**AIM: Write R commands to use mathematical functions on R Console and Write R script to create a Calculator Application**

**DESCRIPTION:**

R has an array of mathematical functions.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| abs(x) | Takes the absolute value of x |
| log(x,base=y) | Takes the logarithm of x with base y; if base is not specified, returns the natural logarithm |
| exp(x) | Returns the exponential of x |
| sqrt(x) | Returns the square root of x |
| factorial(x) | Returns the factorial of x (x!) |

**PROGRAM:**

**Mathematical functions on R Console:**

> x <- 5.685

> abs(x)

[1] 5.685

> ceiling(x)

[1] 6

> floor(x)

[1] 5

> round(x)

[1] 6

> round(x, digits = 2)

[1] 5.68

> trunc(x)

[1] 5

> sqrt(x)

[1] 2.3843324

> exp(x)

[1] 294.4178

> log(x)

[1] 1.737831

> log(x,base = 2)

[1] 2.50716

> log2(x)

[1] 2.50716

> log(x,base = 10)

[1] 0.7547305

> log10(x)

[1] 0.7547305

> factorial(x)

[1] 402.2475

**Calculator Application Program:**

x <- as.integer(readline(prompt = "Enter operand 1: "))

y <- as.integer(readline(prompt = "Enter operand 2: "))

print("1. Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

op<-as.integer(readline(prompt = "Enter choice:"))

if(op==1)

{

print(x+y)

}

if(op==2)

{

print(x-y)

}

if(op==3)

{

print(x\*y)

}

if(op==4)

{

print(x/y)

}

**OUTPUT:**

> x<-as.integer(readline(prompt = "Enter operand 1:"))

Enter operand 1: 6

> y<-as.integer(readline(prompt = "Enter operand 2:"))

Enter operand 2: 8

> print("1.Add")

[1] "1.Add"

> print("2.Subtract")

[1] "2.Subtract"

> print("3.Multiply")

[1] "3.Multiply"

> print("4.Divide")

[1] "4.Divide"

> op<-as.integer(readline(prompt = "Enter choice:"))

Enter choice: 1

> if(op==1)

+ {

+ print(x+y)

+ }

[1] 14

> if(op==2)

+ {

+ print(x-y)

+ }

> if(op==3)

+ {

+ print(x\*y)

+ }

> if(op==4)

+ {

+ print(x/y)

+ }

**AIM: Write R script to find descriptive statistics using str, summary, subset, aggregate functions on MTCARS, CARS and IRIS dataset**

**DESCRIPTION:**

mtcars, cars and iris are the datasets whose descriptive statistics are found using different aggregate functions which are

**str:** str() function in R language is used for displaying the internal structure of a R object.

**summary:** summary() function produce a summary of all records in the found set, or the subsummary values for records in different groups.

**subset:** subset() function in R is used to create subsets of a data frame.

**aggregate:** aggregate() function is used to get the summary statistics of the data by group. The statistics include mean, min, sum, max etc.

**PROGRAM:**

data(mtcars)

nrow(mtcars)

ncol(mtcars)

mean(mtcars$cyl)

median(mtcars$cyl)

max(mtcars$cyl)

max(mtcars$cyl)-min(mtcars$cyl)

quantile(mtcars$mpg,0.25)

quantile(mtcars$mpg,0.75)

str(mtcars)

summary(mtcars)

subset(mtcars,hp>120)

aggregate(mtcars[,1:11],by = list(mtcars$vs),FUN = mean)

data(cars)

nrow(cars)

ncol(cars)

mean(cars$speed)

median(cars$speed)

max(cars$speed)

max(cars$speed)-min(cars$speed)

quantile(cars$speed,0.25)

quantile(cars$speed,0.75)

str(cars)

summary(cars)

subset(cars,dist<42)

aggregate(cars[,],by = list(cars$speed),FUN = mean)

data(iris)

nrow(iris)

ncol(iris)

mean(iris$Sepal.Length)

median(iris$Sepal.Length)

max(iris$Sepal.Length)

max(iris$Sepal.Length)-min(iris$Sepal.Length)

quantile(iris$Sepal.Length,0.25)

quantile(iris$Sepal.Length,0.75)

str(iris)

summary(iris)

subset(iris,Sepal.Length>7)

aggregate(iris[,1:4],by = list(iris$Species),FUN = median)

**OUTPUT:**

> data(mtcars)

> nrow(mtcars)

[1] 32

> ncol(mtcars)

[1] 11

> mean(mtcars$cyl)

[1] 6.1875

> median(mtcars$cyl)

[1] 6

> max(mtcars$cyl)

[1] 8

> max(mtcars$cyl)-min(mtcars$cyl)

[1] 4

> quantile(mtcars$mpg,0.25)

25%

15.425

> quantile(mtcars$mpg,0.75)

75%

22.8

> str(mtcars)

'data.frame': 32 obs. of 11 variables:

$ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...

$ cyl : num 6 6 4 6 8 6 8 4 4 6 ...

$ disp: num 160 160 108 258 360 ...

$ hp : num 110 110 93 110 175 105 245 62 95 123 ...

$ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...

$ wt : num 2.62 2.88 2.32 3.21 3.44 ...

$ qsec: num 16.5 17 18.6 19.4 17 ...

$ vs : num 0 0 1 1 0 1 0 1 1 1 ...

$ am : num 1 1 1 0 0 0 0 0 0 0 ...

$ gear: num 4 4 4 3 3 3 3 4 4 4 ...

$ carb: num 4 4 1 1 2 1 4 2 2 4 ...

> 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

> subset(mtcars,hp>120)

mpg cyl disp hp drat wt qsec vs am gear carb

Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2

Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4

Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4

Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4

Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3

Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3

Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3

Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4

Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4

Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4

Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2

AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2

Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4

Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2

Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4

Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6

Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8

> aggregate(mtcars[,1:11],by = list(mtcars$vs),FUN = mean)

Group.1 mpg cyl disp hp drat wt qsec vs am gear carb

1 0 16.61667 7.444444 307.1500 189.72222 3.392222 3.688556 16.69389 0 0.3333333 3.555556 3.611111

2 1 24.55714 4.571429 132.4571 91.35714 3.859286 2.611286 19.33357 1 0.5000000 3.857143 1.785714

> data(cars)

> nrow(cars)

[1] 50

> ncol(cars)

[1] 2

> mean(cars$speed)

[1] 15.4

> median(cars$speed)

[1] 15

> max(cars$speed)

[1] 25

> max(cars$speed)-min(cars$speed)

[1] 21

> quantile(cars$speed,0.25)

25%

12

> quantile(cars$speed,0.75)

75%

19

> str(cars)

'data.frame': 50 obs. of 2 variables:

$ speed: num 4 4 7 7 8 9 10 10 10 11 ...

$ dist : num 2 10 4 22 16 10 18 26 34 17 ...

> 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

> subset(cars,dist<42)

speed dist

1 4 2

2 4 10

3 7 4

4 7 22

5 8 16

6 9 10

7 10 18

8 10 26

9 10 34

10 11 17

11 11 28

12 12 14

13 12 20

14 12 24

15 12 28

16 13 26

17 13 34

18 13 34

20 14 26

21 14 36

24 15 20

25 15 26

27 16 32

28 16 40

29 17 32

30 17 40

36 19 36

39 20 32

> aggregate(cars[,],by = list(cars$speed),FUN = mean)

Group.1 speed dist

1 4 4 6.00000

2 7 7 13.00000

3 8 8 16.00000

4 9 9 10.00000

5 10 10 26.00000

6 11 11 22.50000

7 12 12 21.50000

8 13 13 35.00000

9 14 14 50.50000

10 15 15 33.33333

11 16 16 36.00000

12 17 17 40.66667

13 18 18 64.50000

14 19 19 50.00000

15 20 20 50.40000

16 22 22 66.00000

17 23 23 54.00000

18 24 24 93.75000

19 25 25 85.00000

> data(iris)

> nrow(iris)

[1] 150

> ncol(iris)

[1] 5

> mean(iris$Sepal.Length)

[1] 5.843333

> median(iris$Sepal.Length)

[1] 5.8

> max(iris$Sepal.Length)

[1] 7.9

> max(iris$Sepal.Length)-min(iris$Sepal.Length)

[1] 3.6

> quantile(iris$Sepal.Length,0.25)

25%

5.1

> quantile(iris$Sepal.Length,0.75)

75%

6.4

> str(iris)

'data.frame': 150 obs. of 5 variables:

$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...

$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...

$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...

$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

$ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

> summary(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

Min. : 4.300 Min. : 2.000 Min. : 1.000 Min. : 0.100 setosa : 50

1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:50

Median: 5.800 Median :3.000 Median :4.350 Median :1.300 virginica :50

Mean : 5.843 Mean :3.057 Mean :3.758 Mean :1.199

3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800

Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500

> subset(iris,Sepal.Length>7)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

103 7.1 3.0 5.9 2.1 virginica

106 7.6 3.0 6.6 2.1 virginica

108 7.3 2.9 6.3 1.8 virginica

110 7.2 3.6 6.1 2.5 virginica

118 7.7 3.8 6.7 2.2 virginica

119 7.7 2.6 6.9 2.3 virginica

123 7.7 2.8 6.7 2.0 virginica

126 7.2 3.2 6.0 1.8 virginica

130 7.2 3.0 5.8 1.6 virginica

131 7.4 2.8 6.1 1.9 virginica

132 7.9 3.8 6.4 2.0 virginica

136 7.7 3.0 6.1 2.3 virginica

> aggregate(iris[,1:4],by = list(iris$Species),FUN = median)

Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width

1 setosa 5.0 3.4 1.50 0.2

2 versicolor 5.9 2.8 4.35 1.3

3 virginica 6.5 3.0 5.55 2.0

**AIM: Write R script to read/import data from text file, CSV file, web URL, Excel data sheet and write/export data to the disk**

**DESCRIPTION:**

Data could exist in various formats. For each format [R](https://www.guru99.com/r-tutorial.html) has a specific function and argument.

One of the most widely data store is the .csv (comma-separated values) file formats. R loads

an array of libraries which is convenient to open csv files combined with the read.csv()

function.

read.csv(file, header = TRUE, sep = ",")

[Excel](https://www.guru99.com/excel-tutorials.html) files are very popular among data analysts. Spreadsheets are easy to work with and

flexible.

The function read\_excel() is of great use when it comes to opening xls and xlsx extention.

read\_excel(PATH, sheet = NULL, range= NULL, col\_names = TRUE)

The default delimiter of the read.csv() function is a comma, but you can use other delimiters by supplying the ‘sep’ argument to the function.

**PROGRAM:**

mydata<-read.table('mydataset.txt')

mydata

mydata<-read.table('mydataset.txt',header = TRUE)

mydata

mydata<-read.table('mydatasetcomma.txt',sep = ',')

mydata

mydata<-read.table('mydatasetcomma.txt',sep = ',',header = TRUE)

mydata

mydata<-read.csv('mydataset.csv')

mydata

mydata<-read.csv('mydataset.csv',nrows = 3)

mydata

mydata<-read.table('http://assets.datacamp.com/course/compfin/sbuxPrices.csv',sep = ',',header = T)

mydata

install.packages("readxl")

library(readxl)

mydata<-read\_excel("mydataset.xlsx",sheet = "Sheet1")

mydata

id<-c(100,101,102)

sname<-c("Mango","Apple","Berry")

cost<-c(200,100,300)

fruits<-data.frame(id,sname,cost)

fruits

write.csv(fruits,file = "Fruits")

**OUTPUT:**

> mydata<-read.table('mydataset.txt')

> mydata

V1 V2 V3 V4 V5

1 mpg cyl disp hp

2 Mazda RX4 21 6 160 110

3 Mazda RX4 Wag 21 6 160 110

4 Datsun 710 22.8 4 108 93

5 Hornet 4 Drive 21.4 6 258 110

6 Hornet Sportabout 18.7 8 360 175

7 Valiant 18.1 6 225 105

> mydata<-read.table('mydataset.txt',header = TRUE)

> mydata

X mpg cyl disp hp

1 Mazda RX4 21.0 6 160 110

2 Mazda RX4 Wag 21.0 6 160 110

3 Datsun 710 22.8 4 108 93

4 Hornet 4 Drive 21.4 6 258 110

5 Hornet Sportabout 18.7 8 360 175

6 Valiant 18.1 6 225 105

> mydata<-read.table('mydatasetcomma.txt',sep = ',')

> mydata

V1 V2 V3 V4 V5

1 mpg cyl disp hp

2 Mazda RX4 21 6 160 110

3 Mazda RX4 Wag 21 6 160 110

4 Datsun 710 22.8 4 108 93

5 Hornet 4 Drive 21.4 6 258 110

6 Hornet Sportabout 18.7 8 360 175

7 Valiant 18.1 6 225 105

> mydata<-read.table('mydatasetcomma.txt',sep = ',',header = TRUE)

> mydata

X mpg cyl disp hp

1 Mazda RX4 21.0 6 160 110

2 Mazda RX4 Wag 21.0 6 160 110

3 Datsun 710 22.8 4 108 93

4 Hornet 4 Drive 21.4 6 258 110

5 Hornet Sportabout 18.7 8 360 175

6 Valiant 18.1 6 225 105

> mydata<-read.csv('mydataset.csv')

> mydata

X mpg cyl disp hp drat

1 Mazda RX4 21.0 6 160 110 3.90

2 Mazda RX4 Wag 21.0 6 160 110 3.90

3 Datsun 710 22.8 4 108 93 3.85

4 Hornet 4 Drive 21.4 6 258 110 3.08

5 Hornet Sportabout 18.7 8 360 175 3.15

6 Valiant 18.1 6 225 105 2.76

> mydata<-read.csv('mydataset.csv',nrows = 3)

> mydata

X mpg cyl disp hp drat

1 Mazda RX4 21.0 6 160 110 3.90

2 Mazda RX4 Wag 21.0 6 160 110 3.90

3 Datsun 710 22.8 4 108 93 3.85

> mydata<-read.table('http://assets.datacamp.com/course/compfin/sbuxPrices.csv',sep = ',',header = T)

> mydata

Date Adj.Close

1 3/31/1993 1.13

2 4/1/1993 1.15

3 5/3/1993 1.43

4 6/1/1993 1.46

5 7/1/1993 1.41

6 8/2/1993 1.44

7 9/1/1993 1.63

8 10/1/1993 1.59

9 11/1/1993 1.32

10 12/1/1993 1.32

11 1/3/1994 1.43

12 2/1/1994 1.38

13 3/1/1994 1.45

14 4/4/1994 1.77

15 5/2/1994 1.69

16 6/1/1994 1.50

17 7/1/1994 1.72

18 8/1/1994 1.68

19 9/1/1994 1.37

20 10/3/1994 1.61

21 11/1/1994 1.59

22 12/1/1994 1.63

23 1/3/1995 1.43

24 2/1/1995 1.42

25 3/1/1995 1.43

26 4/3/1995 1.40

27 5/1/1995 1.73

28 6/1/1995 2.12

29 7/3/1995 2.22

30 8/1/1995 2.38

31 9/1/1995 2.25

32 10/2/1995 2.33

33 11/1/1995 2.51

34 12/1/1995 2.50

35 1/2/1996 1.99

36 2/1/1996 2.09

37 3/1/1996 2.77

38 4/1/1996 3.22

39 5/1/1996 3.22

40 6/3/1996 3.36

41 7/1/1996 3.09

42 8/1/1996 3.89

43 9/3/1996 3.92

44 10/1/1996 3.86

45 11/1/1996 4.12

46 12/2/1996 3.40

47 1/2/1997 4.07

48 2/3/1997 4.00

49 3/3/1997 3.52

50 4/1/1997 3.55

51 5/1/1997 3.74

52 6/2/1997 4.63

53 7/1/1997 4.87

54 8/1/1997 4.87

55 9/2/1997 4.97

56 10/1/1997 3.92

57 11/3/1997 4.15

58 12/1/1997 4.56

59 1/2/1998 4.35

60 2/2/1998 4.70

61 3/2/1998 5.39

62 4/1/1998 5.72

63 5/1/1998 5.71

64 6/1/1998 6.35

65 7/1/1998 4.98

66 8/3/1998 3.75

67 9/1/1998 4.30

68 10/1/1998 5.16

69 11/2/1998 5.48

70 12/1/1998 6.67

71 1/4/1999 6.19

72 2/1/1999 6.29

73 3/1/1999 6.67

74 4/1/1999 8.78

75 5/3/1999 8.77

76 6/1/1999 8.93

77 7/1/1999 5.53

78 8/2/1999 5.44

79 9/1/1999 5.89

80 10/1/1999 6.46

81 11/1/1999 6.31

82 12/1/1999 5.76

83 1/3/2000 7.61

84 2/1/2000 8.35

85 3/1/2000 10.65

86 4/3/2000 7.19

87 5/1/2000 8.08

88 6/1/2000 9.08

89 7/3/2000 8.91

90 8/1/2000 8.71

91 9/1/2000 9.52

92 10/2/2000 10.62

93 11/1/2000 10.83

94 12/1/2000 10.52

95 1/2/2001 11.87

96 2/1/2001 11.32

97 3/1/2001 10.09

98 4/2/2001 9.20

99 5/1/2001 9.28

100 6/1/2001 10.94

101 7/2/2001 8.58

102 8/1/2001 8.02

103 9/4/2001 7.10

104 10/1/2001 8.14

105 11/1/2001 8.42

106 12/3/2001 9.06

107 1/2/2002 11.30

108 2/4/2002 10.94

109 3/1/2002 11.00

110 4/1/2002 10.85

111 5/1/2002 11.54

112 6/3/2002 11.81

113 7/1/2002 9.33

114 8/1/2002 9.56

115 9/3/2002 9.82

116 10/1/2002 11.33

117 11/1/2002 10.34

118 12/2/2002 9.69

119 1/2/2003 10.80

120 2/3/2003 11.15

121 3/3/2003 12.25

122 4/1/2003 11.18

123 5/1/2003 11.73

124 6/2/2003 11.67

125 7/1/2003 12.99

126 8/1/2003 13.50

127 9/2/2003 13.69

128 10/1/2003 15.02

129 11/3/2003 15.30

130 12/1/2003 15.77

131 1/2/2004 17.41

132 2/2/2004 17.78

133 3/1/2004 18.01

134 4/1/2004 18.50

135 5/3/2004 19.30

136 6/1/2004 20.68

137 7/1/2004 22.34

138 8/2/2004 20.56

139 9/1/2004 21.61

140 10/1/2004 25.14

141 11/1/2004 26.75

142 12/1/2004 29.65

143 1/3/2005 25.67

144 2/1/2005 24.63

145 3/1/2005 24.56

146 4/1/2005 23.54

147 5/2/2005 26.05

148 6/1/2005 24.56

149 7/1/2005 24.98

150 8/1/2005 23.31

151 9/1/2005 23.82

152 10/3/2005 26.89

153 11/1/2005 28.95

154 12/1/2005 28.54

155 1/3/2006 30.14

156 2/1/2006 34.54

157 3/1/2006 35.78

158 4/3/2006 35.44

159 5/1/2006 33.90

160 6/1/2006 35.91

161 7/3/2006 32.55

162 8/1/2006 29.49

163 9/1/2006 32.38

164 10/2/2006 35.90

165 11/1/2006 33.56

166 12/1/2006 33.68

167 1/3/2007 33.22

168 2/1/2007 29.38

169 3/1/2007 29.82

170 4/2/2007 29.50

171 5/1/2007 27.40

172 6/1/2007 24.95

173 7/2/2007 25.37

174 8/1/2007 26.20

175 9/4/2007 24.91

176 10/1/2007 25.37

177 11/1/2007 22.24

178 12/3/2007 19.46

179 1/2/2008 17.98

180 2/1/2008 17.10

181 3/3/2008 16.64

> install.packages("readxl")

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/

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/readxl\_1.4.1.zip'

Content type 'application/zip' length 1663793 bytes (1.6 MB)

downloaded 1.6 MB

package ‘readxl’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Student\AppData\Local\Temp\RtmpOSUiyd\downloaded\_packages

> library(readxl)

> mydata<-read\_excel("mydataset.xlsx",sheet = "Sheet1")

New names:

\* `` -> `...1`

> mydata

# A tibble: 6 x 5

...1 mpg cyl disp hp

<chr> <dbl> <dbl> <dbl> <dbl>

1 Mazda RX4 21 6 160 110

2 Mazda RX4 Wag 21 6 160 110

3 Datsun 710 22.8 4 108 93

4 Hornet 4 Drive 21.4 6 258 110

5 Hornet Sportabout 18.7 8 360 175

6 Valiant 18.1 6 225 105

>

> id<-c(100,101,102)

> sname<-c("Mango","Apple","Berry")

> cost<-c(200,100,300)

> fruits<-data.frame(id,sname,cost)

> fruits

id sname cost

1 100 Mango 200

2 101 Apple 100

3 102 Berry 300

> write.csv(fruits,file = "Fruits")

**AIM: Write R script to find data distribution using box plot and scatter plot on IRIS dataset, find outlier using plot on sample dataset, plot histogram and bar chart on MTCARS dataset, plot pie chart on IRIS dataset**

**DESCRIPTION:**

Box plot show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.

A scatter plot shows the relationship between two quantitative variables measured for same individuals. The values of one variable appear on the horizontal axis whereas the other on vertical axis. Each individual appears as a point on the graph.

A histogram is a graphical representation of data points organized and grouping them into logical ranges.

A bar chart plots numeric values for levels of a categorical feature as bars. Levels are plotted on one chart axis, and values are plotted on the other axis. Each categorical value claims one bar, and the length of each bar corresponds to the bar’s value.

A pie chart is a way of summarizing a set of nominal data or displaying the different values of a given variable. This type of chart is a circle divided into a series of segments. Each segment represents a particular category.

**PROGRAM:**

data(iris)

boxplot(iris$Sepal.Length)

boxplot(iris$Sepal.Width~iris$Species)

plot(iris$Petal.Length)

plot(iris$Petal.Length,pch=20)

salary<-c(20,30,40,50,200)

mean(salary)

median(salary)

boxplot(salary)

data(mtcars)

hist(mtcars$mpg)

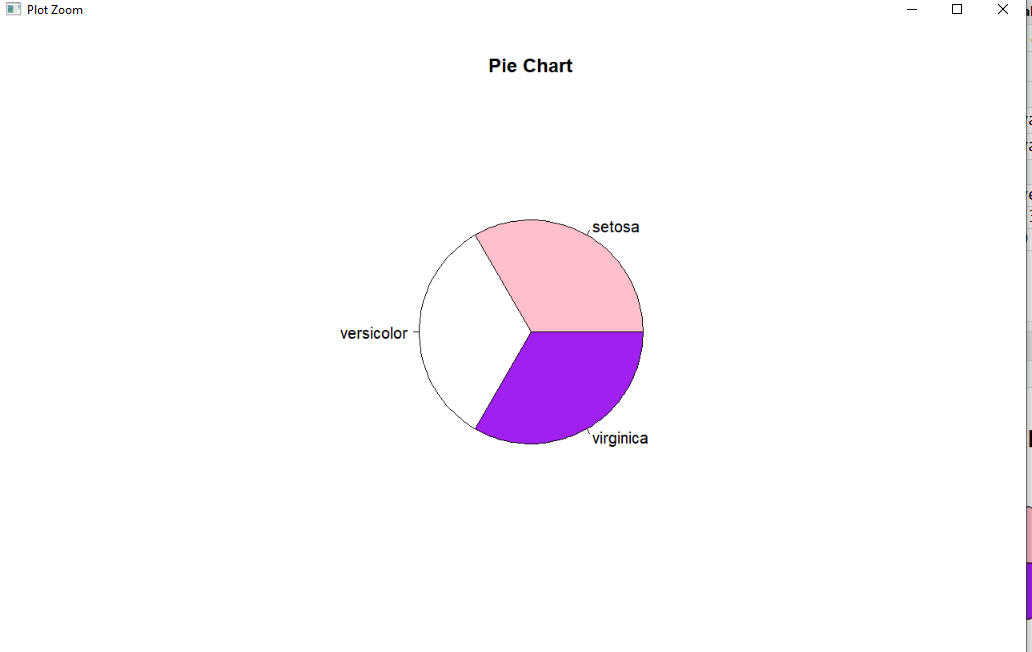
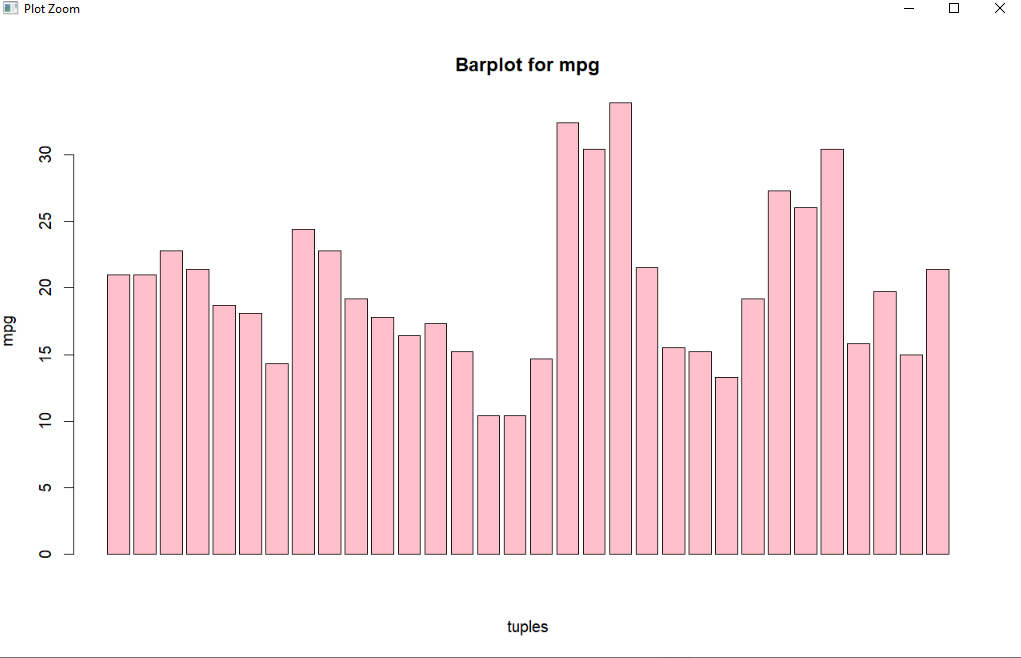
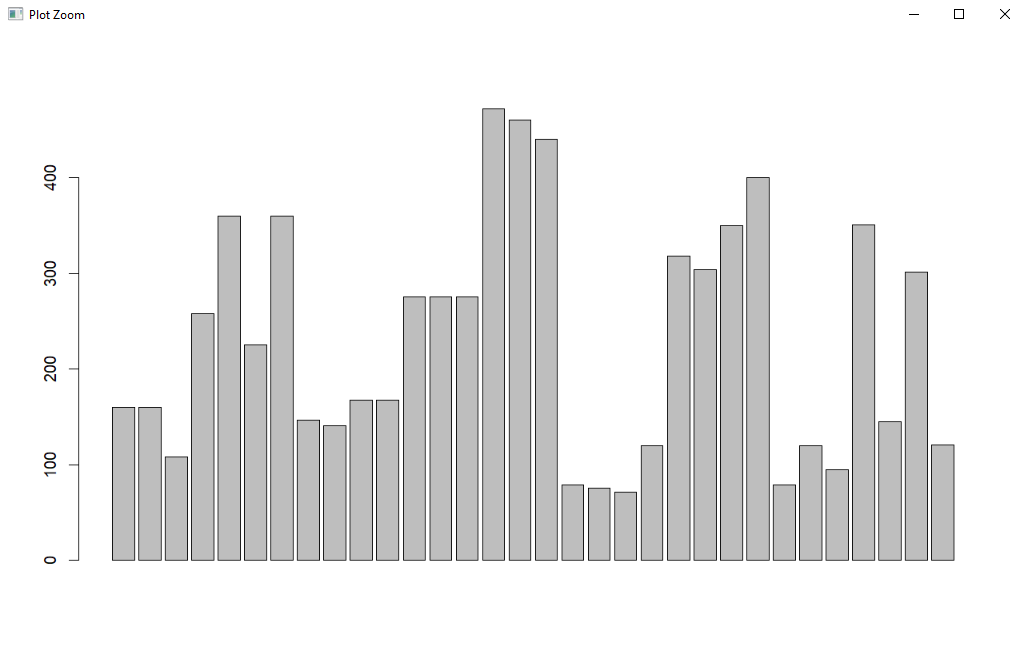
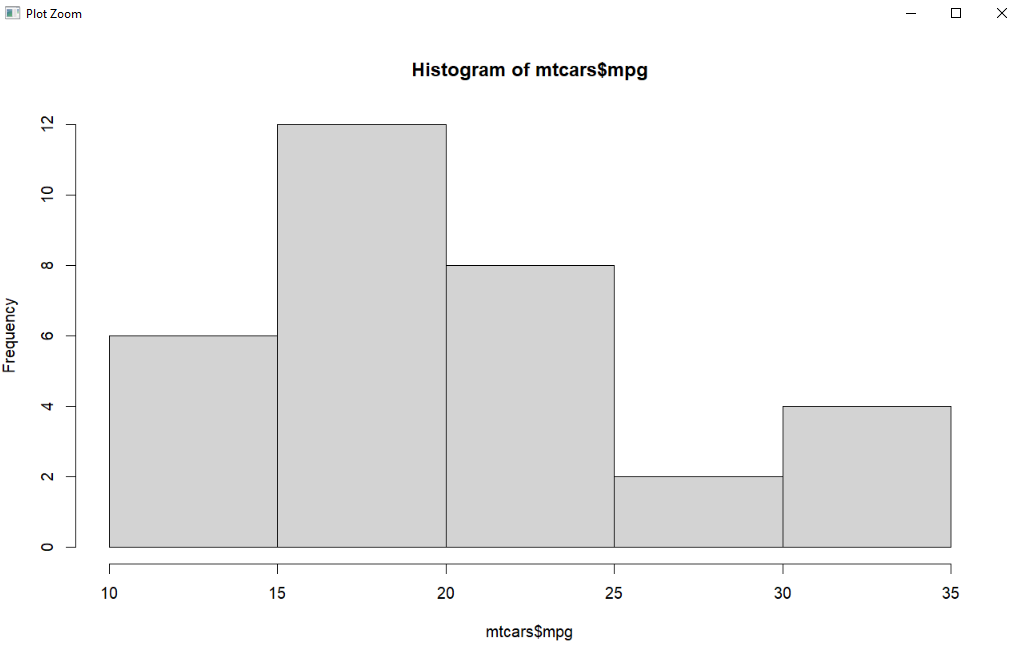
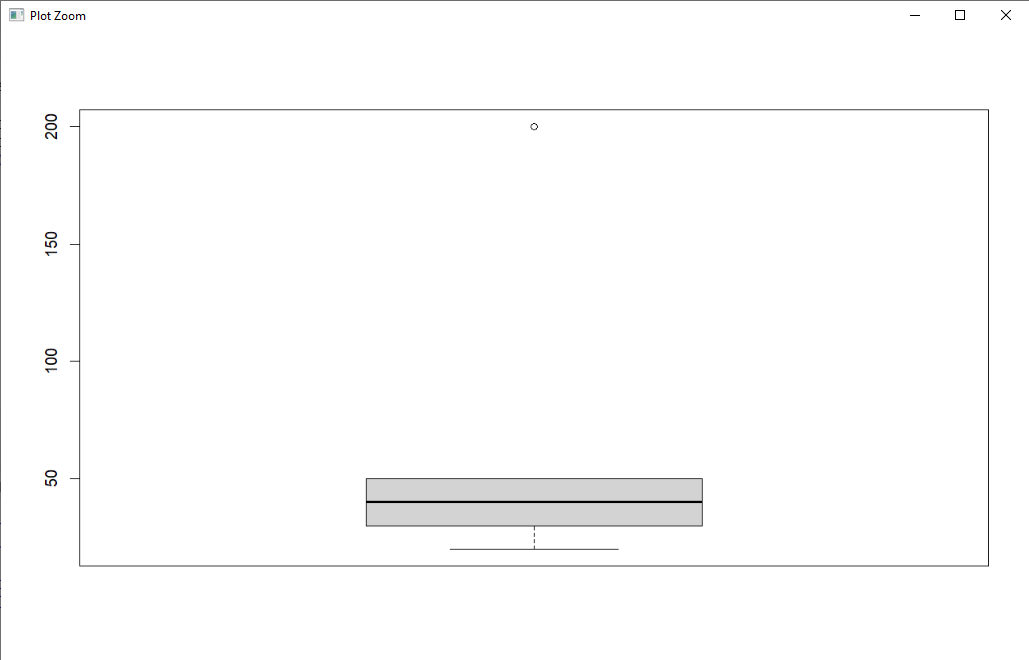
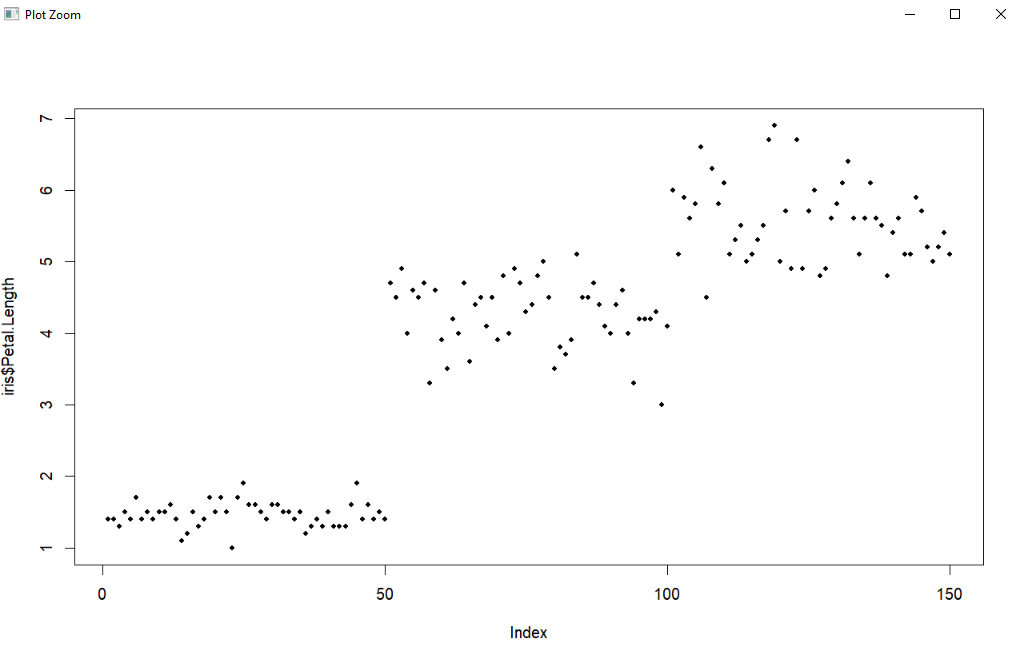
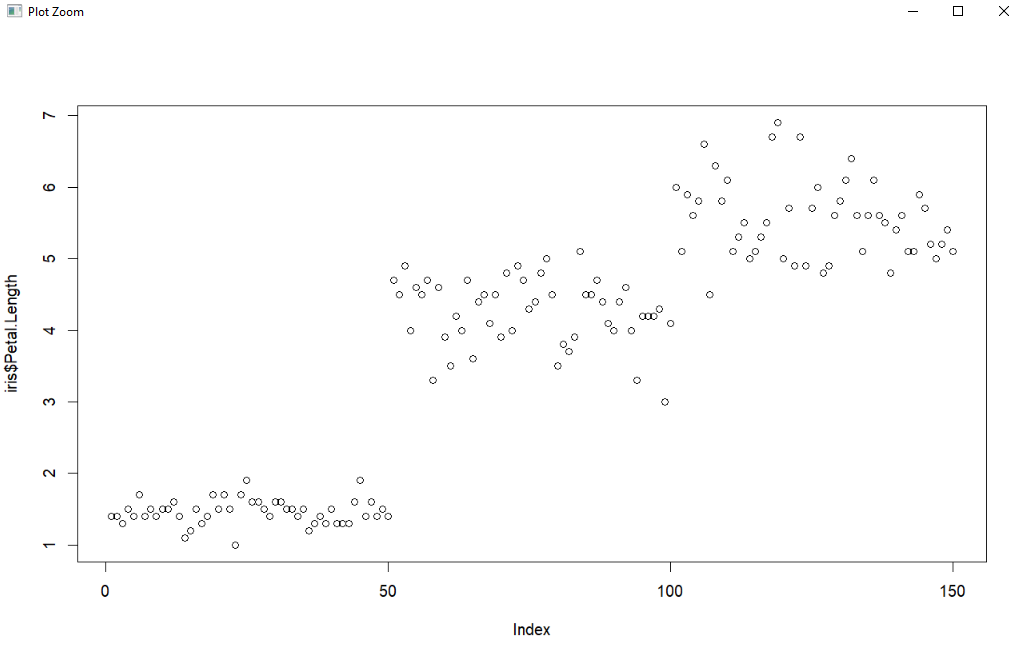
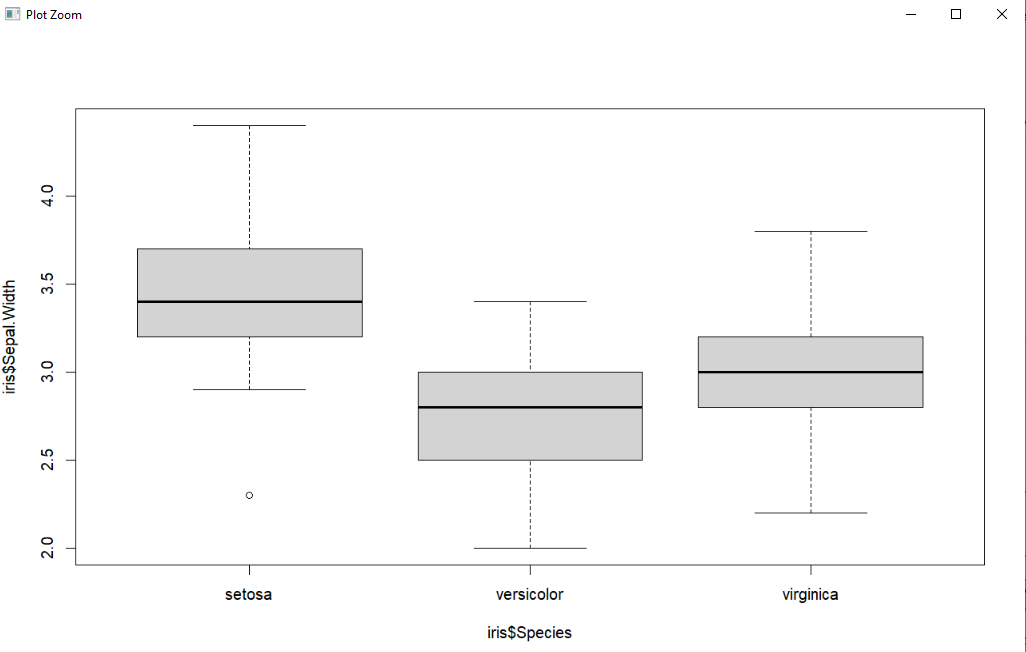
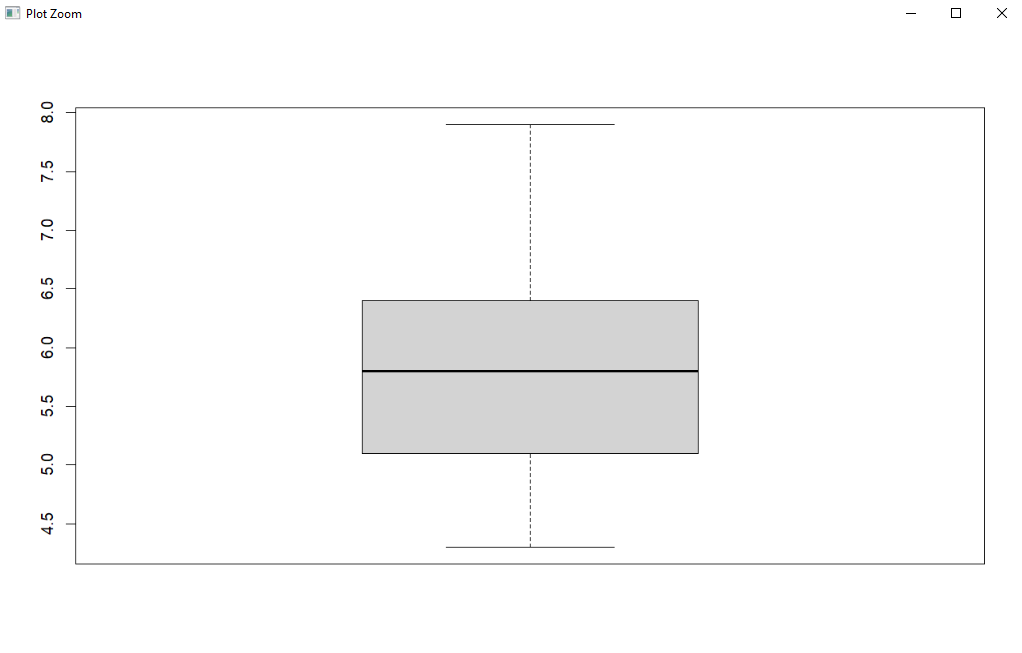
barplot(mtcars$disp)

barplot(mtcars$mpg,main="Barplot for mpg", xlab = "tuples",ylab="mpg",col="pink")

mydata<-factor(iris$Species)

mytable<-table(mydata)

pie(mytable,main=”Pie Chart”,col=c(“pink”,”white”,”purple”))



**OUTPUT:**

> data(iris)

> boxplot(iris$Sepal.Length)

> boxplot(iris$Sepal.Width~iris$Species)

> plot(iris$Petal.Length)

> plot(iris$Petal.Length,pch=20)

> salary<-c(20,30,40,50,200)

> mean(salary)

[1] 68

> median(salary)

[1] 40

> boxplot(salary)

> data(mtcars)

> hist(mtcars$mpg)

> barplot(mtcars$disp)

> barplot(mtcars$mpg,main="Barplot for mpg", xlab = "tuples",ylab="mpg",col="pink")

> mydata<-factor(iris$Species)

> mytable<-table(mydata)

> pie(mytable,main="Pie Chart",col=c("pink","white","purple"))

**AIM: Write R script to find correlation matrix, plot the correlation matrix, visualize the results and find the covariance on IRIS dataset.**

**DESCRIPTION:**

A correlation matrix is simply a table which displays the correlation coefficients for different

variables. The matrix depicts the correlation between all the possible pairs of values in a

table. Each cell in the table shows the correlation between two specific variables.

Covariance is a measure of the relationship between two random variables and to what

extent, they change together. Or we can say, in other words, it defines the changes between

the two variables, such that change in one variable is equal to change in another variable.

**PROGRAM:**

data(iris)

iris

mydata<-iris[,c(1,2,3,4)]

mydata

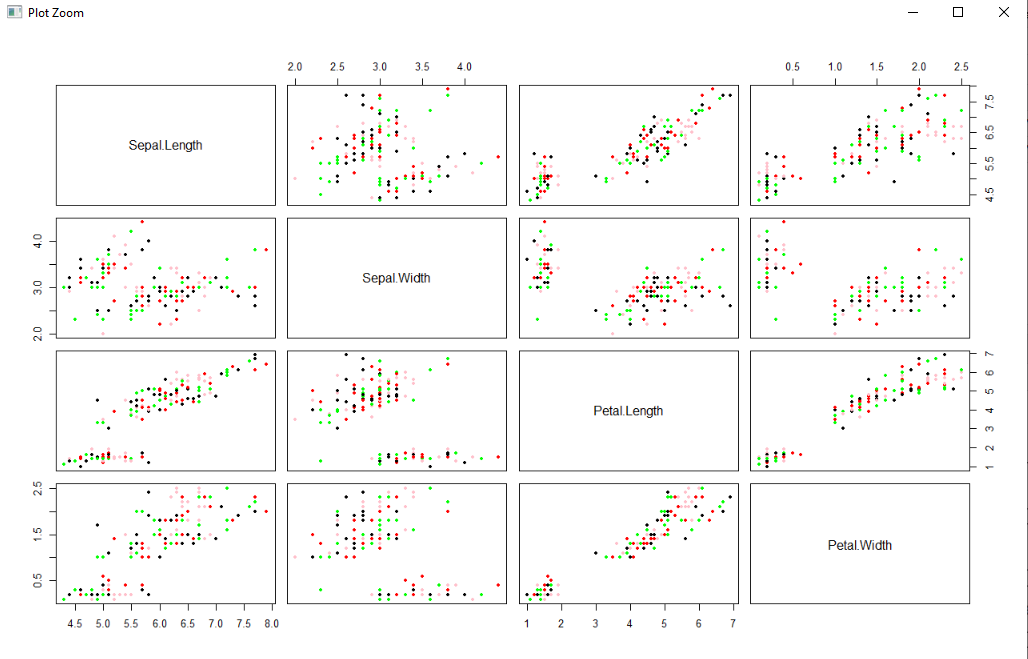
cor(mydata)

cor(mydata$Sepal.Length,mydata$Sepal.Width)

pairs(mydata,col=c("pink","green","black","red"),pch=20)

cov(mydata)

cov(mydata$Sepal.Length,mydata$Sepal.Width)

****

**OUTPUT:**

> data(iris)

> mydata<-iris[,c(1,2,3,4)]

> cor(mydata)

Sepal.Length Sepal.Width Petal.Length Petal.Width

Sepal.Length 1.0000000 -0.1175698 0.8717538 0.8179411

Sepal.Width -0.1175698 1.0000000 -0.4284401 -0.3661259

Petal.Length 0.8717538 -0.4284401 1.0000000 0.9628654

Petal.Width 0.8179411 -0.3661259 0.9628654 1.0000000

> cor(mydata$Sepal.Length,mydata$Sepal.Width)

[1] -0.1175698

> pairs(mydata,col=c("pink","green","black","red"),pch=20)

> cov(mydata)

Sepal.Length Sepal.Width Petal.Length Petal.Width

Sepal.Length 0.6856935 -0.0424340 1.2743154 0.5162707

Sepal.Width -0.0424340 0.1899794 -0.3296564 -0.1216394

Petal.Length 1.2743154 -0.3296564 3.1162779 1.2956094

Petal.Width 0.5162707 -0.1216394 1.2956094 0.5810063

> cov(mydata$Sepal.Length,mydata$Sepal.Width)

[1] -0.042434

**AIM: Write R script to perform logistic regression to find relationship between variables and check the model for fit using crashdata dataset**

**DESCRIPTION:**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

The sigmoid function is a mathematical function used to map the predicted values to probabilities.

It maps any real value into another value within a range of 0 and 1.

**PROGRAM:**

mydata<-read.csv("crashdata.csv")

mytestdata<-read.csv("crashtestdata.csv")

str(mydata)

summary(mydata)

mydata[6]<-as.factor(mydata$CarType)

str(mydata)

glm(formula = mydata$CarType~.,family="binomial",data=mydata)

fit<-glm(formula = mydata$CarType~.,family="binomial",data=mydata)

fit

summary(fit)

train<-predict(fit,type="response")

plot(train)

test<-predict(fit,type="response",newdata = mytestdata)

plot(test)

mytestdata

mytestdata[test<=0.5,"Predict"]<-"Hatchback"

mytestdata[test>=0.5,"Predict"]<-"SUV"

mytestdata

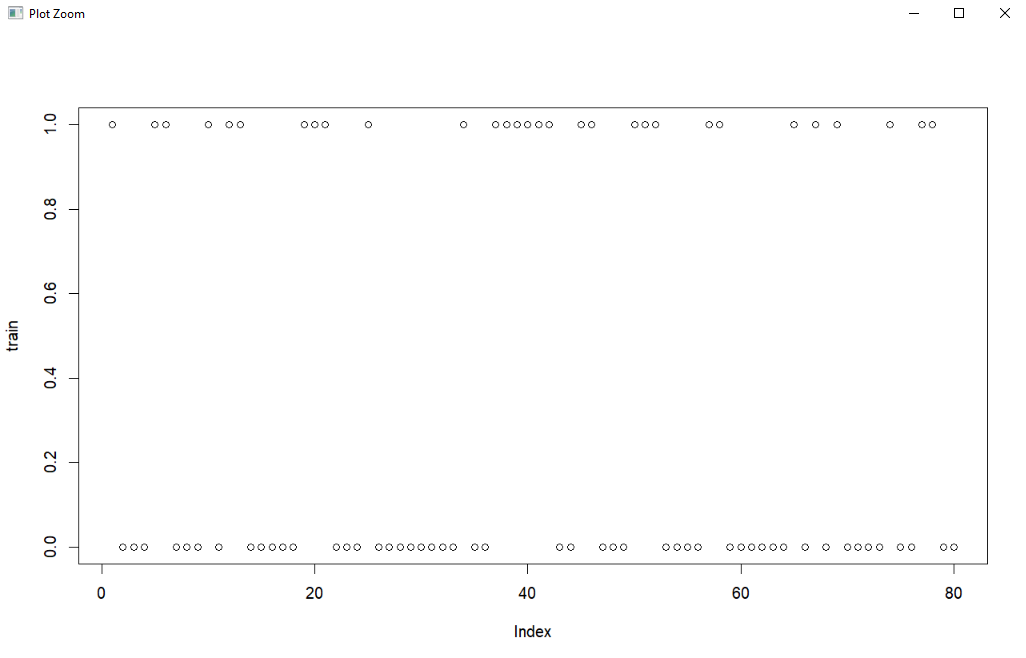
install.packages("caret")

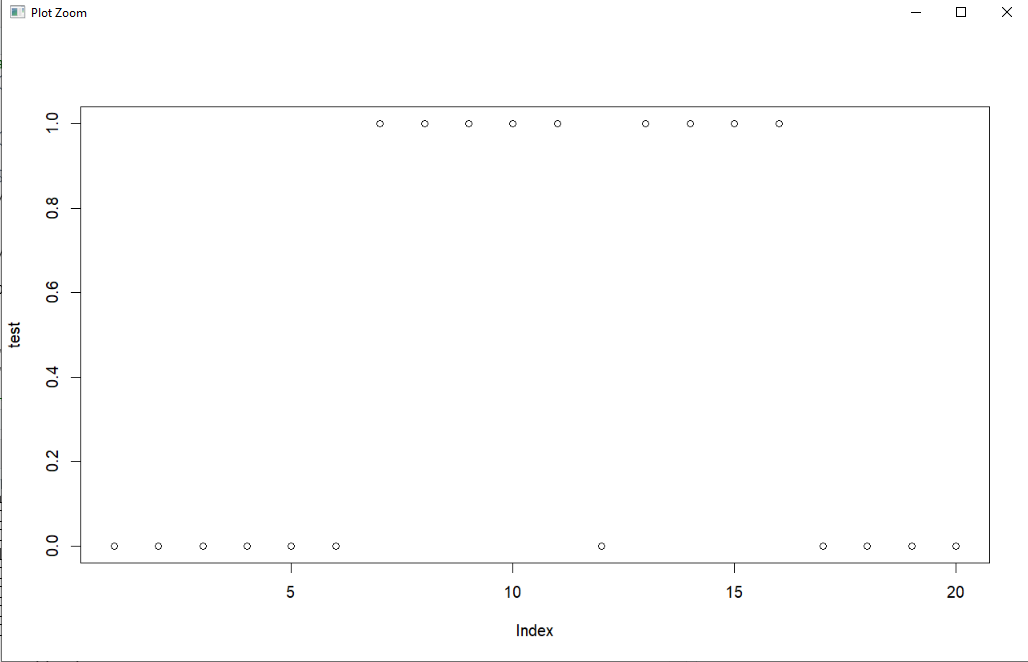
library(caret)

mytable<-table(mytestdata[,7],mytestdata[,6])

mytable

confusionMatrix(mytable,positive = "Hatchback")





**OUTPUT:**

> mydata<-read.csv("crashdata.csv")

> mytestdata<-read.csv("crashtestdata.csv")

> str(mydata)

'data.frame': 80 obs. of 6 variables:

$ ManHI : num -5.27 -4.82 9.57 2.84 0 0.4 5.94 5.78 0.86 7.36 ...

$ ManBI : num -1.3 -5.38 -7.5 -2.85 2.68 6.34 3.14 -1.75 -4.32 7.42 ...

$ IntI : num 2.86 9.72 -7.61 0.92 -4.15 0.83 -6.65 -6.85 8.1 0.27 ...

$ HVACi : num -4.85 -0.97 1.33 5.51 0.85 5.03 6.62 0.73 -8.96 -8.62 ...

$ Safety : num 4.04 -4.57 -5.1 -6.64 5.58 -8.1 -1.32 5.5 3.1 3.08 ...

$ CarType: chr "SUV" "Hatchback" "Hatchback" "Hatchback" ...

> summary(mydata)

ManHI ManBI IntI HVACi

Min. :-9.9300 Min. :-9.9400 Min. :-9.9900 Min. :-9.8200

1st Qu.:-5.1950 1st Qu.:-5.7050 1st Qu.:-5.5725 1st Qu.:-5.6750

Median : 0.6350 Median :-1.8150 Median :-0.4150 Median : 0.8700

Mean :-0.0935 Mean :-0.9277 Mean :-0.1349 Mean : 0.1197

3rd Qu.: 5.0500 3rd Qu.: 3.4175 3rd Qu.: 4.9775 3rd Qu.: 5.0625

Max. : 9.5700 Max. : 9.6100 Max. : 9.7200 Max. : 9.8900

Safety CarType

Min. :-9.8000 Length:80

1st Qu.:-4.6775 Class :character

Median : 0.8300 Mode :character

Mean : 0.5437

3rd Qu.: 4.6225

Max. : 9.9900

> mydata[6]<-as.factor(mydata$CarType)

> str(mydata)

'data.frame': 80 obs. of 6 variables:

$ ManHI : num -5.27 -4.82 9.57 2.84 0 0.4 5.94 5.78 0.86 7.36 ...

$ ManBI : num -1.3 -5.38 -7.5 -2.85 2.68 6.34 3.14 -1.75 -4.32 7.42 ...

$ IntI : num 2.86 9.72 -7.61 0.92 -4.15 0.83 -6.65 -6.85 8.1 0.27 ...

$ HVACi : num -4.85 -0.97 1.33 5.51 0.85 5.03 6.62 0.73 -8.96 -8.62 ...

$ Safety : num 4.04 -4.57 -5.1 -6.64 5.58 -8.1 -1.32 5.5 3.1 3.08 ...

$ CarType: Factor w/ 2 levels "Hatchback","SUV": 2 1 1 1 2 2 1 1 1 2 ...

> glm(formula = mydata$CarType~.,family="binomial",data=mydata)

Call: glm(formula = mydata$CarType ~ ., family = "binomial", data = mydata)

Coefficients:

(Intercept) ManHI ManBI IntI HVACi Safety

-22.76 -13.48 36.02 -44.90 -58.50 -27.36

Degrees of Freedom: 79 Total (i.e. Null); 74 Residual

Null Deviance: 105.9

Residual Deviance: 5.359e-08 AIC: 12

> fit

Call: glm(formula = mydata$CarType ~ ., family = "binomial", data = mydata)

Coefficients:

(Intercept) ManHI ManBI IntI HVACi Safety

-22.76 -13.48 36.02 -44.90 -58.50 -27.36

Degrees of Freedom: 79 Total (i.e. Null); 74 Residual

Null Deviance: 105.9

Residual Deviance: 5.359e-08 AIC: 12

> summary(fit)

Call:

glm(formula = mydata$CarType ~ ., family = "binomial", data = mydata)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.316e-04 -2.100e-08 -2.100e-08 2.100e-08 1.266e-04

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -22.76 12007.54 -0.002 0.998

ManHI -13.48 3077.29 -0.004 0.997

ManBI 36.02 7221.18 0.005 0.996

IntI -44.90 8853.08 -0.005 0.996

HVACi -58.50 11461.92 -0.005 0.996

Safety -27.36 5396.42 -0.005 0.996

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1.0585e+02 on 79 degrees of freedom

Residual deviance: 5.3590e-08 on 74 degrees of freedom

AIC: 12

Number of Fisher Scoring iterations: 25

> train<-predict(fit,type="response")

> plot(train)

> test<-predict(fit,type="response",newdata = mytestdata)

> plot(test)

> mytestdata

ManHI ManBI IntI HVACi Safety CarType

1 1.94 2.21 3.38 1.78 -7.19 Hatchback

2 -0.02 -3.33 0.79 -6.63 7.99 SUV

3 -0.49 -4.48 5.00 8.33 -2.77 Hatchback

4 5.76 1.35 7.92 -0.43 4.29 Hatchback

5 2.51 -8.74 4.53 -1.91 3.95 Hatchback

6 -4.47 8.42 -0.05 5.57 9.62 Hatchback

7 -9.89 -2.25 -5.00 -9.23 9.38 SUV

8 -9.94 -3.23 2.81 -2.98 -1.12 SUV

9 -8.37 4.21 -8.95 6.66 7.34 SUV

10 8.48 0.38 -3.02 -1.92 -7.43 SUV

11 0.79 0.96 -4.03 -2.28 6.20 SUV

12 5.32 2.08 5.55 7.89 -6.80 Hatchback

13 -7.26 -0.11 -5.27 -7.14 1.20 SUV

14 0.69 3.37 3.70 -5.73 -5.86 SUV

15 -5.53 -0.12 1.61 2.31 -8.66 SUV

16 8.29 1.44 -7.26 5.06 -7.00 SUV

17 9.09 -2.26 1.64 2.80 -1.22 Hatchback

18 5.04 4.52 0.28 8.26 4.59 Hatchback

19 4.55 -3.88 -2.02 -1.20 -0.42 Hatchback

20 -5.55 6.02 8.87 5.26 -2.27 Hatchback

> mytestdata[test<=0.5,"Predict"]<-"Hatchback"

> mytestdata[test>=0.5,"Predict"]<-"SUV"

> mytestdata

ManHI ManBI IntI HVACi Safety CarType Predict

1 1.94 2.21 3.38 1.78 -7.19 Hatchback Hatchback

2 -0.02 -3.33 0.79 -6.63 7.99 SUV Hatchback

3 -0.49 -4.48 5.00 8.33 -2.77 Hatchback Hatchback

4 5.76 1.35 7.92 -0.43 4.29 Hatchback Hatchback

5 2.51 -8.74 4.53 -1.91 3.95 Hatchback Hatchback

6 -4.47 8.42 -0.05 5.57 9.62 Hatchback Hatchback

7 -9.89 -2.25 -5.00 -9.23 9.38 SUV SUV

8 -9.94 -3.23 2.81 -2.98 -1.12 SUV SUV

9 -8.37 4.21 -8.95 6.66 7.34 SUV SUV

10 8.48 0.38 -3.02 -1.92 -7.43 SUV SUV

11 0.79 0.96 -4.03 -2.28 6.20 SUV SUV

12 5.32 2.08 5.55 7.89 -6.80 Hatchback Hatchback

13 -7.26 -0.11 -5.27 -7.14 1.20 SUV SUV

14 0.69 3.37 3.70 -5.73 -5.86 SUV SUV

15 -5.53 -0.12 1.61 2.31 -8.66 SUV SUV

16 8.29 1.44 -7.26 5.06 -7.00 SUV SUV

17 9.09 -2.26 1.64 2.80 -1.22 Hatchback Hatchback

18 5.04 4.52 0.28 8.26 4.59 Hatchback Hatchback

19 4.55 -3.88 -2.02 -1.20 -0.42 Hatchback Hatchback

20 -5.55 6.02 8.87 5.26 -2.27 Hatchback Hatchback

> install.packages("caret")

Error in install.packages : Updating loaded packages

> install.packages("caret")

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/

Warning in install.packages :

package ‘caret’ is in use and will not be installed

> library(caret)

> mytable<-table(mytestdata[,7],mytestdata[,6])

> mytable

Hatchback SUV

Hatchback 10 1

SUV 0 9

> confusionMatrix(mytable,positive = "Hatchback")

Confusion Matrix and Statistics

Hatchback SUV

Hatchback 10 1

SUV 0 9

Accuracy : 0.95

95% CI : (0.7513, 0.9987)

No Information Rate : 0.5

P-Value [Acc > NIR] : 2.003e-05

Kappa : 0.9

Mcnemar's Test P-Value : 1

Sensitivity : 1.0000

Specificity : 0.9000

Pos Pred Value : 0.9091

Neg Pred Value : 1.0000

Prevalence : 0.5000

Detection Rate : 0.5000

Detection Prevalence : 0.5500

Balanced Accuracy : 0.9500

'Positive' Class : Hatchback

**AIM: Write R script to build KNN classification model using servicetraindata dataset and evaluate the performance**

**DESCRIPTION:**

K-Nearest Neighbour or K-NN is a Supervised Non-linear classification algorithm. In the KNN algorithm, K specifies the number of neighbours and its algorithm is as follows:

Choose the number K of neighbor.

Take the K Nearest Neighbor of unknown data point according to distance.

Among the K-neighbors, Count the number of data points in each category.

Assign the new data point to a category, where you counted the most neighbors.

For the Nearest Neighbor classifier, the distance between two points is expressed in the form of Euclidean Distance.

**PROGRAM:**

mytraindata<-read.csv("servicetraindata.csv")

mytestdata<-read.csv("servicetestdata.csv")

str(mytraindata)

str(mytestdata)

head(mytraindata)

mytraindata[6]<-as.factor(mytraindata$Service)

mytraindata[6]

mytestdata[6]<-as.factor(mytestdata$Service)

mytestdata[6]

str(mytraindata)

install.packages("class")

library(class)

nn<-knn(train=mytraindata[,1:5],test=mytestdata[,-6],cl=mytraindata$Service,k=3)

nn

install.packages("caret")

library(caret)

confusionMatrix(data = nn,mytestdata$Service)

**OUTPUT:**

> mytraindata<-read.csv("servicetraindata.csv")

> mytestdata<-read.csv("servicetestdata.csv")

> str(mytraindata)

'data.frame': 315 obs. of 6 variables:

$ OilQual : num 103.4 26.8 62.4 45.5 104.4 ...

$ EnginePerf : num 103.5 26.2 63.7 49.9 103.3 ...

$ NormMileage: num 103.1 31.3 59.7 48.8 103.1 ...

$ TyreWear : num 106.2 29.2 64.7 48.1 105.8 ...

$ HVACwear : num 105.7 31.3 58.6 48 106.5 ...

$ Service : chr "No" "Yes" "Yes" "No" ...

> mytraindata[6]<-as.factor(mytraindata$Service)

> mytestdata[6]<-as.factor(mytestdata$Service)

> str(mytraindata)

'data.frame': 315 obs. of 6 variables:

$ OilQual : num 103.4 26.8 62.4 45.5 104.4 ...

$ EnginePerf : num 103.5 26.2 63.7 49.9 103.3 ...

$ NormMileage: num 103.1 31.3 59.7 48.8 103.1 ...

$ TyreWear : num 106.2 29.2 64.7 48.1 105.8 ...

$ HVACwear : num 105.7 31.3 58.6 48 106.5 ...

$ Service : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 1 1 ...

> install.packages("class")

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/

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/class\_7.3-20.zip'

Content type 'application/zip' length 107990 bytes (105 KB)

downloaded 105 KB

package ‘class’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Student\AppData\Local\Temp\Rtmpqecxdz\downloaded\_packages

> library(class)

Warning message:

package ‘class’ was built under R version 4.1.3

> nn<-knn(train=mytraindata[,1:5],test=mytestdata[,-6],cl=mytraindata$Service,k=3)

> nn

[1] No No No No No No No Yes Yes No No No No No No No Yes No No Yes Yes No Yes No No No No

[28] No No No No No No No No Yes No No No No No Yes No Yes No No No Yes Yes No Yes No Yes No

[55] No No No No No No No No No No Yes Yes Yes No Yes No No Yes No No No No No No Yes Yes Yes

[82] Yes No Yes No No Yes Yes Yes No No No Yes No Yes No No No No No No No No No No Yes No No

[109] No No No Yes No Yes No Yes Yes No Yes No No No No No Yes No No No No No No No Yes No No

Levels: No Yes

> install.packages("caret")

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/

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/caret\_6.0-93.zip'

Content type 'application/zip' length 3588908 bytes (3.4 MB)

downloaded 3.4 MB

package ‘caret’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Student\AppData\Local\Temp\Rtmpqecxdz\downloaded\_packages

> library(caret)

Loading required package: ggplot2

Loading required package: lattice

Warning messages:

1: package ‘caret’ was built under R version 4.1.3

2: package ‘ggplot2’ was built under R version 4.1.3

> confusionMatrix(data = nn,mytestdata$Service)

Confusion Matrix and Statistics

Reference

Prediction No Yes

No 99 0

Yes 0 36

Accuracy : 1

95% CI : (0.973, 1)

No Information Rate : 0.7333

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Sensitivity : 1.0000

Specificity : 1.0000

Pos Pred Value : 1.0000

Neg Pred Value : 1.0000

Prevalence : 0.7333

Detection Rate : 0.7333

Detection Prevalence : 0.7333

Balanced Accuracy : 1.0000

'Positive' Class : No

**AIM: Write R script to perform K means clustering using tripdetails dataset and plot the clustering data**

**DESCRIPTION:**

[K Means Clustering](https://www.geeksforgeeks.org/k-means-clustering-introduction/) in [R Programming](https://www.geeksforgeeks.org/introduction-to-r-programming-language/) is an Unsupervised Non-linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster. In the unsupervised algorithm, high reliance on raw data is given with large expenditure on manual review for review of relevance is given. It is used in a variety of fields like Banking, healthcare, retail, Media, etc.

K-Means clustering groups the data on similar groups. The algorithm is as follows:

Choose the number K clusters.

Select at random K points, the centroids (Not necessarily from the given data).

Assign each data point to closest centroid that forms K clusters.

Compute and place the new centroid of each centroid.

Reassign each data point to new cluster.

After final reassignment, name the cluster as Final cluster.

**PROGRAM:**

mydata<-read.csv("tripdetails.csv")

mydata

str(mydata)

summary(mydata)

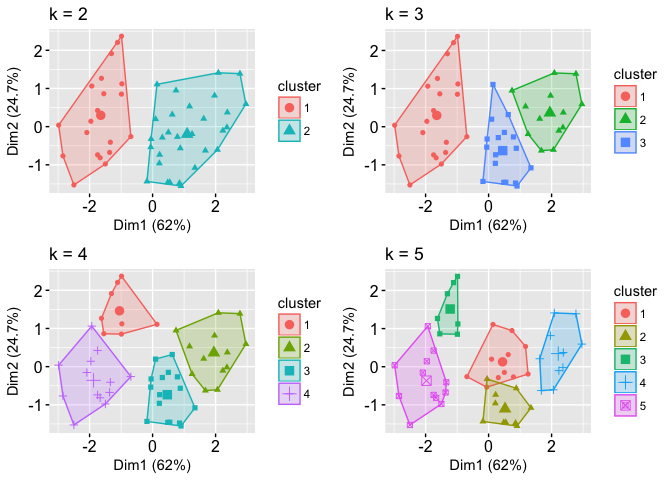
myclusters<-kmeans(mydata[-1],5)

myclusters

install.packages("factoextra")

library(factoextra)

fviz\_cluster(myclusters,data = mydata,geom = "point")



**OUTPUT:**

> mydata<-read.csv("tripdetails.csv")

> str(mydata)

'data.frame': 91 obs. of 8 variables:

$ TripID : int 1 2 3 4 5 6 7 8 9 10 ...

$ TripLength : int 21 148 18 22 183 18 20 21 181 174 ...

$ MaxSpeed : int 51 130 38 43 108 43 37 38 99 100 ...

$ MostFreqSpeed: int 14 106 16 48 90 13 15 14 108 92 ...

$ TripDuration : int 93 156 100 36 171 64 85 69 155 133 ...

$ Brakes : int 307 226 351 17 88 136 121 114 86 106 ...

$ IdlingTime : int 27 5 26 4 5 25 26 25 5 5 ...

$ Honking : int 112 114 107 5 29 21 23 20 25 34 ...

> summary(mydata)

TripID TripLength MaxSpeed MostFreqSpeed TripDuration Brakes

Min. : 1.0 Min. : 16.00 Min. : 35.00 Min. : 12.00 Min. : 22.00 Min. : 14.0

1st Qu.:23.5 1st Qu.: 20.00 1st Qu.: 42.00 1st Qu.: 15.50 1st Qu.: 34.50 1st Qu.: 36.5

Median :46.0 Median : 21.00 Median : 54.00 Median : 42.00 Median : 88.00 Median :100.0

Mean :46.0 Mean : 70.77 Mean : 70.36 Mean : 50.65 Mean : 87.37 Mean :135.4

3rd Qu.:68.5 3rd Qu.:163.00 3rd Qu.:105.50 3rd Qu.: 89.00 3rd Qu.:133.00 3rd Qu.:198.0

Max. :91.0 Max. :210.00 Max. :138.00 Max. :118.00 Max. :171.00 Max. :429.0

IdlingTime Honking

Min. : 4.00 Min. : 4.00

1st Qu.: 5.00 1st Qu.: 20.00

Median : 5.00 Median : 25.00

Mean :11.59 Mean : 49.92

3rd Qu.:24.00 3rd Qu.: 97.50

Max. :32.00 Max. :155.00

> myclusters<-kmeans(mydata[-1],5)

> install.packages("factoextra")

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/

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/factoextra\_1.0.7.zip'

Content type 'application/zip' length 416648 bytes (406 KB)

downloaded 406 KB

package ‘factoextra’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Student\AppData\Local\Temp\RtmpOUtYGQ\downloaded\_packages

> library(factoextra)

Loading required package: ggplot2

Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

Warning messages:

1: package ‘factoextra’ was built under R version 4.1.3

2: package ‘ggplot2’ was built under R version 4.1.3

> fviz\_cluster(myclusters,data = mydata,geom = "point")