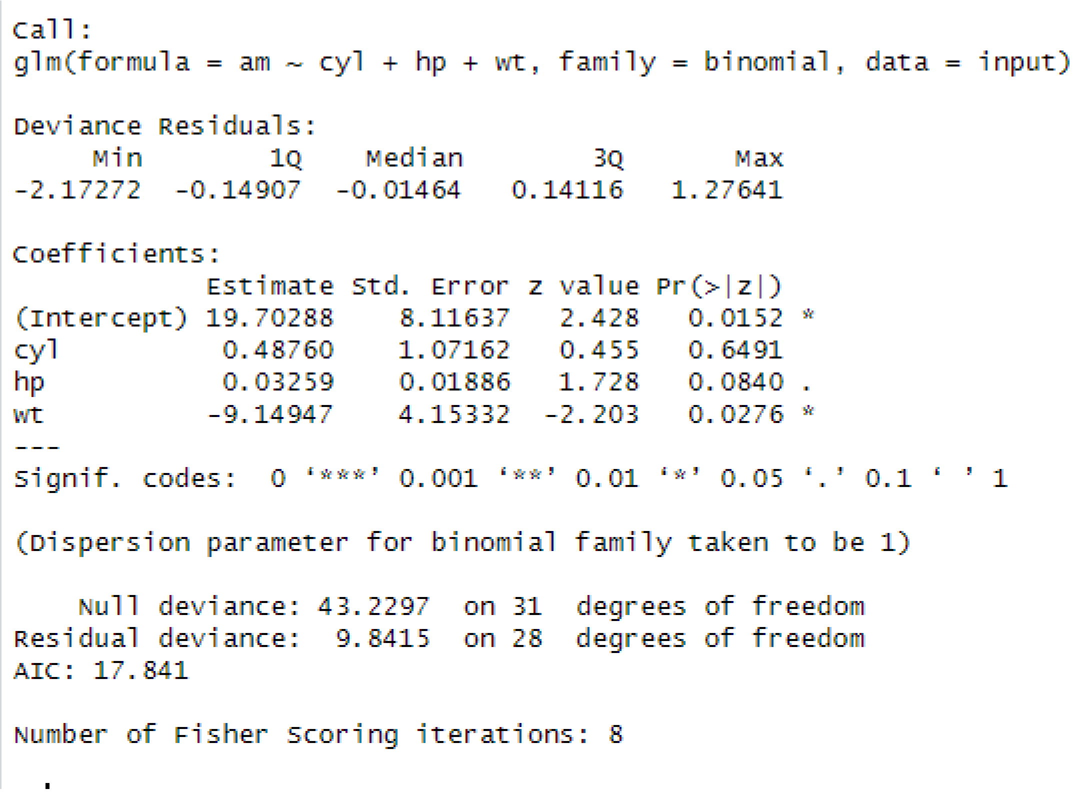
Data	
input	32 obs. of 4 variables

```
print(head(input))
```



```
> print(head(input))
                  am cyl  hp    wt
Mazda RX4          1   6 110 2.620
Mazda RX4 Wag      1   6 110 2.875
Datsun 710         1   4  93 2.320
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
```

```
am.data = glm(formula = am~cyl+hp+wt, data=input,
family=binomial)
print(summary(am.data))
```

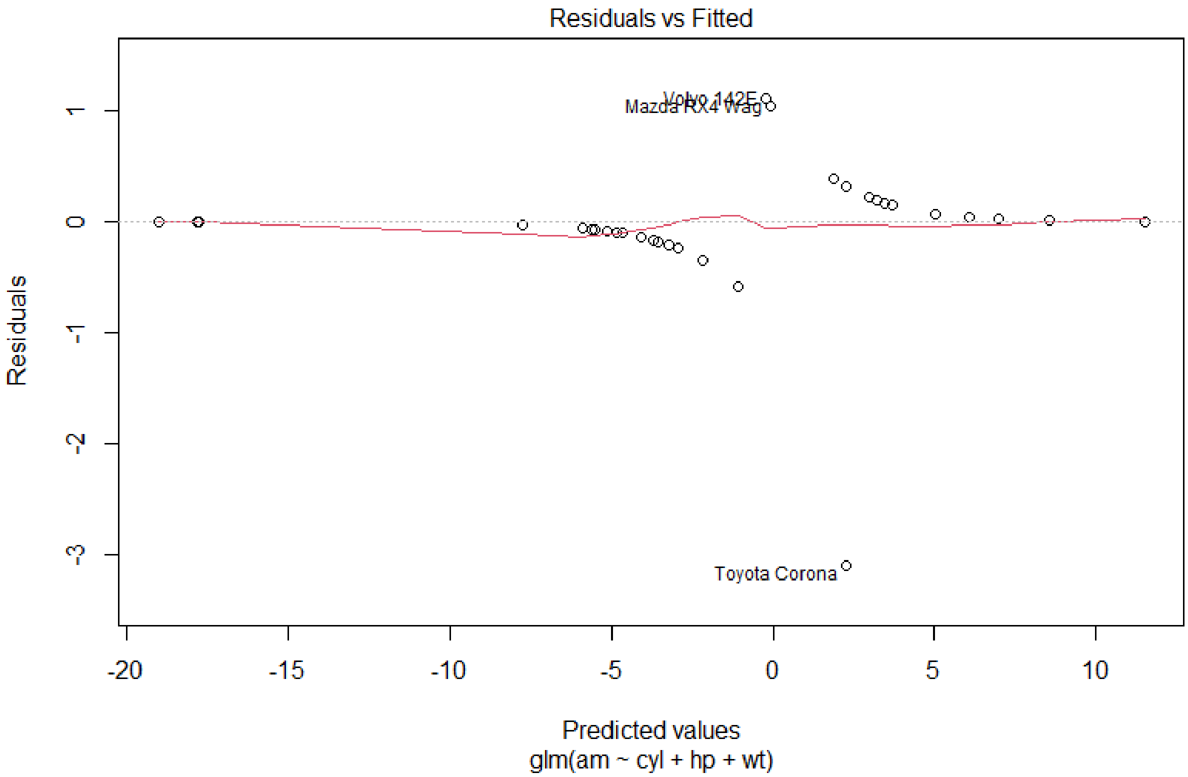


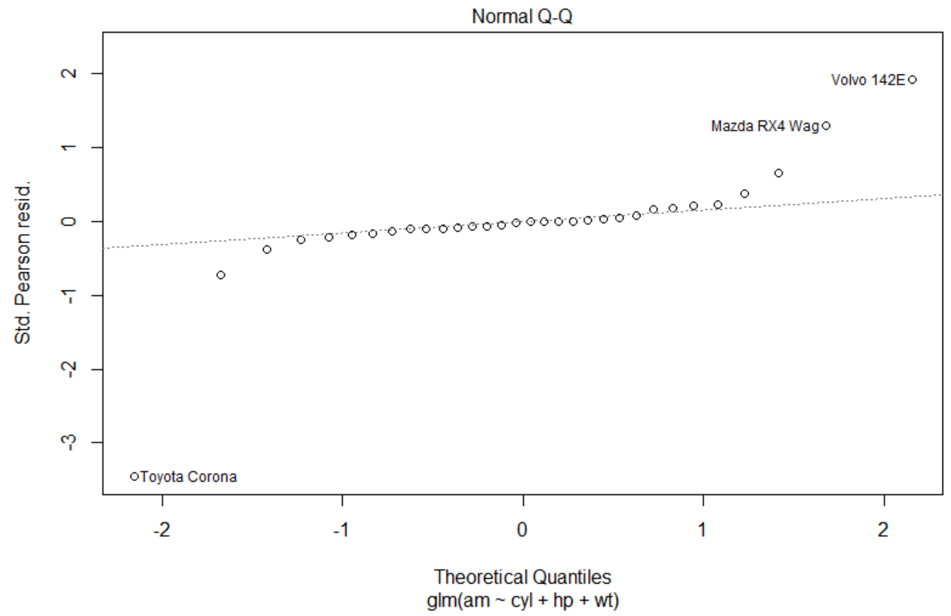
```
Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.17272  -0.14907  -0.01464   0.14116   1.27641  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491  
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.2297  on 31  degrees of freedom
Residual deviance:  9.8415  on 28  degrees of freedom
AIC: 17.841

Number of Fisher Scoring iterations: 8
```

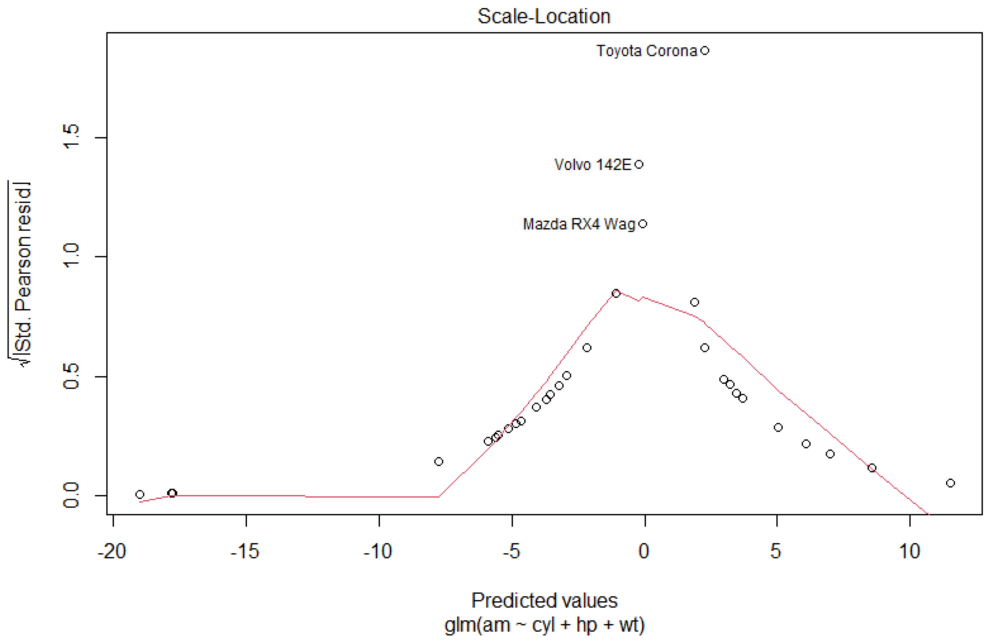
```
print(am.data)
```

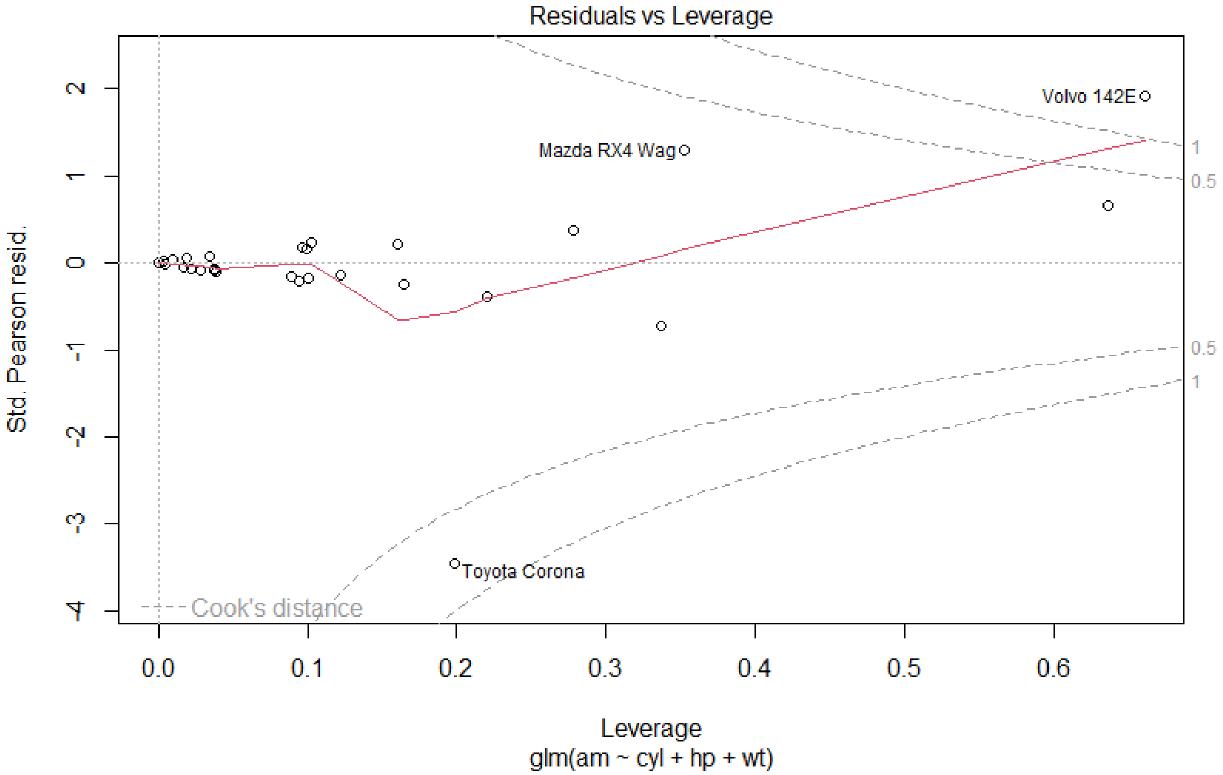
```
Call:  glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Coefficients:
(Intercept)          cyl           hp           wt  
   19.70288      0.48760      0.03259     -9.14947  

Degrees of Freedom: 31 Total (i.e. Null);  28 Residual
Null Deviance:      43.23 
Residual Deviance: 9.841        AIC: 17.84
```

```
plot(am.data)
```



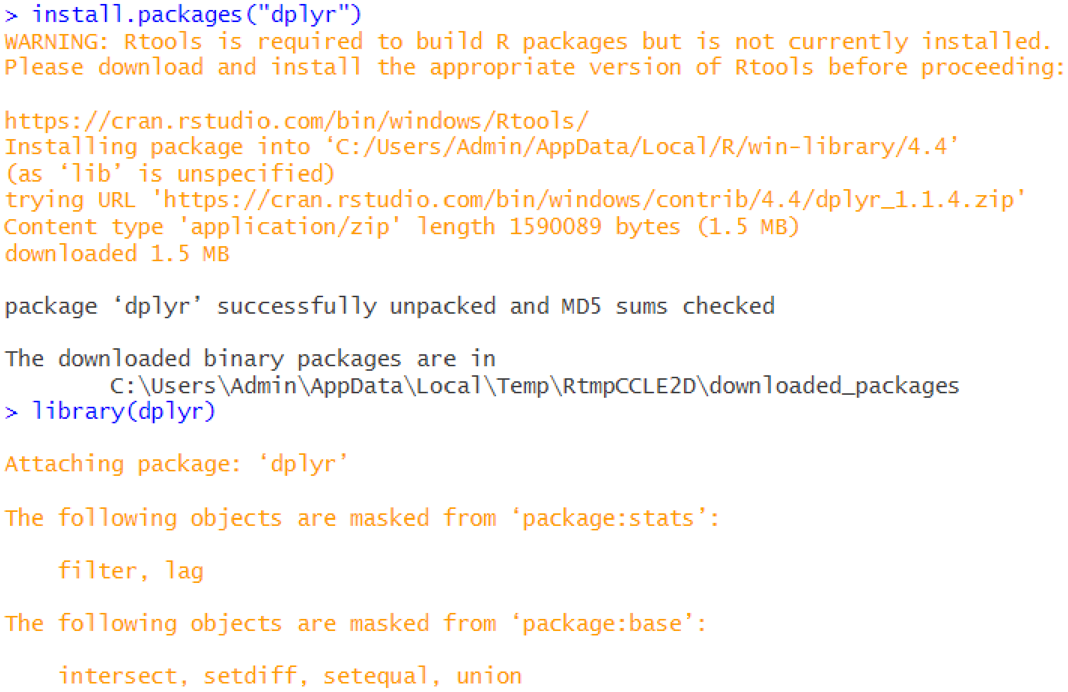

Normal Q-Q
Std. Pearson resid.
2
1
0
-1
-2
-3
Volvo 142E
Mazda RX4 Wag
Toyota Corona
-2
-1
0
1
2
Theoretical Quantiles
glm(am ~ cyl + hp + wt)


Scale-Location
√|Std. Pearson resid|
1.5
1.0
0.5
0.0
Toyota Corona
Volvo 142E
Mazda RX4 Wag
-20
-15
-10
-5
0
5
10
Predicted values
glm(am ~ cyl + hp + wt)



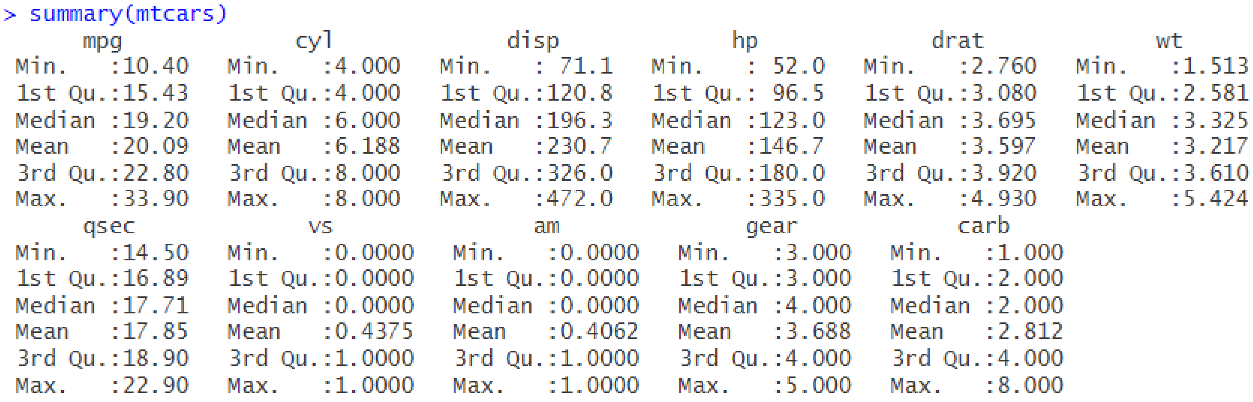
2]

```
install.packages("dplyr")
library(dplyr)
```



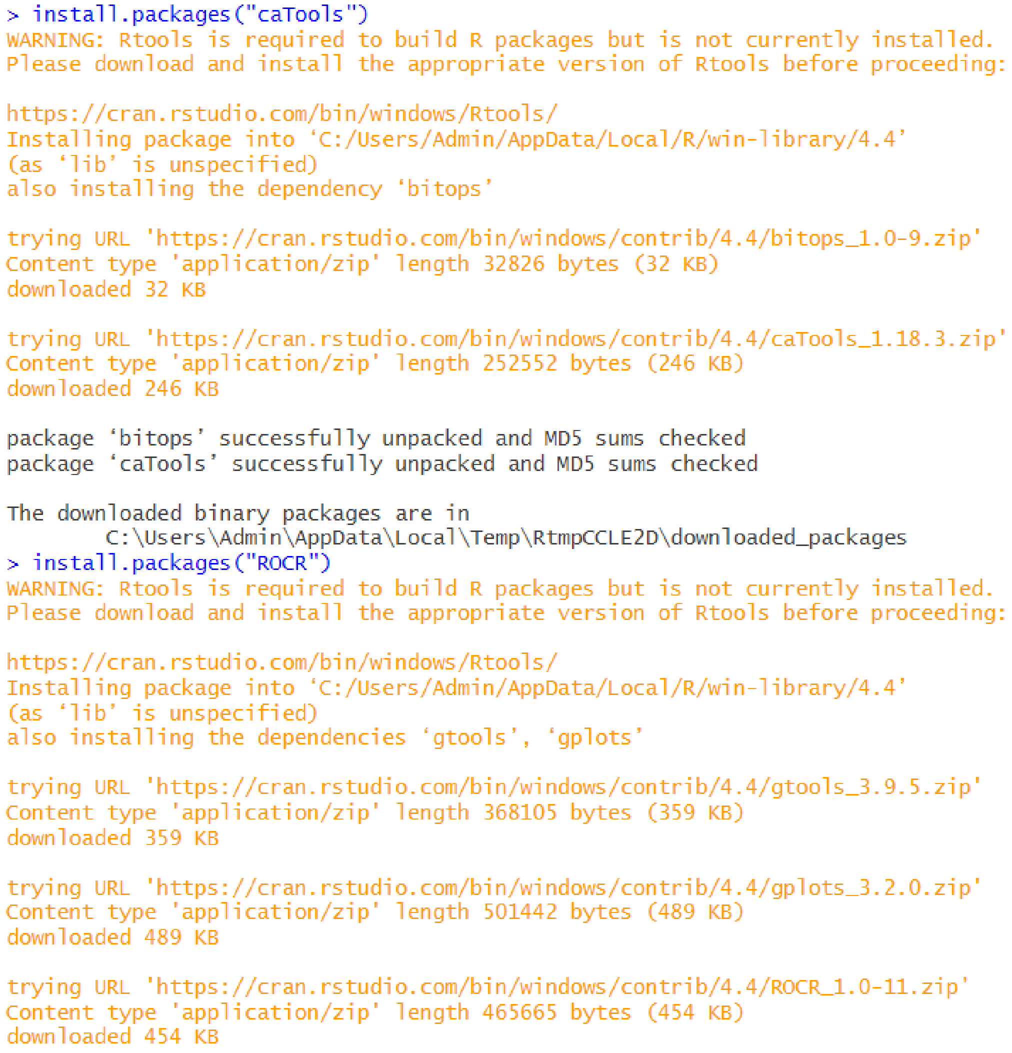
```
> install.packages("dplyr")
WARNING: Rtools is required to build R packages but is not currently installed.
Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Admin/AppData/Local/R/win-library/4.4’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/dplyr_1.1.4.zip'
Content type 'application/zip' length 1590089 bytes (1.5 MB)
downloaded 1.5 MB

package ‘dplyr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Admin\AppData\Local\Temp\RtmpCCLE2D\downloaded_packages
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

```
summary(mtcars)
```



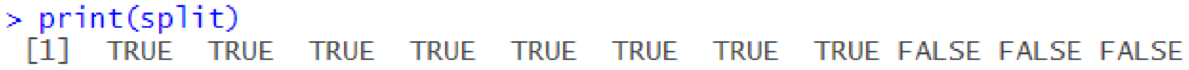
```
> summary(mtcars)
      mpg             cyl             disp             hp             drat             wt       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760   Min.   :1.513  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695   Median :3.325  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597   Mean   :3.217  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930   Max.   :5.424  
      qsec             vs               am              gear            carb      
 Min.   :14.50   Min.   :0.0000   Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :17.71   Median :0.0000   Median :0.0000   Median :4.000   Median :2.000  
 Mean   :17.85   Mean   :0.4375   Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :22.90   Max.   :1.0000   Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

```
install.packages("caTools")
install.packages("ROCR")
```



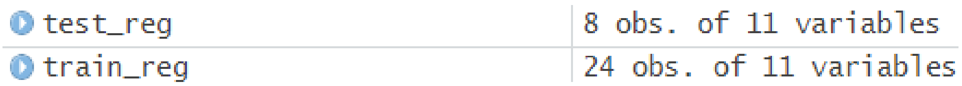
```
> install.packages("caTools")
WARNING: Rtools is required to build R packages but is not currently installed.
Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Admin/AppData/Local/R/win-library/4.4’
(as ‘lib’ is unspecified)
also installing the dependency ‘bitops’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/bitops_1.0-9.zip'
Content type 'application/zip' length 32826 bytes (32 KB)
downloaded 32 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/caTools_1.18.3.zip'
Content type 'application/zip' length 252552 bytes (246 KB)
downloaded 246 KB

package ‘bitops’ successfully unpacked and MD5 sums checked
package ‘caTools’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Admin\AppData\Local\Temp\RtmpCCLE2D\downloaded_packages
> install.packages("ROCR")
WARNING: Rtools is required to build R packages but is not currently installed.
Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Admin/AppData/Local/R/win-library/4.4’
(as ‘lib’ is unspecified)
also installing the dependencies ‘gtools’, ‘gplots’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/gtools_3.9.5.zip'
Content type 'application/zip' length 368105 bytes (359 KB)
downloaded 359 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/gplots_3.2.0.zip'
Content type 'application/zip' length 501442 bytes (489 KB)
downloaded 489 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/ROCR_1.0-11.zip'
Content type 'application/zip' length 465665 bytes (454 KB)
downloaded 454 KB
```

```
library(caTools)
library(ROCR)

split <- sample.split(mtcars, SplitRatio=0.8)
split
```



```
> print(split)
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

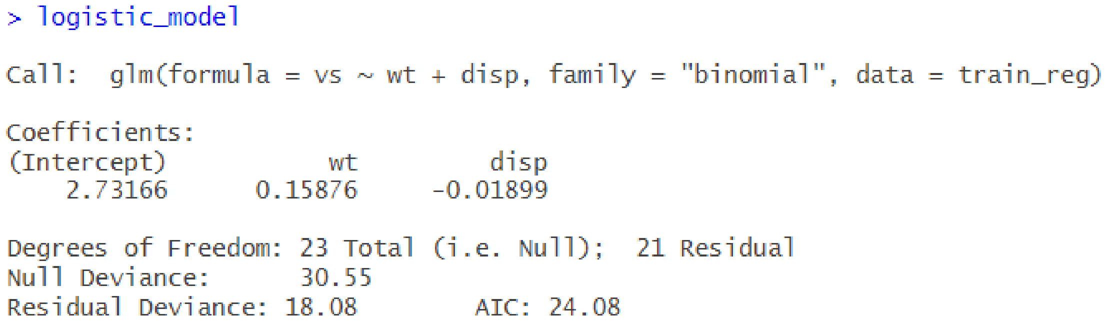
```
train_reg <- subset(mtcars, split=="TRUE")
test_reg <- subset(mtcars, split=="FALSE")
```



test_reg	8 obs. of 11 variables
train_reg	24 obs. of 11 variables

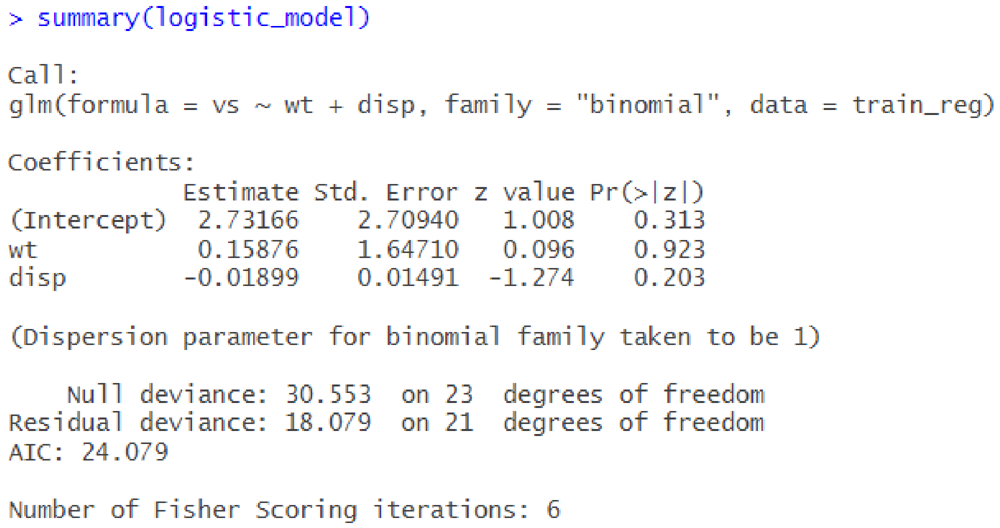
```
logistic_model <- glm(vs~wt+disp, data=train_reg,
family="binomial")
```

```
logistic_model
```



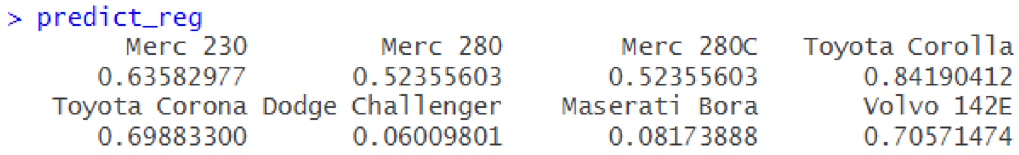
```
> logistic_model

Call:  glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Coefficients:
(Intercept)           wt         disp  
    2.73166      0.15876     -0.01899  

Degrees of Freedom: 23 Total (i.e. Null);  21 Residual
Null Deviance:	    30.55 
Residual Deviance: 18.08 	AIC: 24.08
```

```
summary(logistic_model)
```



```
> summary(logistic_model)

Call:
glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.73166    2.70940   1.008    0.313
wt           0.15876    1.64710   0.096    0.923
disp        -0.01899    0.01491  -1.274    0.203

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 30.553  on 23  degrees of freedom
Residual deviance: 18.079  on 21  degrees of freedom
AIC: 24.079

Number of Fisher Scoring iterations: 6
```

```
predict_reg <- predict(logistic_model, test_reg,
type="response")
predict_reg
```



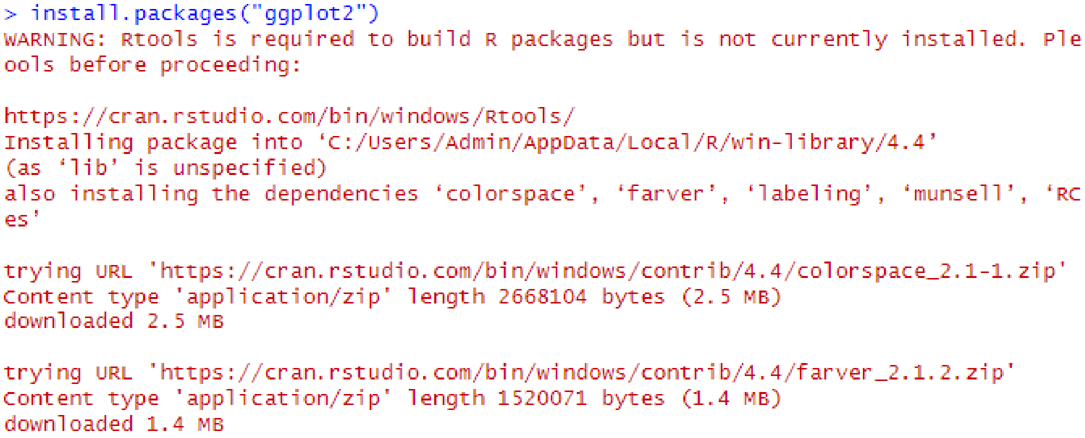
```
> predict_reg
        Merc 230         Merc 280        Merc 280C   Toyota Corolla 
      0.63582977       0.52355603       0.52355603       0.84190412 
   Toyota Corona Dodge Challenger    Maserati Bora       Volvo 142E 
      0.69883300       0.06009801       0.08173888       0.70571474 
```

Practical 9

K - Means Clustering

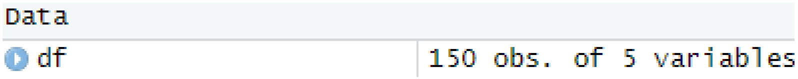
Functions:
aes() - to define aesthetic mappings for plots.
ggplot() - for creating visually appealing and complex plots.
geom_point() - specifically used to create scatter plots.
set.seed() - to initialize a pseudorandom number generator.
kmeans() - for performing k-means clustering.
clusplot() - to create a visual representation of clustering results.

```
install.packages("ggplot2")
```



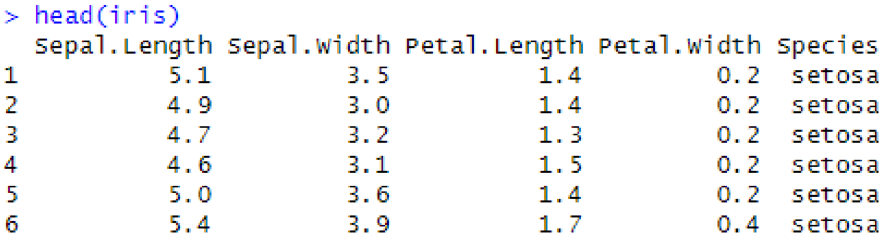
```
> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently installed. Ple
ools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
also installing the dependencies 'colorspace', 'farver', 'labeling', 'munsell', 'RC
es'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/colorspace_2.1-1.zip'
Content type 'application/zip' length 2668104 bytes (2.5 MB)
downloaded 2.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/farver_2.1.2.zip'
Content type 'application/zip' length 1520071 bytes (1.4 MB)
downloaded 1.4 MB
```

```
df <- iris
```



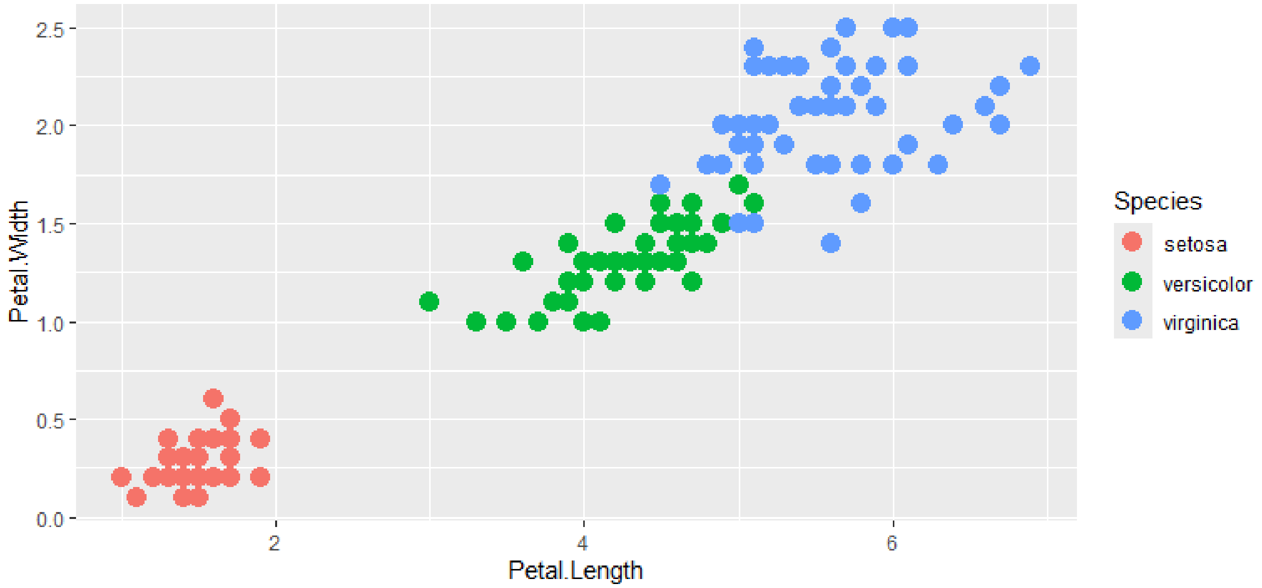
```
Data
df                          150 obs. of 5 variables
```

```
head(iris)
```

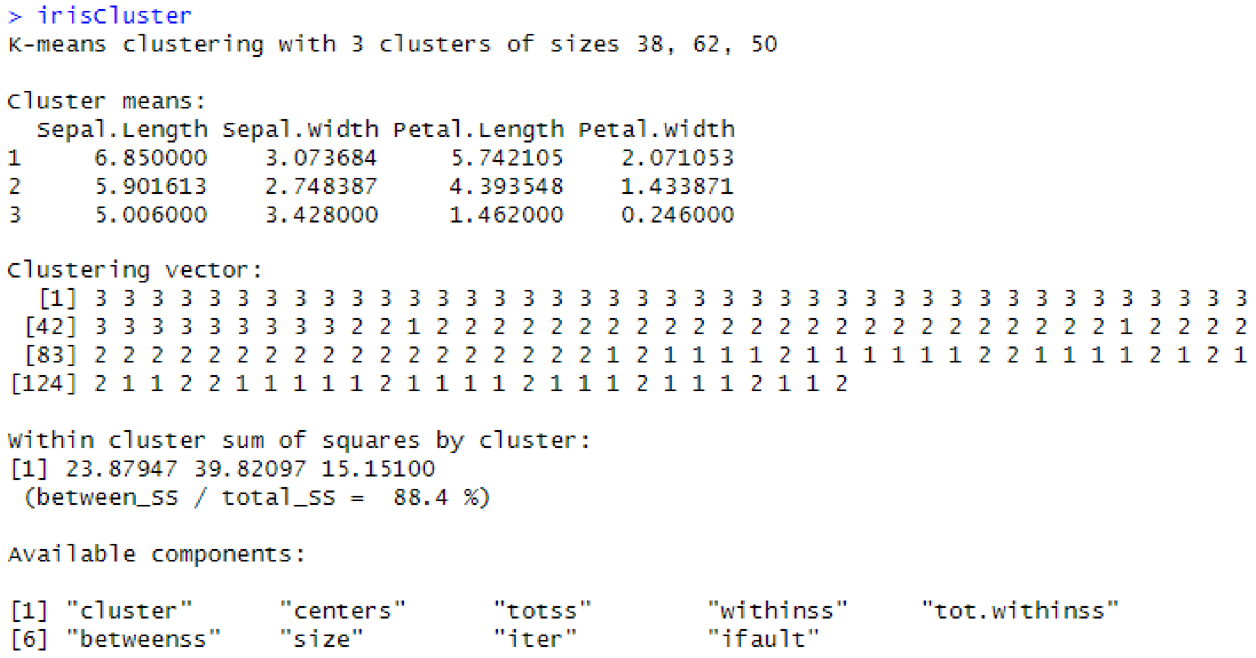


```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
ggplot(df,aes(Petal.Length,Petal.Width))+geom_point(aes(col=Spec
ies),size=4)
# aes (aesthetic) - how data is mapped to the visualization
```

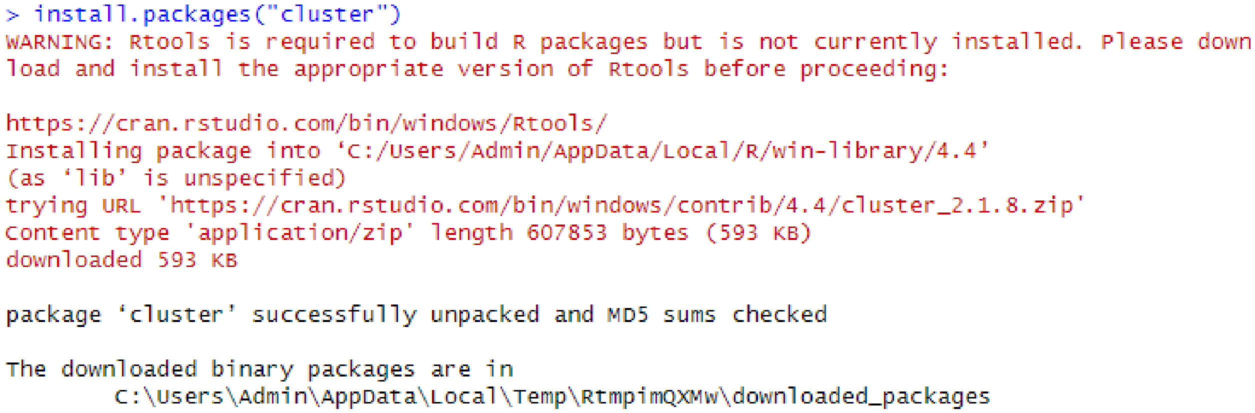


```
set.seed(101) # generate random no.s from 101
irisCluster <- kmeans(df[,1:4],center=3,nstart=20)
irisCluster
```



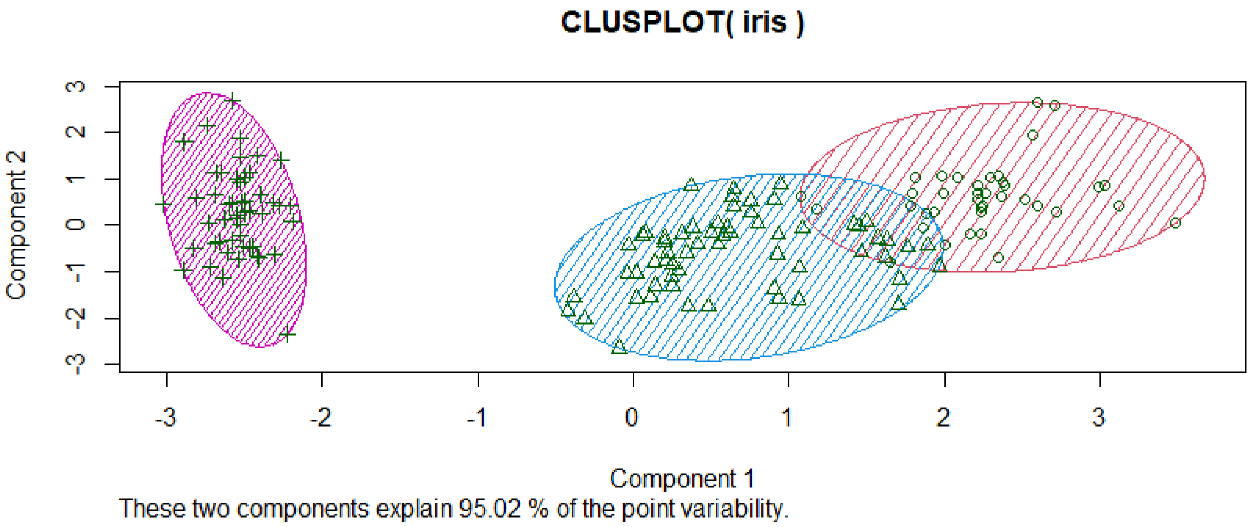
```
> irisCluster
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [42] 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
 [83] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1
[124] 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```
install.packages("cluster")
```



```
> install.packages("cluster")
WARNING: Rtools is required to build R packages but is not currently installed. Please down
load and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Admin/AppData/Local/R/win-library/4.4’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/cluster_2.1.8.zip'
Content type 'application/zip' length 607853 bytes (593 KB)
downloaded 593 KB

package ‘cluster’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Admin\AppData\Local\Temp\RtmpimQXMw\downloaded_packages
```

```
library(cluster)
clusplot(iris,irisCluster$cluster,color=T,shade=T,labels=0,lines
=0)
```


CLUSPLOT(iris)
Component 2
Component 1
These two components explain 95.02 % of the point variability.