Data Science With *R* Language

Prag Robotics Private Limited

Table of Contents

Table of Figures

1 INTRODUCTION TO R LANGUAGE

1.1 R language

R language is an environment for statistical modelling and a successor to S language. It was developed by Bell Laboratories. R provides a variety of statistical and graphical techniques like Linear modelling, non-linear modelling, statistical tests, Hypothesis analysis, time series analysis, classification[1] and clustering. S language is often the choice for research in the field of statistics, but R provides an open-source channel. Data analysis in R is done in a series of steps, which includes

- Programming
- Transformation
- Discovery
- Modelling
- Communication

Programming: R is used as a programming tool to describe the data with programming terms. It can follow all the classical programming steps.

Transformation: Out of all the data gathered from various resources and databases, R has the ability and functions to perform data cleaning operations to generate a dataset which would reduce the error.

Discovery: In data analysis[2], after processing the data, the algorithms try to find patterns among the datasets. This step is called discovery.

Communicate: In communication, the algorithm performs it's calculations and generates the results. These results in return must be interpreted to the world in the form of graphs and plots.

With its abilities to discover patterns in the data, R became the reliable tool for industries, analysts and scientists. With the advent of Artificial Intelligence, Machine learning and deep learning R has become an essential tool.

Companies using R language include:

- Facebook
- Google

[1] A subset of supervised learning in machine learning
[2] Data Analysis is process of cleaning, inspecting and modelling data to have an inference of it

- Novartis
- IBM
- Ford
- Mckinsey

1.2 Applications of R language

- R is used for Data analysis
- R is used as a fundamental tool for finance
- Many quantitative analysts use R as their programming tool.
- R helps in data importing, cleaning and data mining
- R provides the environment for statistical computing and design

1.3 Some real-world applications include

- Health care
- Financial services
- Time- series prediction
- Profit & Loss prediction in Business Intelligence
- Inventory control
- Travel
- Monitoring and Diagnostics etc

1.4 R studio

R studio is an Integrated Development Environment for R language. It is used to type the code, test it and debug the code

Components of RStudio:

- Source
- Console
- Environment
- Workspace history
- Files
- Plots
- Packages
- Help

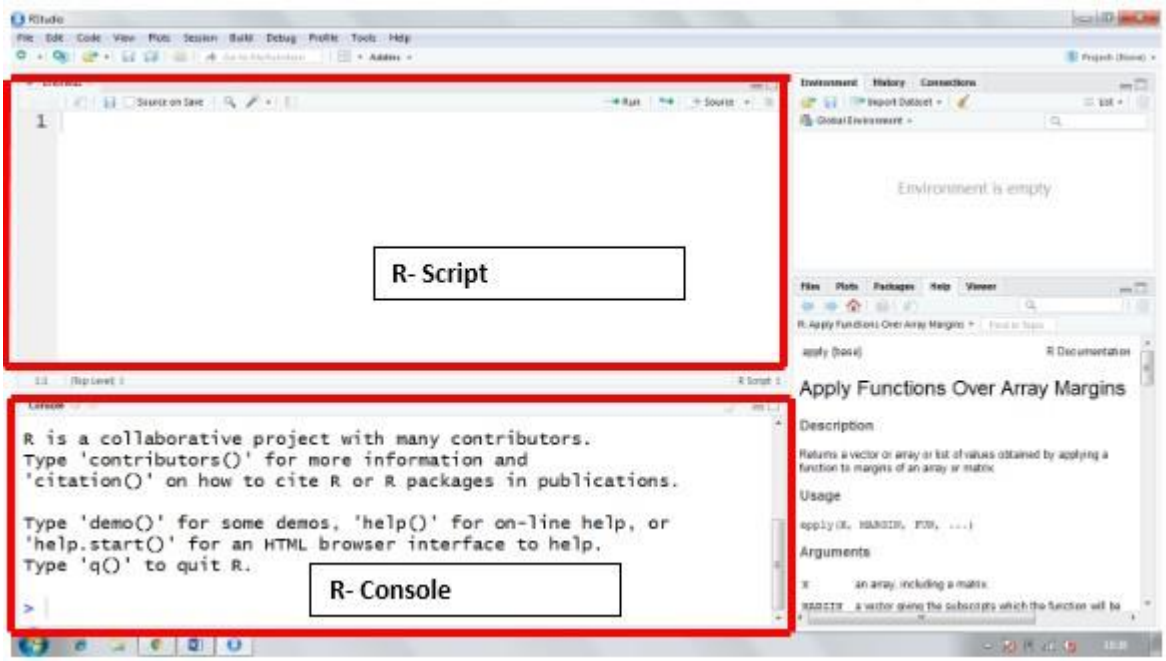

Figure 1.1 Elements of R studio-Script and console

Console and Scripts: Console is a sub window in R-Studio where the outputs of the executed commands are displayed. The commands entered in the console will be cleared as the console is cleared. Commands entered in the console cannot be stored. This will rise concerns if we are dealing with a big project. Another efficient way of writing the code is the R-scripts. The code can be written here and debugged to check for errors. Once the code is executed the output is displayed in the console.

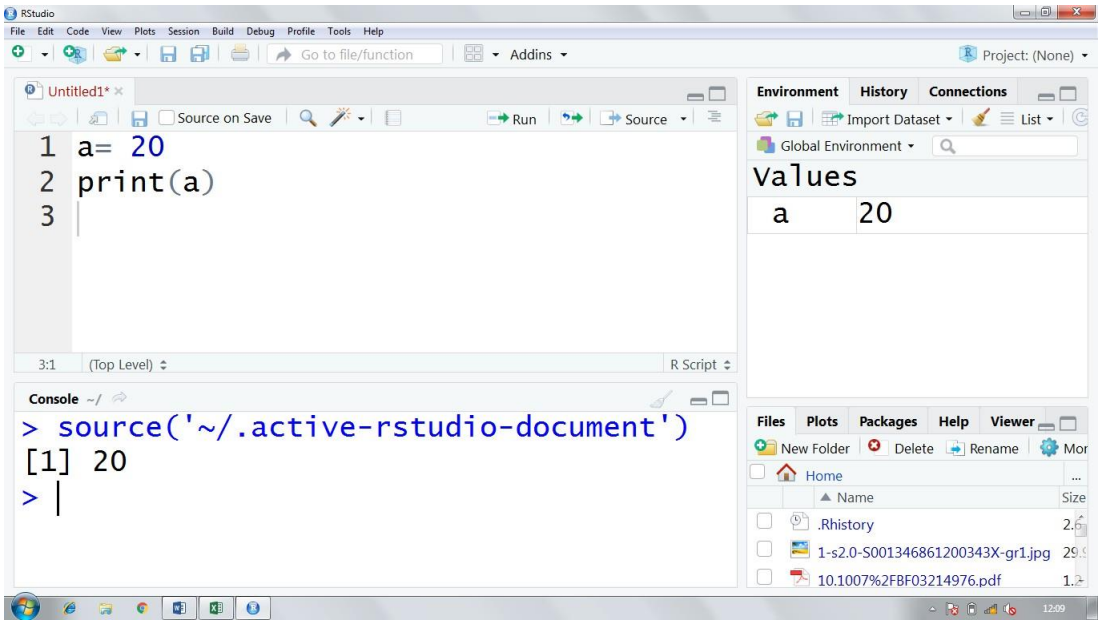

Figure 1.2 Elements of R Studio-Environment

Environment: In environment, there will be a repository of all the variables and their assigned values such as numbers, characters and data. A variable assigned or not can be verified by checking the environment which is on the top left quadrant of the R-studio window.

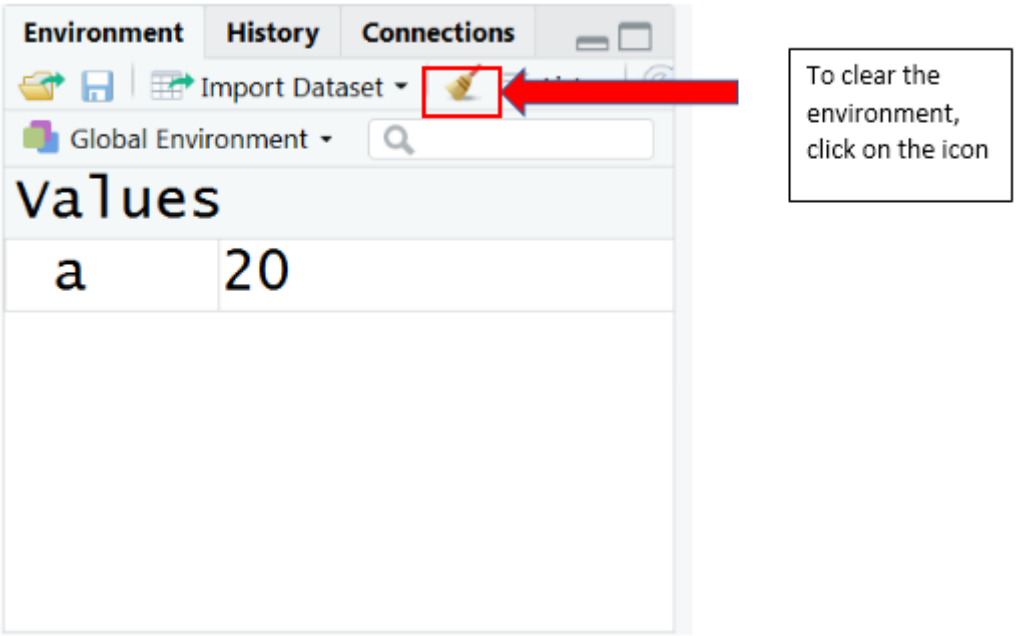

Figure 1.3 Elements of R Studio-Cleaning Environment

Note: The environment must be cleared before starting a new script or else the program will assume previously assigned values.

History: The history tab consists of all the executed commands both in the script and console. This will allow us to backtrack the executed commands.

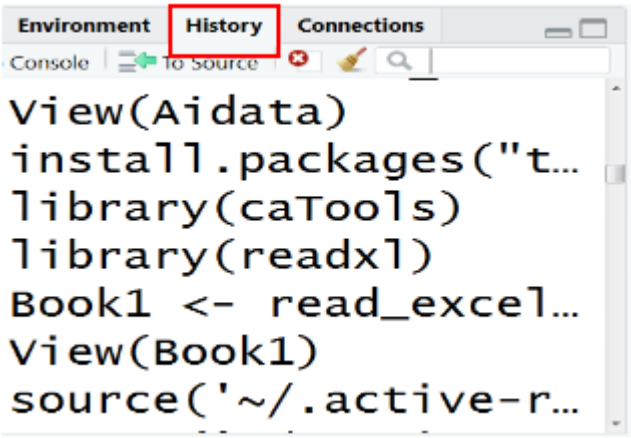

Figure 1.4 Elements of R Studio-History

Files: This is a file manager where all document folders are visible. This pane is used to relocate, download, rename various files used in the program.

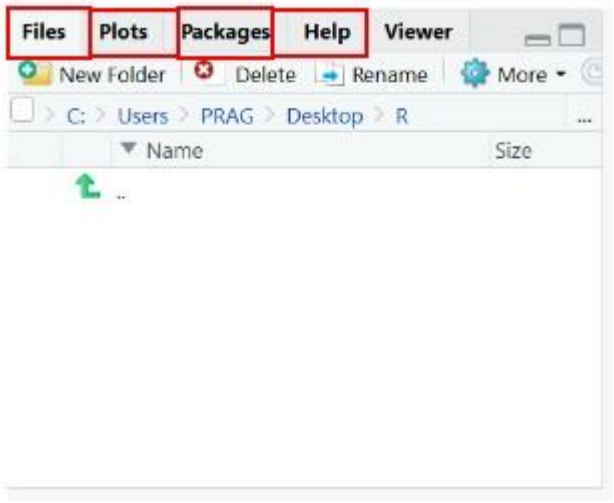

Figure 1.5 Elements of R Studio

Plots: All the graphical representation of the data, data visualization and diagrams associated with the program are displayed.

Figure 1.6 Elements of R Studio-Plots

Packages: Packages pane is a repository of all the additional packages present in RStudio. Each package or library is imported when necessary. Any packages which are not present in the repository can be installed by clicking on the install option.

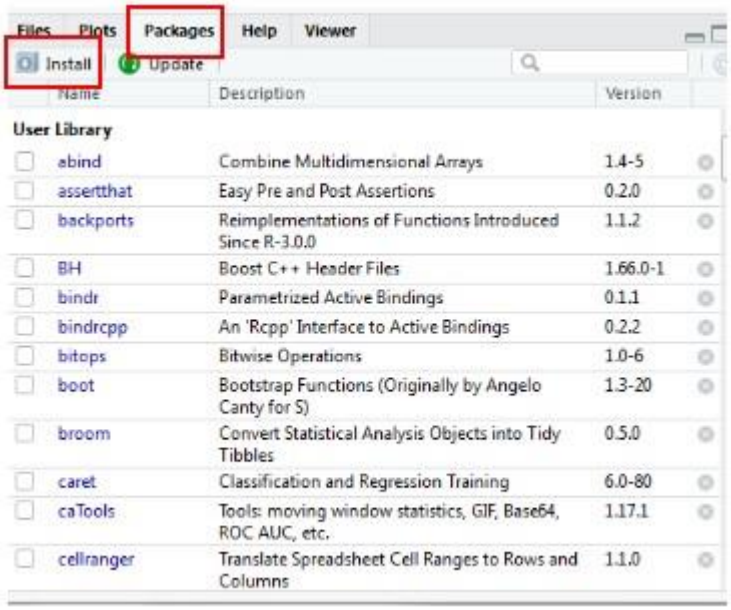

Figure 1.7 Elements of R Studio-Packages

Some of the packages required for machine learning are:

- ggplot
- caTools
- caret
- tree
- kernlab
- mboost
- Rweka
- CORELearn
- nnet
- randomforest
- rpart
- gbm

Help: The most important pane is the Help pane, all the documentation, manuals and sample program are available here.

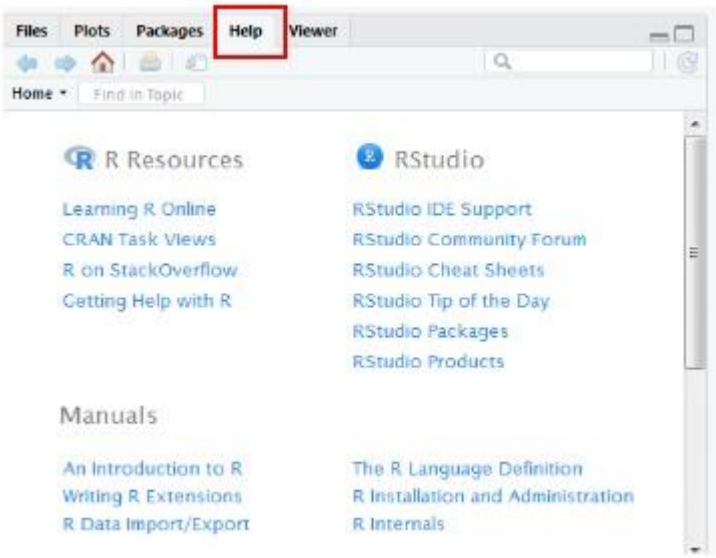

Figure 1.8 Elements of R Studio- Help

1.5 Shortcut keys

Key shortcut	**Function**
CTRL+L	CLEARS CONSOLE
CTRL+SHIFT+S	EXECUTES THE CODE
CTRL+SHIFT+N	OPEN NEW SCRIPT
CTRL+SHIFT+H	CHANGE WORKING DIRECTORY
CTRL+5	SHOW FILES
CTRL+7	SHOW PACKAGES
CTRL+5	SHOW HELP
CTRL+5	SHOW ENVIRONMENT
CTRL+6	SHOW PLOTS

1.6 Exercise

1. _____ is an Interactive Development Environment for R Language.
2. _____ is the pane used to visualise the graphs.
3. _____ is the shortcut to clear the console.
4. R language is a_____programming language.
5. _____ is the keyboard shortcut to view the environment.

2 Basics of R language

2.1 Variables, Datatypes and Assignments

A variable store a value, string vectors, array, various datatypes and objects. This variable will have the same value until it is changed in the program. Firstly, a variable is initialized to a value and is updated as the program progresses. A variable can be an alphabet or a word. There are three ways of assigning a variable in R language.

I. Use '=' symbol after the variable

A=15

Note: The above statement assigns a value 15 to the variable A

II. Use towards left or <- symbol after the variable

b<-" Statistics"

Note: The above statement assigns a string Statistics to the variable a

I. Use towards left or -> symbol after the variable

c-> 15000

Note: The above statement assigns a value 15 to the variable c

I. Use towards left or -> symbol after the variable

c-> 15000

Note: The above statement assigns a value 15000 to the variable c

I. Use an inbuilt function assign (“variable1”, value)

D=assign(“x”,35)

Note: The above statement assigns a value 35 to the variable x

 # Assign a variable and print

> Practice= “I am using RStudio”

>print (Practice)

When executed the program gives the following result in the console

[1] “I am using RStudio”

Note: To execute the syntax in the script select all the syntaxes and click ctrl+shift+s

2.1.1 Datatypes

In R language the datatype of the assigned variable changes dynamically. A variable which belongs to a numeric datatype can belong to character datatype in a different instance. Let us see different data types and structures in R language.

Variable	Datatype
A=15	Integer
A=” Hi”	Character
A=True or False	Logical
A=15.5	Numeric

Note: R language is case sensitive. Same alphabet with different cases is different variable.

```
>a=35
> A=” Hello world”
print(a)
print(A)
```

```
[1] 35
[2] "Hello world"
```

R language has inbuilt functions to check what is the datatype of the variable. For example, class () and typeof () gives the datatype of the variable.

```
# Checking the data type
> A=25
>class(A)
```

Once we execute the code the output of the command will be the datatype,

```
[1] integer
```

```
# Checking the datatype
> B=" Hello world"
>typeof(B)
```

Once we execute the code the output of the command will be the datatype,

```
[1] character
```

Note: The datatypes in R language change dynamically. Let us see an example,

```
# Consider assigning a value of 12000 to a variable sigma and during the execution of the program assign a character to the same variable

>sigma<- 12000

cat ("The class of sigma is", class(sigma)," \n")

>sigma=" character"

cat ("The class of sigma is", class(sigma)," \n")

>sigma<-14.5

cat ("The class of sigma is", class(sigma)," \n")
```

Now, please execute the code and check the output in the console.

```
[1] The class of sigma is Integer
[2] The class of sigma is character
[3] The class of sigma is numeric
```

A variable can be deleted, or the assigned value can be removed by using the unbuilt function rm (). After executing the function, if we type the variable it will indicate an error.

```
# Remove a variable
>Hours=60
>rm (Hours)
```

Executing the above-mentioned command will give an error because the variable has been removed. Type the variable name in the console to check the output.

```
>Hours

Error: Object 'Hours' not found
```

2.1.1.1 Exercise

1. The inbuilt function used to remove an assigned variable is_____.
2. The inbuilt function to find the datatype of a variable is_____.
3. Is A=15 and a= 45 same? _____
4. In the following code, b=" Hi How are you?", what is the datatype of b?
5. What is the syntax of printing a statement in R language? _____
6. What is the datatype for decimal values in R language_____?

Programming Assignment*: Write a sample program for creating a simple biodata consisting of Name, Height, Weight, Aadhar ID, blood group and state.*

2.2 Data Structures

2.2.1 Vectors

Vectors are the most widely used data structure in R language. A vector in R language is a collection of numbers.

```
# Create a vector containing names and numbers
>A<- c (1,2,3,4,5)
>D = c (“Naveen”, “Sandeep”, “Gokul”)
>print(A)
>print(D)
```

If we execute the code the following outputs are displayed in the console.

```
[1] 1 2 3 4 5

[2] Naveen Sandeep Gokul
```

The function c () concatenates all the numbers and prints as a single vector. Similarly, if we need to generate a sequence of numbers, then we need to use seq (). If we want to generate numbers between a range, then the colon operator helps us to do it.

```
# Create a vector containing names and numbers
>A<- 1:10
>print(A)
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Similarly, let us try to generate a sequence using seq () function.

```
# Create a vector containing names and numbers
>A=seq (1,10)
>print(A)
```

If we execute the command the output is printed in the console as a single array consisting of numbers ranging from 1 to 10.

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Individual element of the vector can be accessed by giving the following command variable name [position of the value].

```
# Create a vector containing numbers and print the 3rd number
>A=seq (1,10)
>print (A [3])
```

If we execute the command the output is printed in the console as a number is printed.

```
[1] 3
```

2.2.2 Arrays

In R language Array is a data object which can store data in many dimensions. We can create an array using the function array (). In general, we need to create vectors and give the dimensions of the array. The arguments required for the array () function are two vectors and the dimension of the array.

```
# Create an array
>A<- c (1,2,3)
>B= c (4,5,6,7,8,9)
> C=array (c (A, B), dim (3,3,1))
Print(C)
```

If we execute the above-mentioned script, it prints a matrix of 3x3 matrix.

```
     1    2    3
[1] 4    5    6
     7    8    9
```

2.2.3 Matrices

In R language Matrices are portrayed as a 2-dimensional data structures, these are like a vector but additionally they have the dimensional Input. The inbuilt function to create a matrix in R language is matrix (). The dimension of a matrix always involves the number of rows and columns. Any one dimension Is enough, either the number of rows or the number of columns, the other dimension will be derived based on the length of the matrix elements. So, the syntax to create a matrix is shown below.

```
>matrix (10:19, nrow=3, ncol=3)
```

The output of the above-mentioned operation will be a 3x3 matrix with elements ordered in column wise.

```
      10    13    16
[1] *11     14    17,
      12    15    18
```

Under keen observation it can be deduced that the matrix is being filled column wise, to fill a matrix row wise the syntax takes in a Boolean input byrow = TRUE

```
>matrix (10:19, nrow=3, byrow=TRUE)
```

Now the elements in the matrix are filled row wise.

```
      10    11    12
[1] *13     14    15,
      16    17    18
```

An alternative way of filling a matrix is using the inbuilt functions cbind (), r(bind). cbind () combines two vectors into matrix by filling in the columns. Similarly, the rbind () matrix combines the vectors to form a matrix by filling row wise.

```
> cbind(c  (10,11,12), c(13,14,15))

>rbind(c  (10,11,12), c(13,14,15))
```

Once we executed the above syntax, the output will be a 3x2 and a 2x3 matrix.

```
      10    13
[1]   11    14,
      12    15

[2] -  10    11    12
       13    14    15
```

The third methodology to create a matrix in R language is by setting the dimensions using the dim () function.

```
>a<- c (1,2,3,4,5,6,7,8,9)

>dim(a) <- c (3,3)

>print(a)
```

The inputs given to the dim () function will be the number of rows and columns, the above syntax will create a matrix of 3x3 using the elements of the variable a.

```
     1    4    7
[1] *2    5    8,
     3    6    9
```

2.2.3.1 How to access individual elements

The individual elements of the matrix can be accessed by using the variable name and square bracket. For example, if we have a matrix A, then the individual elements can be accessed by A [row, column]

```
       10    11    12
>A= *13      14    15,
       16    17    18

> b= A [1,]
# The above operation will print a matrix which has only the first row but as a vector

b=10    11    12

> d= A [-1,]
# The above operation will print a matrix which excludes the first row

d= -  13    14    15
      16    17    18

>class(A)

> [1] "matrix"

>The output of class(A) will be "matrix"
```

The dimension of the matrix can be found by the inbuilt function dim () and the class of the matrix can be found by the inbuilt function class ().

2.2.3.2 Matrix Operations

2.2.3.2.1 Matrix Addition

The matrix addition in R language is a simple operation which involves creating two matrices and adding them by a simple '+' operator.

```
>A= matrix (c (1,2,3,4), nrow = 2)

>B=matrix (c (2,3,4,5), nrow = 2)

>Print(A)

>Print(B)

>C= A+B

>print(C)
```

The above-mentioned program creates two square matrices using the matrix () inbuilt functions. Later both the matrices are added using '+' operator. The output is stored in a variable named C

```
[1] - 1  3
      2  4

[2]  2  4
    -3  5

[3] 3  7
    5  9
```

2.2.3.2.2 Matrix subtraction

The subtraction operation in R language works in the same fashion as the addition operation. The operator'- 'is used to perform the operation.

```
>A= matrix (c (1,2,3,4), nrow = 2)

>B=matrix (c (2,3,4,5), nrow = 2)

>Print(A)

>Print(B)

>C= A-B

>print(C)
```

The above-mentioned program creates two square matrices using the matrix () inbuilt functions. Later both the matrices are added using '+' operator. The output is stored in a variable named C

```
[1] -  1   3
       2   4

[2] -  2   4
       3   5

[3]  -1   -1
     -1   -1
```

2.2.3.2.3 Matrix Multiplication

Matrix multiplication is different from the multiply operator, Since the operation of matrix multiplication is different from the normal multiplication, we must provide the symbol % before and after the multiply operator.

```
>A= matrix (c (1,2,3,4), nrow = 2)

>B=matrix (c (2,3,4,5), nrow = 2)

>Print(A)

>Print(B)

>C= A%*%B
```

```
>print(C)
```

```
[1] -1  3
     2  4

[2] -2  4
     3  5

[3] 11  19
    18  28
```

2.2.3.2.4 Matrix Determinant

The determinant of a given 3x3 matrix is done by the inbuilt function det (). The function takes in the input of the matrix and returns a value which will be the determinant of the respective matrix.

```
>x=c (3,1,2)

>y=c (-1,2,3)

>z=c (1, -2,1)

>eqn = matrix(c(x,y,z), ncol = 3)

>print(eqn)

> D=det(eqn)
```

The above operation creates a 3x3 matrix and performs determinant operation, it finally returns a value which is stored in variable D.

```
     3   1   2
[1] -1   2   3
     1  -2   1

[2] 28
```

2.2.3.2.5 Matrix Inverse

The inverse operation of a matrix is done by the inbuilt function solve (). The operation includes creating a matrix of 3x3 form and giving that matrix as an input to the function solve ().

```
> A=matrix (c (1:4), nrow = 2)

>print(A)

>B=solve(A)

> print(B)
```

The above-mentioned program will create a square matrix A with the given elements, next it will perform the inverse operation on the matrix, and it will print the determinant value.

```
[1] -1    3
     2    4

[2] . −2   15
      1   05
```

2.3 Basic Arithmetic Operation

2.3.1 Addition

The Addition operation in R language is performed using the inbuilt function sum (). This operation includes creating two variables with numeric or integer datatype and giving the variables as input to the function sum ()

```
>current_price = 15000

> interest = 1500

>Total_amount = sum (current_price, interest)

>print (Total_amount)
```

In the above-mentioned program, we have created two variables namely current_price and interest each with its own value. Now these values are summed up using the function sum () and the output is stored in a variable Total_amount which is printed in the end.

```
[1] 16500
```

2.3.2 Subtraction

Unlike the sum () function for the addition operation, R language doesn't provide any inbuilt function for the subtraction operation. It is the simple '- 'operator.

```
>Total_amount = 20000

>Expenditure = 8000

>Remaining = Total_amount – Expenditure

> print (Remaining)
```

The above-mentioned program creates two variables Total_amount and Expenditure each with values of 20000, 8000 and the '- 'operator is used to subtract the variables. Finally, the output Is stored in a variable Remaining and is printed.

```
[1] 12000
```

2.3.3 Multiplication

Multiplication operation in R language is performed by the inbuilt function prod (). The arguments taken by the function are simply either two values or two variables containing values.

```
>a=15

> b=20

>C=prod(a,b)

>print(C)
```

The above-mentioned program creates two variables a=15 and b=20 and these variables are passed as arguments for the inbuilt function prod () and the output is stored in a variable C.

```
[1] 300
```

2.3.4 Average

The average of the given elements is determined by the inbuilt function mean (). The arguments to the functions will be a vector of elements. The output will be the average of all the elements.

```
> a1= 10

> a2 = 20

> average = mean (c (a1, a2))

> print(average)
```

The above-mentioned program will create two variables a1 = 10 and a2 = 20, these variables are passed as inputs for the mean () function and the output is printed

```
[1] 15
```

2.3.5 Square root

The square root operation in R language is performed by the inbuilt function sqrt (). The inputs to the function will be a simple variable consisting the value or a number itself.

```
>a=15
> b=20
>C=sqrt(a)
>D=sqrt(b)
>print(C)
>print(D)
```

The above-mentioned program will create two variables a=15, b=20 and it will give these values as inputs to the function sqrt () and the output is printed finally.

```
[1] 225
[2] 400
```

2.3.6 Logical Operations

The logical operations generally involve AND, OR, NOR, NOT, NAND, EXOR etc. These operations generally give a Boolean output.

2.3.6.1 Logical AND operation

The logical AND operation in R language is performed by the & operator. The basic concept of AND operation is multiplication. Its written as,

$$Y = A.B$$

Where,

Y is the output

A and B are the inputs

The working of the operator is same as its truth table.

Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

The output is going to be either TRUE or FALSE.

```
>a= c (3, TRUE,0,1)

>b= c (4, FALSE,1,0)

>print(a&b)
```

The above-mentioned program has two vectors are of mixed values for AND operation and the output are printed finally.

```
TRUE FALSE FALSE FALSE
```

2.3.6.2 Logical OR operation

The logical OR operation in R language is performed by the | operator. The basic concept of OR operation is addition. Its written as,

$$Y= A+B$$

Where,

Y is the output

A and B are the inputs

The working of the operator is same as its truth table.

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

The output is going to be same as previous as in TRUE or FALSE.

```
>a= c (1:10)

>print((a<4)  |(a>7))
```

The mentioned program performs AND operation for the range provided for which the values outside the range is taken as positive input and the one within range as inverted input and prints the output.

```
TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
```

2.3.6.3 Logical NOT operation

The logical NOT operation in R language is performed by the! operator. The basic concept of NOT operation is inverting. Its written as,

$$Y = \bar{A}$$

Where,

Y is output and

A is input

The working of the operator is same as its truth table.

Input A	Output Y
0	1
1	0

The output is going to be same as previous as in TRUE or FALSE.

```
>a= c (1,0)

>print(! a)
```

The above program performs NOT operation of the inputs and the output is printed.

```
FALSE TRUE
```

2.3.7 Trigonometry

The basic operations of sin, cos, tan involves the concept of trigonometry. These generally gives the output in the form of angle.

In general, the ratio of the opposite side of the angle to be found to that of hypotenuse is used to find sin component. In R language the representation is direct form as sin, cos and tan. The important point to be noted is that, if the representation of any angle is in the form of radians.

```
>building.height= 40

>ladder.angle= 60

>ladder.height= building.height/sin(pi/3)

>ladder.distance= ladder.height*cos(pi/3)

>Building.height= ladder.distance*tan(pi/3)

>print(ladder.height)

>print(ladder.distance)

>print (Building.height)
```

The above-mentioned program performs the trigonometry operations and the outputs are visualized.

```
46.18802
7.1245
45.6347
```

2.3.8 Exercise

1. Write a program describing the annual cost towards the company for an employ working in a multinational company. The final CTC should include his Basic pay of 25000, Food allowance of 5000, Dearness allowance of 10000, Travel allowance of 10% of basic pay and a PF amount of 14% of the basic pay.

   Hint: Use the sum () and find out the final CTC

2.4 Functions

2.4.1 Creating Function in R Studio

To create a function in R language we need to use the syntax function () which takes the input as a variable which is to be entered.

```
> a= function (x) {

      b=x^4
      print(b)
}
```

In the above-mentioned function, a is the name of the function and x is the input argument of the function. Inside the function we define the operation the function needs to achieve. In this case, once the user gives the input a(x), the function calculates the square of the input and stores it in the variable b. Finally, it prints the b value. So, after executing the command, we need to give an input to the function as for which values of x should the calculation be performed.

```
a [30] # so as per the function when we give an x value, the calculated value is displayed

[1] 900
```

3 Data Pre-processing

Data pre-processing is a very important step before performing a data analysis related task. The data available outside has tremendous amounts of noise, inconsistency, junk and most of them are present in different ranges in various databases. This is one of the steps which is ignored more often, this leads to unwanted noise in the data which will increase the error. Some of the most important data stages of data pre-processing are as follows

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

3.1 Data Cleaning

In real life, the data available is very noisy and has garbage in it. Garbage means the redundant data which will create aberration of the learning process. The whole idea of data analysis is to find hidden pattern in a cluster of data. These junk values will create unwanted patterns which will in turn create erroneous inference of the given data. One of the most important steps is data exploration. In data exploration, we import the dataset and try to understand the data. R studio allows many formats of data like.csv, xlsx, yppl and so on.

3.1.1 Importing data in RStudio

To import the dataset, we need to follow three steps

1. On the right top window, in the environment window, click on the option 'Import Dataset'

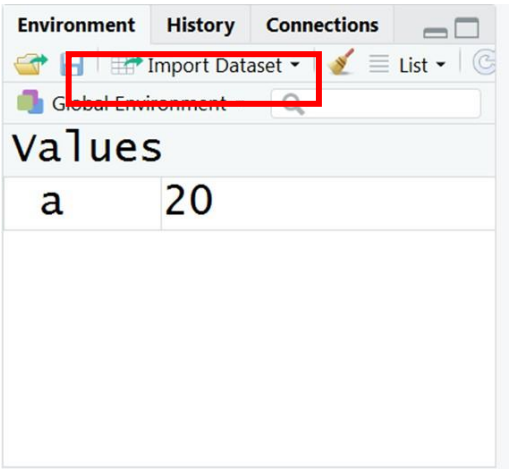

Figure 3.1steps to Import Dataset

2. Click on the data format. By default, R studio provides few formats. For example, let us get data from Excel

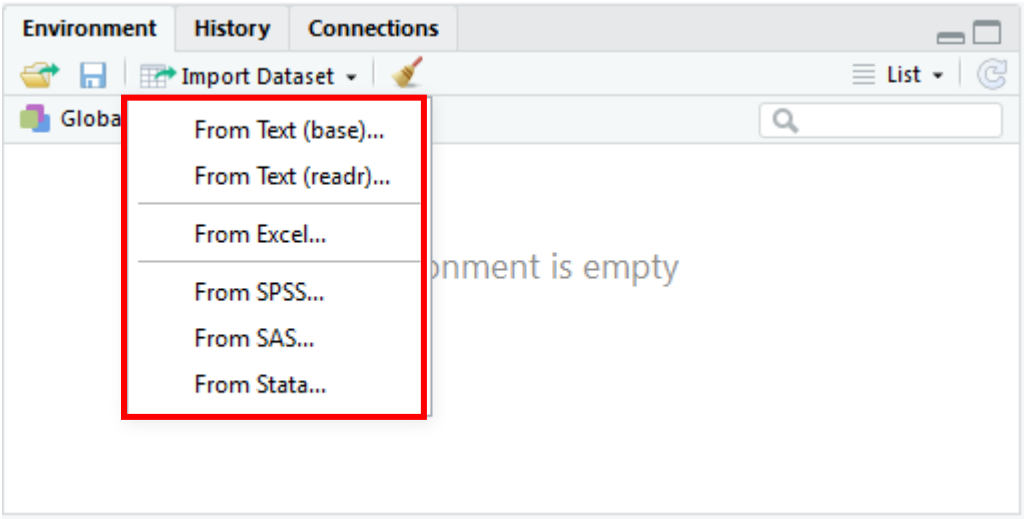

Figure 3.2 steps to Import Dataset-Import Dataset

3. Click on browse to find the dataset in the stored folder.

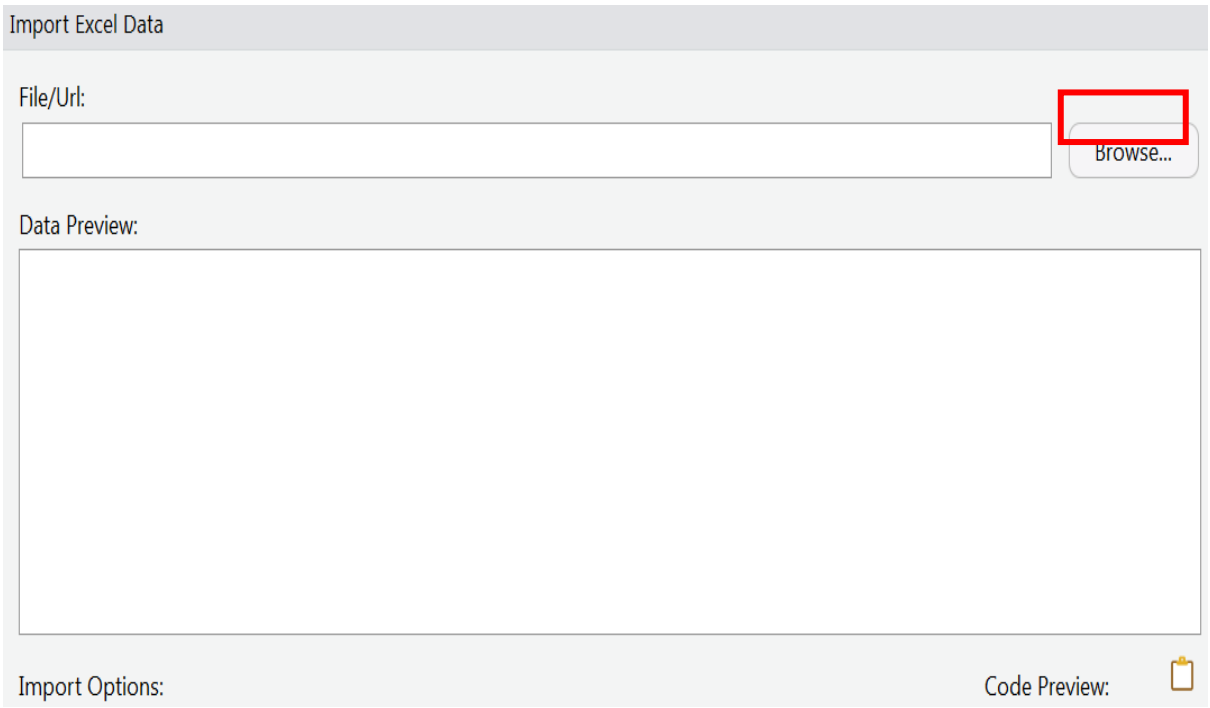

Figure 3.3 steps to Import Dataset-Import Excel sheet

4. A preview of the data is shown in the Data preview window

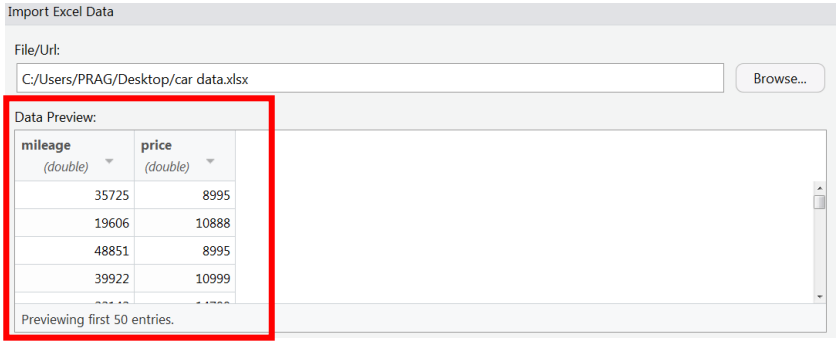

Figure 3.4 steps to Import Dataset-Data Preview

5. Finally click on import if there is no problem with the data. The data is viewed in a new window

	mileage	price
1	35725	8995
2	19606	10888
3	48851	8995
4	39922	10999
5	22142	14799
6	105246	7989
7	34032	14490
8	32384	13995
9	57596	10495
10	63887	9995
11	58550	12921
12	40527	12000

Figure 3.5 steps to Import Dataset- Dataset

Now the above-mentioned image shows the imported data

3.1.2 Data visualization

This is one of the powerful tools in R Language. Data visualization is the next step in data cleaning where we try to have a graphical representation of the data. Some of the powerful data visualization functions in R language include plot (), hist () and histogram

3.1.2.1 Scatter plot

The scatter plot in R language is done by the inbuilt function plot ()

```
>library(readxl)

>dataset <- read_excel(dataset_name)

>View(dataset)

>plot(dataset_name)
```

The above-mentioned program will generate a scatter plot which plots all the datapoints with x and y axis.

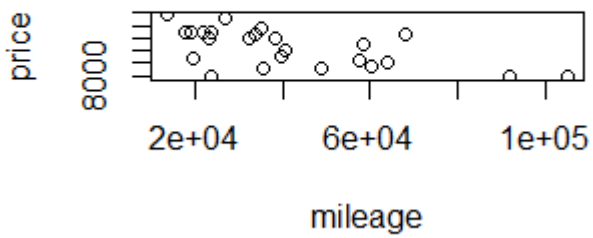

Figure 3.6 Scatter Plot

3.1.2.2 Bar plot

The function barplot () generates a bar graph of the dataset

```
>x=c(1,2,3,4,5)

>y=c(30,70,100,150,200)

>barplot(x,y)
```

The above mentioned program will generate two vectors x,y and in turn will put them as inputs for the bar graph using the function barplot()

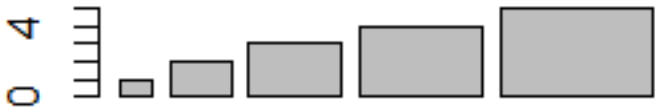

Figure 3.7 Bar plot

3.1.2.3 Histogram

The hist () function generates a histogram of the data. This helps us to understand how the data is distributed along the data spectrum.

```
>library(readxl)

>dataset <- read_excel(car_data)

>View(car_data)

>hist(car_data$price)
```

The above-mentioned program will import a dataset named car_data, then hist () function is applied on one of the dependent attributes to find their distribution

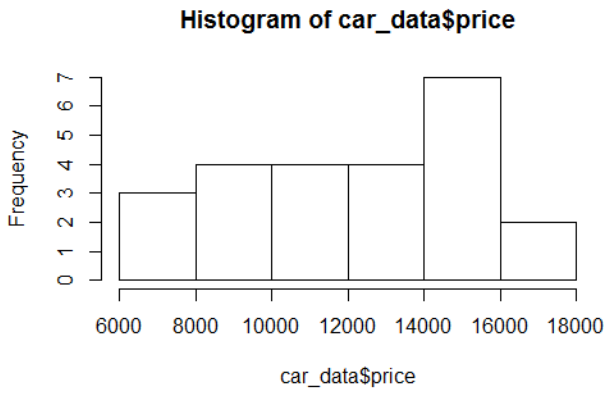

Figure 3.8 Histogram Plot

3.1.2.4 Pie chart

The pie chart allows user to understand what percentage an entity covered in the whole distribution. The pie chart is generated using the inbuilt function pie (). The input of the function will be two vectors containing the string names and the numeric values. The output will be a pie chart

```
>language=c("Telugu","Tamil","Hindi","English")

>Percentage=c (25,35,20,20)

>pie (Percentage, language)
```

The above-mentioned program will create two vectors with languages and percentages of their occupancy. Finally, the variables are given as inputs to the inbuilt function pie ()

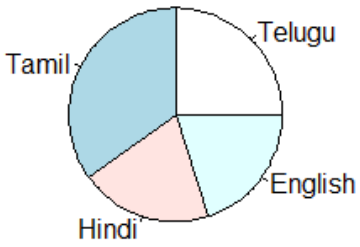

Figure 3.9 Pie Plot

3.1.2.5 Line plot

The line plot in R language is generated by the inbuilt function plot (). The input given to the function are the x, y axis values, type of point and the colour of line.

```
>Hours=c (1,2,3,4,5)

>Distance= c (100,200,300,400,500)

>plot (Hours, Distance, type=" o", col = "blue")
```

The above program will create two variables with their respective values. To generate the plot we use the function plot ().

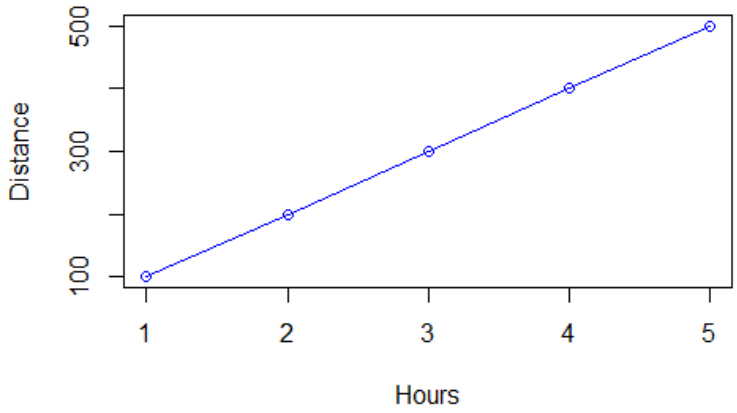

Figure 3.10 Line Plot

3.1.3 Data correction

The next step of the process figuring out whether the data has the following data.

- Missing values
- Categorical values

3.1.3.2 Missing Values

In the complete dataset, there is a high possibility to find uneven distribution of the data or some data might be missing where such data could make some difference in the overall data model. A data with the missing values looks as follows

Data Preview:

mileage (double)	price (double)
35725	8995
19606	NA
48851	8995
39922	NA
22142	14799

Figure 3.11 Missing Data

In the figure the word NA is the acronym for Not Available. It means the data is not available for the respective inputs. Now this missing value will create an error, so these missing values will be filled using various statistical operations, one of them would be finding the mean of the dataset and replace it.

```
>library(readxl)

>dataset <- read_excel(car_data)

>car_data$price=ifelse(is.na(car_data$price),  ave (car_data$price, FUN = function(x) mean (x, na.rm = TRUE)), car_data$price)

>View(car_data)
```

The above-mentioned program will check for the term Not available and it will replace with the average of the data. After the operation the missing values are filled, and the data looks as follows

	mileage	price
1	35725	8995.0
2	19606	10194.5
3	48851	8995.0
4	39922	10194.5
5	22142	14799.0

Figure 3.12 Missing Data – Replaced values

3.1.3.3 Categorical Values

In general, if we observe a dataset, it has numerical values as well as strings. Now these strings cannot be given as input for any mathematical model. So, these must be converted into numbers by numerically organizing them in categories. The process involves generating vectors of various strings or characters then giving specific categories for the data. For example, if we consider a data which has states of a nation which is an important feature set. This data cannot be given as such to the learning model. These state names must be converted into categories

Table 1 Categorical Data-Labels

Status	Category
PASS	**0**
FAIL	**1**

The following figure gives an intuition of how a categorical data looks like. The highlighted column consists of categorical data of FAIL and PASS

	S.NO	Total score	Status
1	1	33	FAIL
2	2	45	PASS
3	3	48	PASS
4	4	99	PASS
5	5	30	FAIL

Figure 3.13 Categorical Data

```
>library(readxl)

>dataset <- read_excel(MARKS)

>MARKS$Status  =factor (MARKS$Status,  levels = c ("FAIL", "PASS"),  labels = c(0,1))

View (MARKS)
```

The above-mentioned program takes the input as the column which has the categorical data to be and converts into categories of 0 and 1. The inbuilt function we use for this is factor (). The input to this function would be the specific column, the labels or the names and the categories or levels. So, after the operation the data looks as follows

	S.NO	Total score	Status
1	1	33	0
2	2	45	1
3	3	48	1
4	4	99	1
5	5	30	0

Figure 3.14 Categorized values

3.2 Data Integration

For data analysis the most key ingredient Is the data itself. This data must be taken from a broad spectrum of entities, sources etc. Data integration means gathering data from various sources available online or customer surveys, several databases and bringing them to one centralized database. Finally, this data is uniformly spread out to be made presentable to the end user. One of the most important concerns of data integration would be how the data is represented in various databases. For example, if the name of a student is represented in a database like student_name and in some other database it says candidate_name, how will the computer know that both the data are the same. All these issues are addressed during data integration.

3.3 Data Transformation

The data gathered from numerous resources might have redundant values, categorical values and values which create noise in the pattern. So, the data transformation steps in. Data pre-processing involves:

- **Normalizing the data**: The data gathered might be in a very big range often uncomfortable for computation and representation. So, in normalization the data is brought within certain limits. For example, the data ranging from -10000 to +900000 might fall under the range of -1 to +1 after normalization.

- **Smoothing the data**: Some unwanted datapoints with in dataset might lead to uncertainty in the solution and aberrations during the learning stage. So, these aberrations can be terminated using smoothing methods.

3.4 Data Reduction

Effective analysis of the given data often takes more time due to limited computation power and the complexity of the data. The complexity of the data in turn plays a critical role in the computational time. To counter this, the next step in Data pre-processing is Data Reduction where the data must be

reduced to a good amount, but this shouldn't compromise the integrity of the database and the final output. Some well-known methods could be

- **Dimensionality Reduction**: In dimensionality reduction, the dimension of the dataset is reduced by reducing the attributes which are redundant and unwanted. This step also helps the user to find the most important attributes on which the data output majorly depends upon.

- **Data compression**: Data compression methods include reducing the size of the data by transformation and compression techniques. Some of the most widely used techniques include Principal component analysis and Wavelet transform.

3.5 Exercise

1. _____ is a process of reducing the attributes in the given dataset.
2. The missing values of a dataset in R language can be replaced by_____of the data.
3. The inbuilt function to make a pie chart for the given data.
4. What are the four data pre-processing techniques?

5. The inbuilt function to draw a scatter plot is_____.
6. What does is.na mean _____

1. Write a program in R language to plot a scatter plot for the following data

Day	Temperature(°c)
1	46
8	38
12	42
15	31
18	40
26	44

4 Introduction to AI and Machine Learning in R language

4.1 What is Artificial Intelligence?

Artificial Intelligence is simply the intelligence demonstrated by machines. These systems are in turn called as intelligent systems. The definition of Artificial Intelligence is not straight forward. It's been derived from various approaches like the school of thought, school of philosophy, psychology and cognitive science. One of the widely famous techniques to know whether a system is intelligent or not is the Turing test.

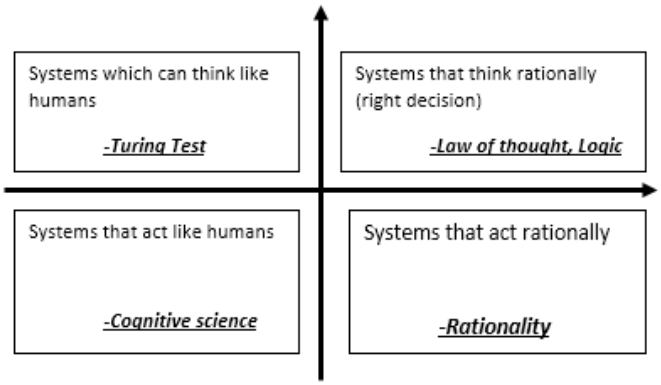

Figure 4.1 Definitions of Artificial Intelligence

4.1.1 Turing Test

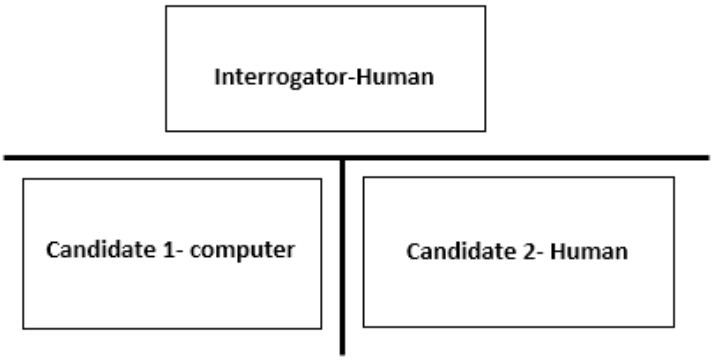

Figure 4.2 Standard Turing Test

Designed by Sir Alan Turing, Turing Test is a test of an artificial system or machine's ability to exhibit intelligent behaviour indistinguishable from humans. This game is called the Standard Turing test.

4.1.1.1 Standard Turing Test

According to the imitation game, there will be three players

- Player 1- Human (Male)
- Player 2- Human (Computer)
- Player 3- Human (Interrogator)

During the test, the players cannot see each other, the task is that the interrogator asks questions to both the players and the players responds to them. Now the trick is that the interrogator has no idea of who is giving the response. It could be either a human or a computer. Now according to the above-mentioned definition, we call a machine to be intelligent if the machine tricks the interrogator and the interrogator believes that the response he got is from a human being, but it came from a machine.

4.1.2 The idea of Rationality

A different school of thought says that if a machine has the ability to perceive the environment with the help of sensors and understand the problem existing in front of it, then given a set of actions and a performance measure, a machine is said to be intelligent if it takes a rational action or decision which will increase its performance measure. But what do you mean by Rationality?

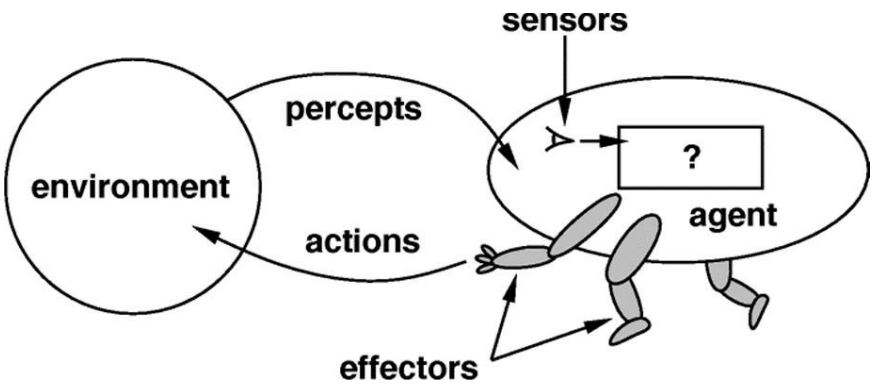

Figure 4.3 Intelligent Agent

4.1.2.1 Rationality

When an agent is said to take a decision, given a set of decisions, an agent is said to be a rational agent if it chooses a decision which will maximize its performance measure or its goal. Such an agent is called rational agent.

4.2 Machine Learning

Now when can you say that a person takes a right decision or a rational decision. The school of thought says a rational decision can be taken when the system can have logical reasoning. But another way of taking a rational decision could be experience. It is a universal fact that a person will gain the ability to take right decision if he has encountered the same problem multiple times. As the person encounters in a different instance, his experience will improve his rationality index. The same rule works for artificial systems or machines. This defines the subset of artificial intelligence, Machine Learning. The systems which learn based on experience.

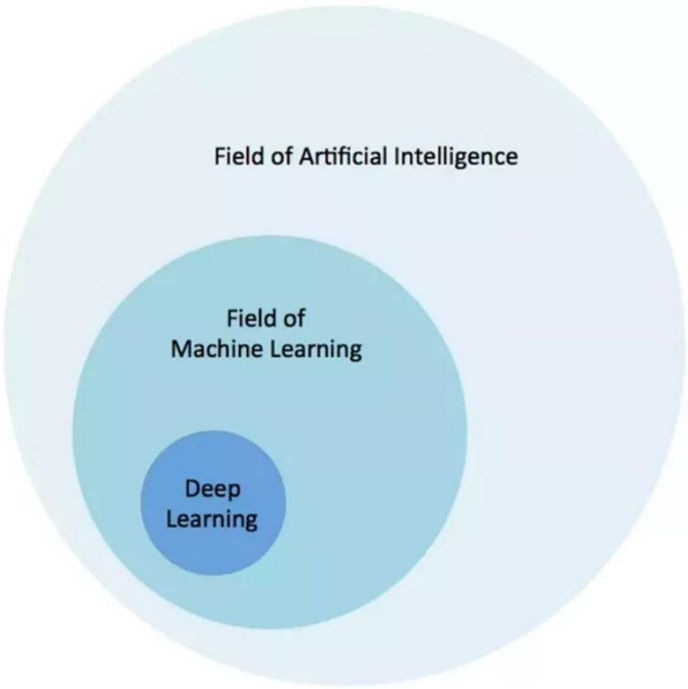

Figure 4.4 Subsets of Artificial Intelligence

According to Tom Mitchell, A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E

4.2.1 Difference between classical programming and Machine learning

Let us compare the traditional programming and a system working with machine learning algorithm. In traditional programming approach, once the problem is given, the programmer tries to define an algorithm and flow chart explaining the approach. So, the inputs to the system will be the data and the algorithm that is applied to the data. The output of the system will be the answer to that problem which will be processed in the algorithm. This approach has been followed for ages, one of the important limitations of the traditional programming is that as the input problem to be solved gets complicated the algorithm required becomes complex, which is a difficult and a mundane approach. Now the new approach to counter this problem is the machine learning approach, in this approach the inputs to the machine will be both the input data and the output data, which will give enough experience to the system. The output coming out of the machine will be an algorithm developed by the system with the help of the experience.

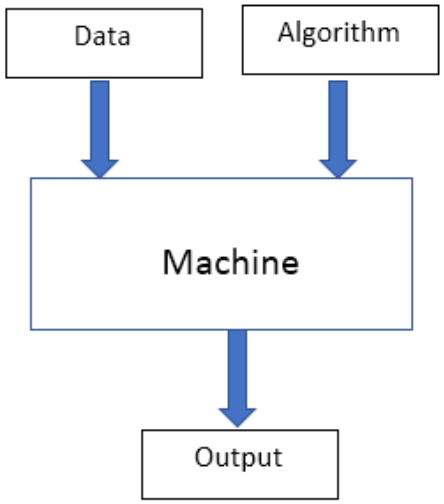

Figure 4.5 Traditional Programming Approach

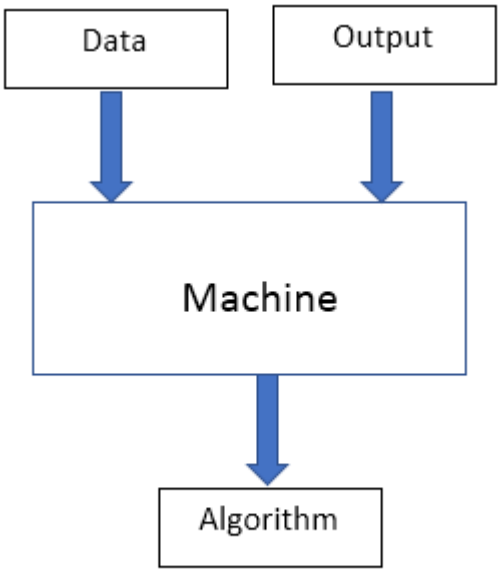

Figure 4.6 Machine Learning Approach

In this approach, programming hundreds of lines of code is not necessary. The data consists of either input and output or only input based on the type of learning. The learning ability of the data depends on the quality and quantity of the data provided to the system.

4.2.2 Types of Machine Learning

Based on the type of data provided and the output, the machine learning is divided into four paradigms.

- Supervised Learning

- Unsupervised Learning
- Semi Supervised Learning
- Reinforcement Learning

4.2.2.1 Supervised Learning

In supervised Learning, when a problem is defined the input features and the corresponding outputs are given to the system. The learning part of the process is based on this data, based on these inputs the machine develops a model or hypothesis. Now during the test stage, an input whose output is not known is fed into the hypothesis and the value is predicted or classified based on the type of data.

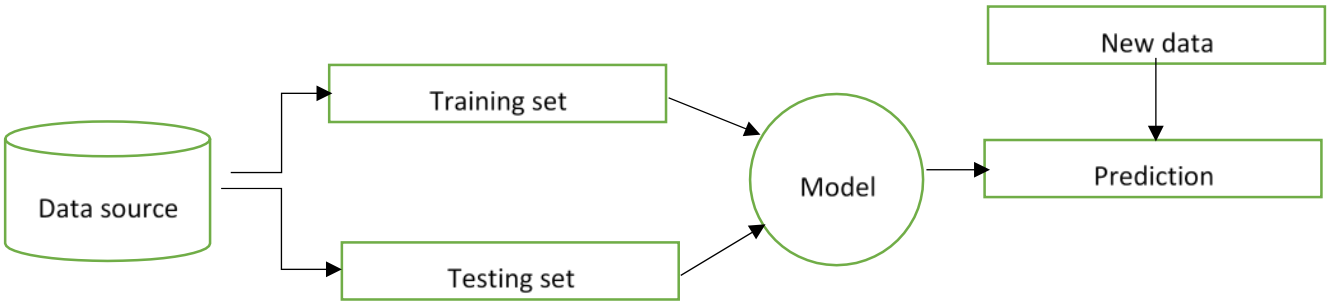

Figure 4.7 Supervised Learning

4.2.2.2 Unsupervised Learning

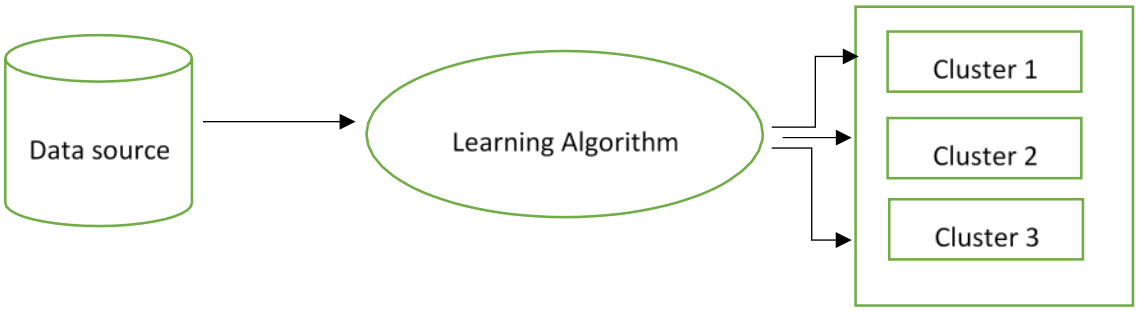

Figure 4.8 Unsupervised Learning

In unsupervised learning, only input data is fed into the system, no corresponding output data. Based on the type of the input and their categories, the learning model tries to perform clusters of the data, when a new input is given the system tries to fit it into any one of the clusters based on the input data characteristics.

4.2.2.3 Semi supervised Learning

In semi supervised learning, the data provided to the system consists of both labelled data and unlabelled data.

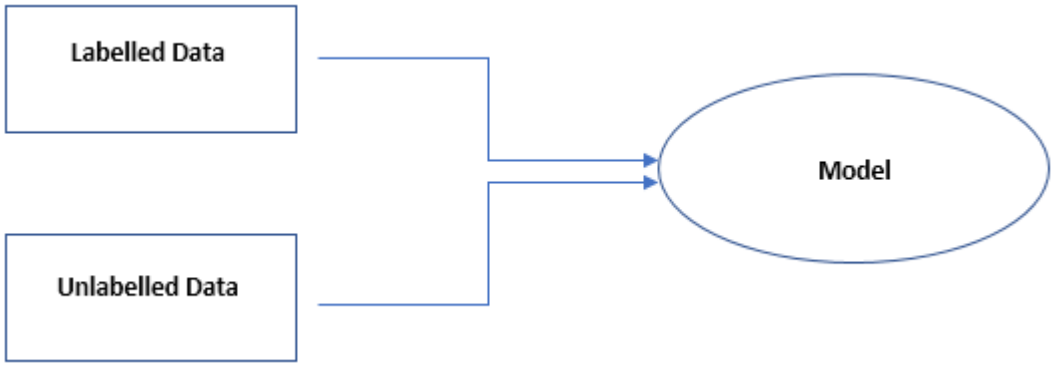

Figure 4.9 Semi-Supervised Learning

4.2.2.4 Reinforcement Learning

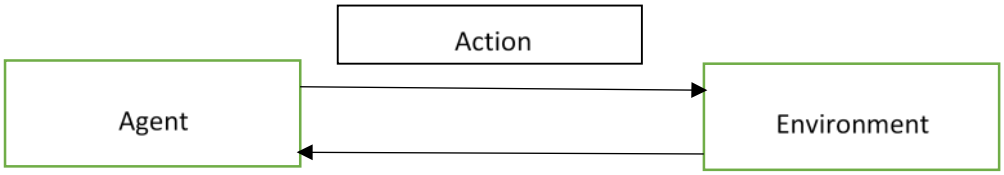

Figure 4.10 Reinforcement Learning

In reinforcement learning we have an agent which acts in the environment. The agent can act, and this action can impact the environment. In a stage, the agent takes an action and the environment goes to a new state and gives some reward to the agent, that reward may be a positive reward can be a negative reward or penalty or can be nothing at that time step.

4.2.3 Supervised Learning in R language

Let us examine the supervised learning in detail. Supervised Learning is further divided into two categories based on the output. Supervised Learning is often divided into a regression problem and a classification problem.

4.2.3.1 Regression

Regression is a technique to determine the statistical relationship between two or more variables where a change in a dependent variable is associated with, and depends on, a change in one or more independent variables.

Suppose the outcome of any process is denoted by a random variable y, called as dependent variable, depends on k independent variables denoted by X_1, X_2….X_k. Then the relationship between y (dependent variable) and X (independent variable) is explained by

$$y = f(x)$$

One of the most used applications of regression model is predicted analysis which us almost used in every industry. Given a set of data, the system will establish a relationship between the dependent and the independent variable and predict the future outcome of the dependent variable. The dependent variable could be a stock price, population of a nation, house price etc. In regression analysis most, important regression models are as follows:

- Linear Regression
- Polynomial Regression
- Multivariant Regression
- Support Vector regression
- Negative binomial regression
- Poisson Regression

Let us have a deep understanding of Linear Regression which has proven to effective in many applications.

4.2.3.1.1 Linear Regression

It is the simplest form of regression. It is a technique in which the dependent variable is continuous in nature. The relationship between the dependent variable and independent variables is assumed to be linear in nature. When you have only 1 independent variable and 1 dependent variable, it is called simple linear regression. The below mentioned equation explains the linear dependency between y (dependent variable) and X (independent variable). Generally, β represents the slope of the equations and C represents the y-intercept.

$$y = \beta X + C$$

When you have more than 1 independent variable and 1 dependent variable, it is called Multiple linear regression

$$y = \beta 1X1 + \beta 2X2 + \beta 3X3 + \in$$

4.2.3.1.1.1 Mathematical Model of Linear Regression

Consider y as a dependent variable and X as one of the features which is an independent variable. If y linearly depends on X, then it must follow the following equation:

$$y = \beta X + C$$

β = Regression Coefficient or Slope

C= y-Intercept

y=dependent variable

X= Independent Variable

β represents the slope of the equation. The mathematical formula used to find the slope is

$$\beta = \sum_{i=1}^{n} (xi - \bar{X})\,(yi - \bar{Y}) / \sum_{i=1}^{n} (xi - \bar{X})^2$$

Where xi = independent variable

$\bar{X}$ Mean of the total number of independent variables

yi= dependent variable

$\bar{Y}$= Mean if the dependent variable

n= total number of datasets

c is the y-intercept. It is defined as the point where the straight line is intercepting the y-axis

$$c = \bar{Y} - \beta\bar{X}$$

Where

$\bar{X}$= Mean of the total number of independent variables

$\bar{Y}$ Mean if the dependent variable

β = Regression Coefficient or Slope

4.2.3.1.1.2 Programming of Linear Regression in R Studio

Before we delve into programming of Linear regression in RStudio, we need to know the important steps required to perform linear regression in R studio.

Steps involved in Programming Linear regression in R language are as follows:

- Import data from various formats
- Check for categorical values and missing data, if found process it

- Import caTools package
- Generate Random numbers
- Divide the dataset into training set and test set
- Assign the training set and test set
- Fit the training set into the linear model function
- Train the system using the training set
- Predict the output using test set
- Data visualization

4.2.3.1.2 Polynomial Regression

Polynomial Regression is a form of regression in which the relationship between the independent variable and the dependent variable is modelled as the nth degree polynomial. It is a non-linear form of regression. This regression is used when the relationship between the dependent and independent variable is not linear. Modelling such a data with traditional linear method will produce error during the prediction stage. A polynomial fit will reduce the error and find a best fit which passes through maximum data points.

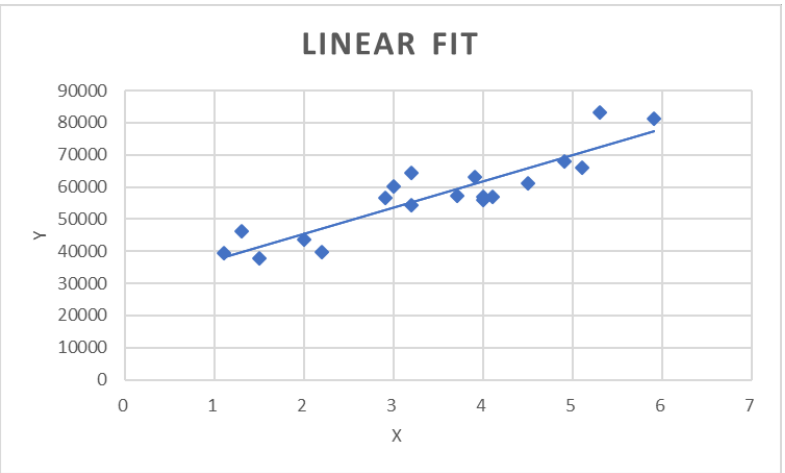

Figure 4.11 Best Linear Fit

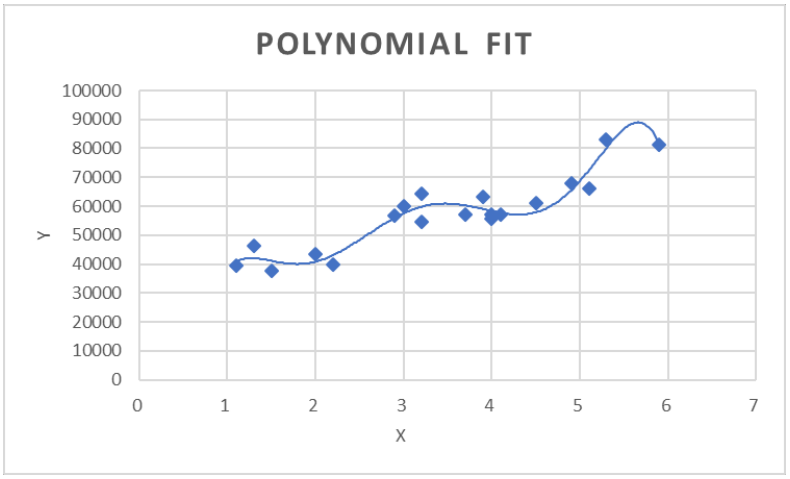

Figure 4.12 Best Polynomial Fit

As the order of the polynomial increases the curve fits well with the data. But one limitation with higher order polynomials is the overfitting of the curve which results in limited learning.

4.2.3.1.2.1 Mathematical Modelling of Polynomial Regression

Consider X_1, X_2,X_i be the input features for the outputs Y_1, Y_2,….., Y_i. The polynomial model would look as follows,

$$y_B = \beta_C + \beta_D x_B + \beta_4 x_B^4 + \cdots + \beta_F x_B^F + \epsilon_B$$

Now the same equation for many input and output pairs or feature set can be depicted as a matrix

$$\begin{bmatrix} y_D \\ y_4 \\ y_K \\ \vdots \\ y_F \end{bmatrix} = \begin{bmatrix} 1 & x_D & x_D^4 & x_D^O \\ 1 & x_4 & x_4^4 & x_4^O \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_F & x_F^4 & x_F^O \end{bmatrix} \begin{bmatrix} \beta_C \\ \beta_D \\ \beta_4 \\ \vdots \\ \beta_F \end{bmatrix} + \begin{bmatrix} \epsilon_D \\ \epsilon_4 \\ \epsilon_K \\ \vdots \\ \epsilon_F \end{bmatrix}$$

The same equation can be represented as

$$\vec{y} = X\vec{\beta} + \vec{\epsilon}$$

Now in the following equation, the data provided consists of the input vector X and the corresponding y and the only unknown in this equation to model the best fit is the coefficients. This coefficient matrix can be obtained by the inverse operation of the input vector,

$$\vec{\beta} = \vec{y}[X]^{TD}$$

4.2.3.2 Multivariant Regression

Linear Regression and Polynomial Regression are classified as Univariant Regression because we observe that there is only one independent variable and one dependent variable. In general, the output of a system often depends upon many parameters. For example, let us take the example of the prediction of salary of an individual. According to Univariant regression, the salary(dependent variable) depends only on the years of experience of the person, but in reality this strategy fails because if a person is paid based on his years of experience then the security assistant with 30 years' experience will be paid more than a higher level executive of greater knowledge. This policy will lead to underfitting and huge errors in the hypothesis. So, let us observe one more strategy, the same statement with multivariant regression involves information of all the parameters like specialization of the candidate, technical competencies, expertise etc. Even though the concept of linear regression and multivariate regression looks the same the consideration of attributes is more in case of multivariate. With only one independent variable the output has a high percentage of error in linear regression. It is known commonly that for any outcome there are always multiple parameters which affects its outcome. Considering all the parameters gives a better result. And the multivariate regression helps in achieving it.

Consider a problem of predicting the probability that a student applying for post-graduation in USA will get an admit in the university. The students are required to list down all the independent parameters on which the admit is dependent upon. (For example, the admit might be based upon his GRE score and TOEFL score)

Mathematics of Multivariate regression

Let's consider X_1, X_2, X_3, …. X_n be the different parameters for an output Y. The general equation of the multivariate regression is,

$$y = A_C + A_D x_D + A_4 x_4 + A_K x_K + \cdots + A_F x_F$$

y= Dependent Variable

$\{x1, x2, \ldots.\}$ = Independent Variables

$A_D, A_4, A_K \ldots A_F$=Coefficients

The regression is fitted with a method known as sum of squares. It basically calculates the square of the difference in each observation with the overall mean. The sum of square is done for the error, regression as well as the total sum of squares. The total sum of errors is the sum of both error and regression. With this, the generalised equation is formed and the fit is done accordingly. With more the number of attributes for an outcome the better the result will be.

Programming steps Multivariate regression in R programming

- Setting working directory.
- Importing dataset.
- Categorizing data.
- Import caTools package.
- Generate random numbers.
- Divide the dataset into two.
- Assign the divided data to training set and test set.
- Fit the training set into the linear model function.
- Train the system using training set.
- View the errors in the attributes using summary.
- Predict the output using test set.

Write down the R language code for Multivariant Regression

4.2.4 Classification

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi class. Some examples of classification problems are

- speech recognition
- handwriting recognition
- bio metric identification
- document classification
- Object detection

The basic difference between regression and classification is the representation of the output or the dependent variable. In regression the output is represented as a continuous function or the output is continuously varying with respect to the independent variable. A continuous function output looks like the below mentioned image.

Figure 4.13 Linear Regression

But in classification the output is discrete, it means in classification we try to classify the output based upon the input parameters. For example, the classification output looks like whether the received email is a spam or not, whether the animal shown in the image is a dog or not etc. The output of a classification-based machine learning problem looks as follows:

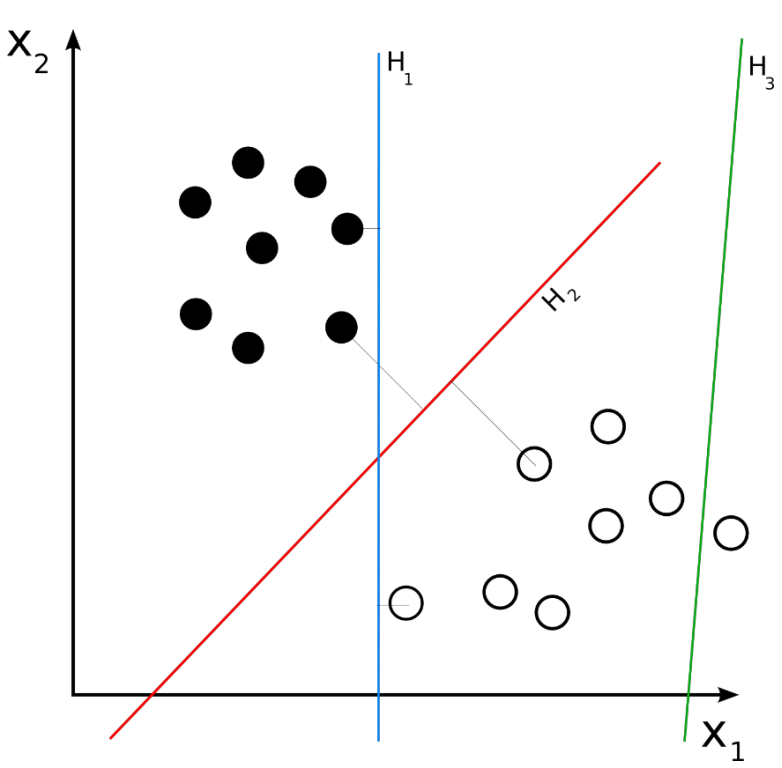

Figure 4.14 Classification

Some of the famous classification algorithms are

- Logistic Regression
- Decision Tree classification
- Naïve Bayes classifier
- Support Vector Machine
- Random Forest
- Artificial Neural Networks

4.2.4.1 Logistic Regression

In general, the inverse of an exponential function is said as log. The logistic regression is by far the most used for prediction in a binomial experiment. The prediction of population growth, stock market etc. are all done by this type of regression. Usually the output of logistic regression is said to be Yes/No or 1/0 or any two-way form. The graph is represented in a S-curve form.

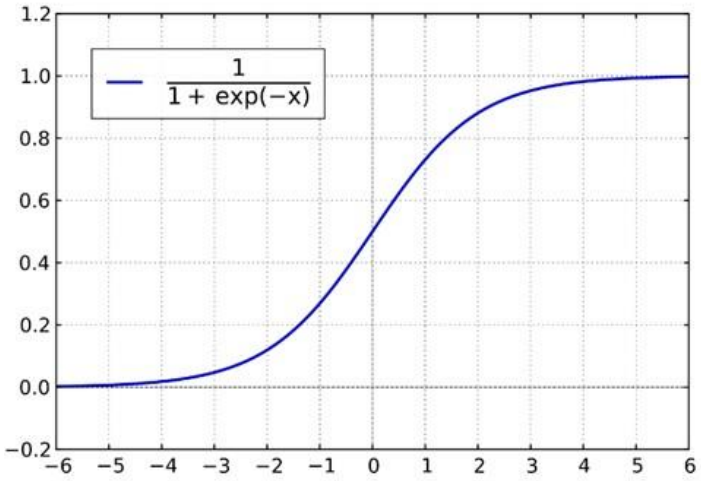

Figure 4.15 Sigmoid Function

The graph line is formed from the equation present in the picture which is a sigmoid function.

Mathematics of Logistic regression:

In this regression the probability plays a vital role. The output depends on the probability of the variable. So, the equation only depends on the success and failure. Let's consider a general equation as,

$H(y) = A_0 + A_1$

With this equation the exponent is taken at first and the probability of the success and failure is found. On dividing the probability rate, we get the equation logarithmic equation as,

$$y = \log \frac{p}{1 - p}$$

The equation of the sigmoid function which helps in the plotting of the S-curve is,

$$Sig(y) = \frac{1}{1 + e^{Ta}}$$

Steps to perform logistic regression in R programming:

- Setting working directory.
- Importing dataset.
- Categorizing data if necessary.
- Import caTools package.
- Generate random numbers.

- Divide the dataset into two with respect to categorical data.
- Assign the divided data to training set and test set.
- Scale the split data.
- Fit the training set into the generalised linear model function.
- Train the system using training set.
- View the errors in the attributes using summary.
- Predict the output using test set.

Write down the R language code for Logistic Regression

4.2.4.2 Support Vector Machine

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered as the critical elements of a data set. For a classification task with only two features, hyperplane is a line that linearly separates and classifies a set of data. The distances between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set.

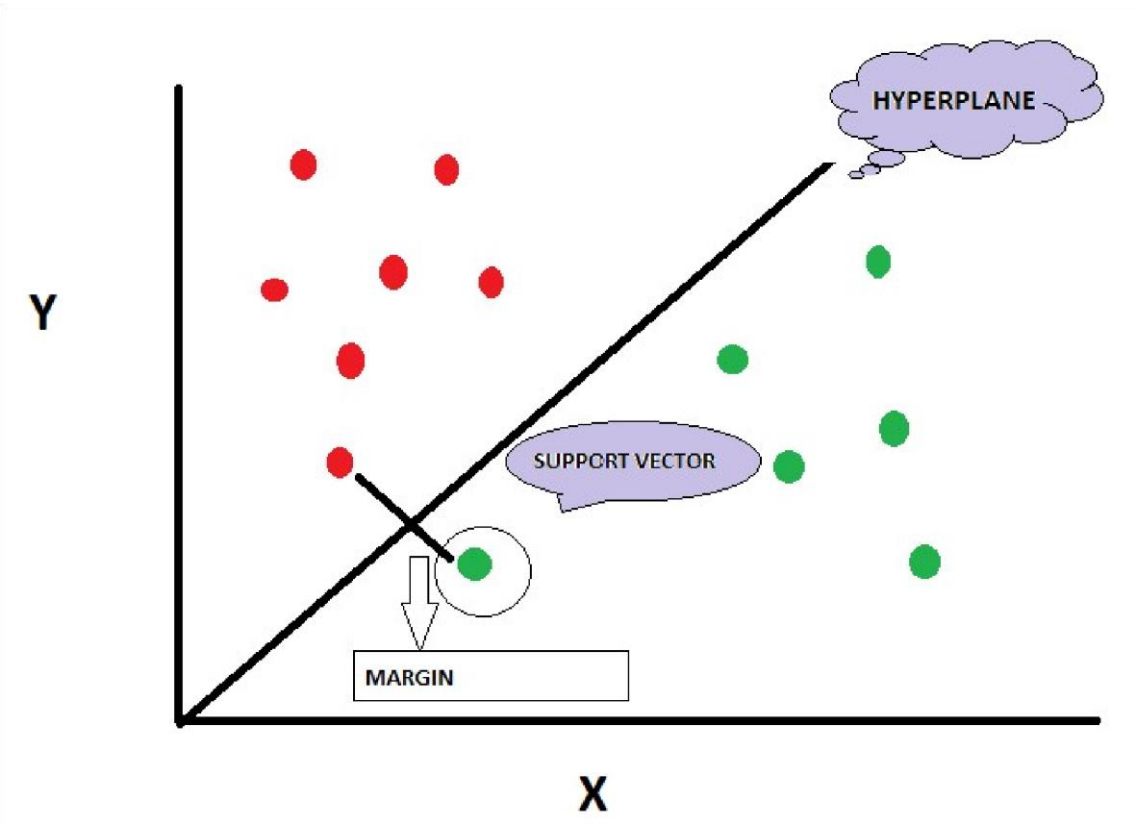

Figure 4.16 Support Vector Machine

Mathematics of Support Vector Machines:

The following two points are the simplest form of defining the hyperplane.

- First is to find the distance between the data points that are near to find the mid-point for the hyperplane.
- Secondly to find the magnitude and direction of the points that are nearby so that it helps in finding the angle at which the hyperplane will be formed.

Basically, when working the math in SVM the objective is formed by the sum of regularization function and loss function. So, when considering an equation at start the gradient descent method with partial differentiation is done and the changes of weights are done to have proper classification.

Even after arriving at a base equation to work with SVM we must apply some tuning parameter which are taken care by the kernel. Linear, radial based and polynomial are different types of Kernels which will make the algorithm accurate.

Steps to perform SVM in R programming:

- Setting working directory.
- Importing dataset.
- Import caTools, e1071, ggplot2, GGally package.
- Generate random numbers.
- Divide the dataset into two with respect to categorical data.
- Assign the divided data to training set and test set.
- Visualizing the dataset with correlation factors.
- Scale the split data.
- Fit the training set into the svm function.
- Train the system using training set for classification.
- Predict the output using test set.
- Create confusion matrix to relate and see the test set and the predicted set for classification.

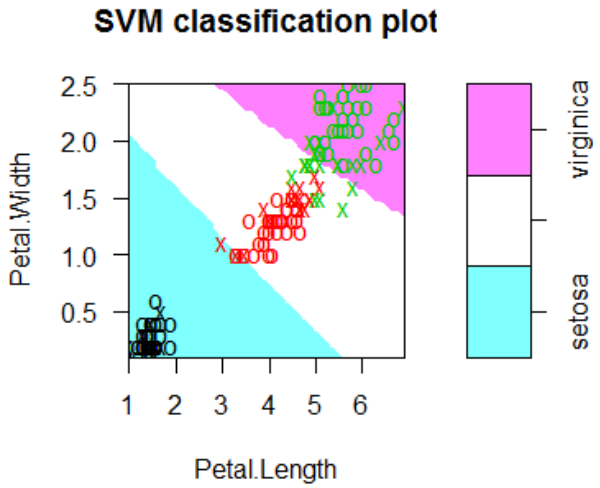

Figure 4.17 Support Vector Machine -RStudio

Write down the R language code for Support Vector Machine

4.2.5 Unsupervised Learning

Unsupervised Learning is a paradigm in machine learning which will try to infer from the given datasets consisting the input data without labelled response. In general, the algorithm will be provided with the input features of the problem, but the system is not taught with what is the output for corresponding input. The algorithm will try to develop inference by figuring out clusters of data points with similar patterns. This area of analysis if called exploratory data analytics where we are trying to fetch out hidden patterns in the data and group them.

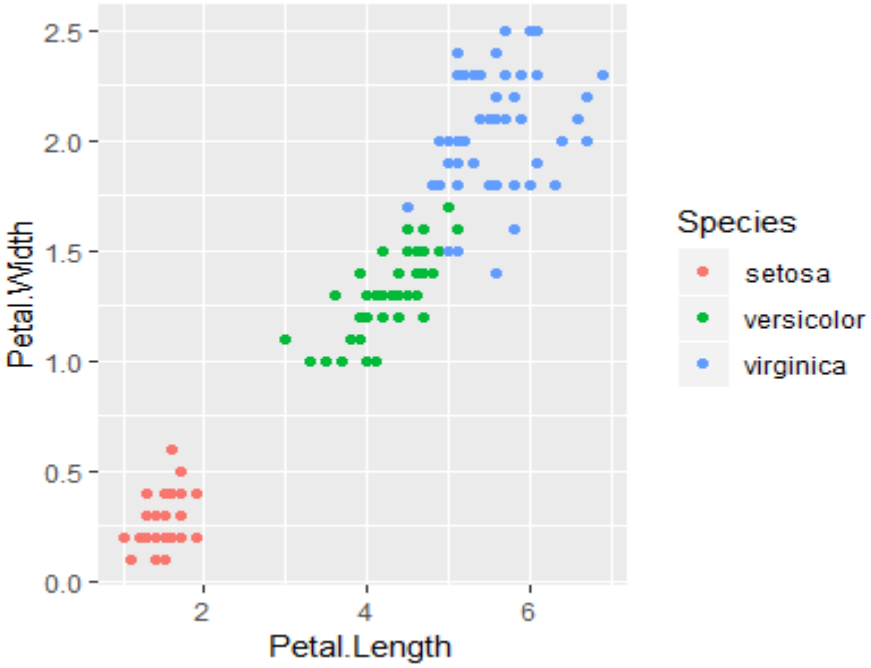

Figure 4.18 IRIS data

4.2.5.1 K-Means Clustering

Clustering is the act of identifying groups of data based on their features and forming clusters. In clustering, we try to group data of similar qualities. So clustering algorithm is classified under unsupervised learning. In unsupervised learning, the labels for each data is missing. So, the only way the algorithm can figure out the labels is by organising the data into individual groups whose attributes are similar. For example, let us consider the IRIS dataset. The IRIS dataset consists of four features of a flower, which are:

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width

The dataset for supervised learning looks as follows:

Table 2 Labelled Dataset

Sepal. Length	Sepal. Width	Petal. Length	Petal. Width	Species
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.4	Setosa
5.4	3.9	1.7	0.4	Setosa

In the Table.2, we can see that the input features are mentioned and the label to the features is also mentioned. So, this is a clear case of Supervised Learning. Now let us observe the data for Unsupervised Learning

Table 3 Unlabelled Data

Sepal. Length	Sepal. Width	Petal. Length	Petal. Width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.4
5.4	3.9	1.7	0.4

In this table, we can observe that the label to the input features are missing, so the learning algorithm cannot learn by any training data or labelled data, so it starts grouping the data with similar instances

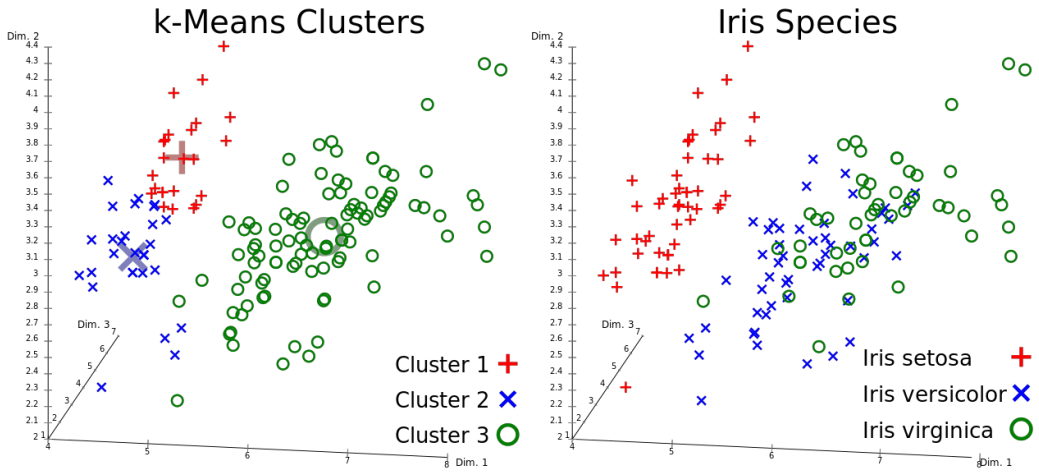

Figure 4.19 Unsupervised (Left) Vs Supervised (Right)

k-Means algorithm is one of the simplest algorithms than can work effectively in Unsupervised Learning. The formation of optimum clusters is done within few iterations. So, in K-Means algorithm we start by assigning random number of clusters (k= 10) and start adjusting the centroids in the data. As the algorithm proceeds, the centroid positions and clusters are updated.

Steps to perform K-Means clustering in R Programming

- Setting working directory
- Importing dataset
- Import ggplot2, ggfortify, stats, DPLYR packages
- Scale the data
- Fit the dataset into K-Means function, we need to specify the number of clusters
- Generate the confusion matrix
- Plot the data to find the Voronoi diagram

Write down the R language code for K-Means Clustering

5 Assessments

5.1 Assessment 1

1. What is R-Language?
   - ○ Statistical programming language
   - ○ Artificial intelligence programming language
   - ○ C programming language
   - ○ Machine language programming

2. The outcome of Regression is_____?
   - ○ Discrete
   - ○ Continuous
   - ○ Can be discrete or continuous depending on the dataset
   - ○ Exponential

3. Which of the following is a classifier?
   - ○ Logistic regression
   - ○ Multivariant Regression
   - ○ Linear Regression
   - ○ Polynomial Regression

4. Which of the following can be used for both prediction and classification?
   - ○ Logistic regression
   - ○ Support Vector Machine
   - ○ Polynomial Regression
   - ○ Multivariant Regression

5. Which of the following has a dichotomous predictor variable?
   - ○ Support Vector machine
   - ○ Random Forest
   - ○ Logistic Regression
   - ○ Perceptron

6. What is a slope?
   - ○ Change in Y / Change in X
   - ○ Change in X * Change in Y
   - ○ Change in Y * Change in X
   - ○ Change in X / change in Y

7. Function for scatter plot?
   - ○ plot ()
   - ○ scatterplot ()
   - ○ line ()
   - ○ scatter ()

8. Which of the following has a better accuracy percentage?
   - ○ Linear regression
   - ○ Polynomial Regression
   - ○ Multivariant Regression
   - ○ Polynomial Regression and Multivariant regression have the same accuracy %

9. How to execute the code line by line in R Studio?

- ○ Select each line and select 'RUN'
- ○ Ctrl + Alt +S
- ○ Ctrl+Alt+L
- ○ Ctrl+Shift +S

10. Which programming language is used in R Studio?
    - ○ Python
    - ○ C++
    - ○ LUA
    - ○ R Language

11. What is the purpose of set.seed(123)?
    - ○ Selecting the first three datapoints of the dataset
    - ○ To generate random numbers to select datapoints
    - ○ To select the datapoints in order
    - ○ None of the above

12. Which of the following library is used for performing SVM in R Studio?
    - ○ GGally
    - ○ caTools
    - ○ e17041
    - ○ e1071

13. What is the purpose of a test set?
    - ○ To validate and train the system
    - ○ To validate the trained system
    - ○ To train the training set
    - ○ To train the system

14. What is the function of factor()?
    - ○ To perform regression
    - ○ To perform classification
    - ○ To convert the levels into labels
    - ○ To convert the labels into levels

15. How do you access the column 'Matches' from a dataset 'Cricket' after importing it into R Studio?
    - ○ Matches$Cricket
    - ○ Cricket$Matches
    - ○ Cricket%Matches
    - ○ Matches%Cricket

16. How to print in R?
    - ○ scanf ()
    - ○ printf ()
    - ○ print ()
    - ○ View ()

17. When is data pre-processing required?
    - ○ After the result
    - ○ After result and before viewing
    - ○ Before importing the data
    - ○ After importing the data

18. Which of the following uses binomial function to perform classification?
    - ○ Support Vector Machine
    - ○ Logistic Regression
    - ○ Polynomial Regression
    - ○ Random Forest

19. How many independent variables does a Linear regression algorithm require?
    - ○ 1
    - ○ 2
    - ○ 3
    - ○ 4

20. What does lm stand for in reg=lm (formula = dependentvariable~. , data=training_set)
    - ○ Linear Model
    - ○ Linear Regression Model
    - ○ Linearized Model
    - ○ Least Model

5.2 Assessment 2

1. How to plot a scatter graph?
   - ○ scatter ()
   - ○ plot ()
   - ○ points ()
   - ○ sct ()

2. How to print in R?
   - ○ scanf ()
   - ○ printf ()
   - ○ print ()
   - ○ View ()

3. Considering two matrices m1 and m2. Choose the correct syntax for matrix multiplication?
   - ○ m1*m2
   - ○ m1*%*m2
   - ○ m1%*%m2
   - ○ m1*%%m2

4. Under which learning does prediction comes?
   - ○ Supervised learning
   - ○ Unsupervised learning
   - ○ Reinforcement learning
   - ○ Self-learning

5. What does lines () syntax do?
   - ○ Draws line in graph
   - ○ Draws the axis line in graph
   - ○ Plot some random line
   - ○ Draws additional line in an existing graph

6. In which learning does clustering comes under?
   - ○ Supervised learning
   - ○ Unsupervised learning
   - ○ Reinforcement learning
   - ○ Self-learning

7. What is the use of creating a function?
   - ○ To call and use it whenever needed
   - ○ To perform the operation with various inputs
   - ○ Makes the code simpler
   - ○ All the above

8. What are the things needed for creating a function?
   - ○ Variables
   - ○ Logic
   - ○ Variables and graph
   - ○ Variables and logic

9. What is the range for split ratio?
   - ○ 0.1-0.9
   - ○ 0.01-0.9
   - ○ 0.1-0.99
   - ○ 0.01-0.99

10. What is the use of data.frame?
   - ○ To log the answers
   - ○ To create the in-between values as table
   - ○ To create data in R
   - ○ To create table of all values used

11. How to represent a character?
   - ○ a= TRUE
   - ○ a="TRUE";
   - ○ a="TRUE"
   - ○ a="TRUE":

12. What is the use of class ()?
   - ○ To segregate into classes
   - ○ To know which data type they belong to
   - ○ To make it as class A & B
   - ○ None of the above

13. What is the use of training set?
   - ○ One part of the split for graph
   - ○ Train the result
   - ○ Testing purpose
   - ○ Give experience to the system

14. What is the inbuilt function for multiplication of two numbers?
   - ○ a= 2*3
   - ○ a= mult (2,3)
   - ○ a= prod (2,3)
   - ○ a= multiply (2,3)

15. test= subset (book1, split==TRUE) What is the use of subset()?
   - ○ Subtract the values if TRUE
   - ○ Select the FALSE values from split and give it to test
   - ○ Subtract the values if not TRUE
   - ○ Select the TRUE values from split and give it to test

16. m1= matrix(c (1,2,3,4),_____) Fill in the blank
   - ○ nrow
   - ○ ncol
   - ○ Either nrow or ncol
   - ○ Both

17. a= c (1:20) What does c(1:20) indicate?
   - ○ Generates continuous even numbers within the range
   - ○ Generates continuous odd numbers within the range

- Generates continuous numbers within the range
- Generates continuous negative numbers of the range

18. What is the function to do square root of a number?

- square.root(25)
- sqrt (25)
- squarert (25)
- sroot(25)

19. What is the syntax for performing categorical data?

- category ()
- fun ()
- factor ()
- is.na ()

20. What is the library to enable syntax for split data?

- Ggplot
- sample.split
- subset
- caTools

5.3 Assessment 3

1. What is the code for generating random number?
   - ○ randomnumber(123)
   - ○ rm (234)
   - ○ set.seed(456)
   - ○ generate (678)

2. What does T, P, E in machine learning mean?
   - ○ Test, Predict, Expose
   - ○ Train, Predict, Experience
   - ○ Task, Performance, Experience
   - ○ Task, Predict, Experience

3. What does lm () in regression code mean?
   - ○ Linear mode
   - ○ Line modelling
   - ○ Line makes
   - ○ Linear modelling

4. From which window in RStudio you find the option "Import dataset"?
   - ○ Console window
   - ○ Script window
   - ○ Environment window
   - ○ Files window

5. What is the use of setting working directory?
   - ○ Making a folder as working area
   - ○ Making the folder to save data
   - ○ Making the folder to work only in RStudio
   - ○ None of the above

6. What are the equations required for linear regression solving?
   - ○ Equation of line
   - ○ Slope equation
   - ○ Y-intercept
   - ○ All the above

7. What is the syntax for finding determinant of a matrix?
   - ○ solve ()
   - ○ det ()
   - ○ determinant ()
   - ○ subset ()

8. What is the syntax for finding inverse of a matrix?
   - ○ solve ()
   - ○ det ()
   - ○ determinant ()
   - ○ subset ()

9. How many ways are there in R language to assign a value or character?

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○

10. What is the syntax for adding additional lines to an existing graph?

- ○ plot ()
- ○ lines ()
- ○ pie ()
- ○ hist ()

11. What is the syntax for subtraction?

- ○ sub ()
- ○ diff ()
- ○ difference ()
- ○ No syntax

12. What is the output of a linear regression?

- ○ Sine wave
- ○ Straight line
- ○ Rectangular
- ○ Scatter

13. Reg = lm(book$price, data= training) What is the use of **"$"** in the code?

- ○ Shows the values
- ○ Selects the parameter on its own
- ○ Shows all attributes present in the data sheet
- ○ None of the above

14. A= **c** (1,2,3,4,5) What does "c" mean?

- ○ Combine
- ○ Complex
- ○ Corresponding
- ○ Concatenate

15. What is regression?

- ○ Relation between two variables
- ○ Straight line outcome
- ○ Statistical relation between variables
- ○ Statistical relation between only two variables.

16. Who gave the machine learning definition with T, P and E?

- ○ Tom Mitchell
- ○ Mitchell Sam
- ○ Tom cruise
- ○ Mitchell McClenaghan

17. How to represent a vector in R?

- ○ a= v (1,2,3,4,5)
- ○ a= c (1,2,3,4,5)

- a= b (1,2,3,4,5)
- a= a (1,2,3,4,5)

18. What value is filled when you perform the task of missing data?

- First value
- Last value
- middle value
- Average value

19. What does ctrl+L do?

- Execute the code
- Clears console
- Does nothing
- Clears code

20. What is the syntax for creating a matrix?

- mat ()
- mt ()
- matrix ()
- matx ()

5.4 Assessment 4

1. The sigmoid function ranges between_____to _____
   - ○ 0 to 1
   - ○ negative infinity to positive infinity
   - ○ 0 to 0.5
   - ○ 1 to 100
2. In machine learning when does a system is said to be learning?
   - ○ When task completes with experience
   - ○ When performance reduces with experience
   - ○ When performance increases with experience
   - ○ When experience reduces with task
3. Let's consider a dependant variable 'a' and an independent variable 'b'. Select the correct option for representing it in the linear regression syntax?
   - ○ reg= lm (formula= a-b, SplitRatio= 0.8)
   - ○ reg= lm (formula= b~a, SplitRatio= 0.8)
   - ○ reg= lm (formula= b-a, SplitRatio= 0.8)
   - ○ reg= lm (formula= a~b, SplitRatio= 0.8)
4. Deep learning is a subset of which field?
   - ○ Artificial Intelligence
   - ○ Machine learning
   - ○ Tensor flow
   - ○ Deep Learning
5. Given any Programming Language, what is the first step to be done to perform machine learning for a given dataset?
   - ○ Import the Dataset into software
   - ○ Analyse the dataset and perform Data Pre-processing techniques
   - ○ Import the library caTools
   - ○ Split the dataset
6. Fill in the blank. Y_pred= predict(regressor,_____)
   - ○ newdata
   - ○ regression variable
   - ○ data
   - ○ graph
7. What is the result of split while executing data split code?
   - ○ Split as 0s and 1s
   - ○ Split as TRUE and FALSE
   - ○ Split as A and B
   - ○ Doesn't split
8. In training_set[,2:5]= scale(training_set[,2:5]), what is the function of scale?

- ○ Scales down the dataset to the give ratio
- ○ Selects all the rows and columns 2 to 5
- ○ Selects all the Columns and 2 to 5 rows
- ○ None of the above

9. What is the graph outcome for logistic regression?
   - ○ Straight line
   - ○ Hyperbola
   - ○ S-Curve
   - ○ U-Curve
10. Expand glm?
    - ○ General Line Mode
    - ○ Generalized Linear Model
    - ○ Generalized Linear Motor
    - ○ General Lacking Mode
11. How many windows are present in RStudio?
    - ○ 3
    - ○ 4
    - ○ 5
    - ○ 11
12. How many types of learning are present in machine learning?
    - ○ 4
    - ○ 2
    - ○ 3
    - ○ 5
13. Censes$population= ifelse(is.na(censes$population), ave(censes$population, FUN = function(x) mean(x, na. rm = TRUE)), censes$population) In the given code above what does “is.na” mean?
    - ○ is not applicable
    - ○ is not assigned
    - ○ is not attractive
    - ○ is not available
14. Censes$population= ifelse(is.na(censes$population), ave(censes$population, FUN = function(x) mean(x, na. rm = TRUE)), censes$population) In the given code what does “na.rm” mean?
    - ○ Not applicable remaining
    - ○ Not available remaining
    - ○ Not available remainder
    - ○ Not assigned remainder
15. What is R-Language?

- ○ Statistical programming language
- ○ Artificial intelligence programming language
- ○ C programming language
- ○ Machine language programming

16. What does ctrl+shift+s do?
    - ○ Execute the code
    - ○ Clears console
    - ○ Does nothing
    - ○ Clears code
17. What is the minimum split ration to be maintained?
    - ○ 70%
    - ○ 80%
    - ○ 75%
    - ○ 50%
18. What is the use of categorizing data?
    - ○ Categorize numbers into alphabets
    - ○ Categorize the data in ascending order
    - ○ Categorize the numbers in order
    - ○ Categorize alphabets to numbers
19. What is the result of split while executing data split code?
    - ○ Split as 0s and 1s
    - ○ Split as TRUE and FALSE
    - ○ Split as A and B
    - ○ Doesn't split
20. How does reinforcement learning works?
    - ○ Providing both input and output
    - ○ Providing only input
    - ○ Reward and Penalty system
    - ○ Doesn't work

5.5 Assessment 5

1. Which of the following follows Least square errors as a cost function?
   - ○ Support Vector machine
   - ○ Linear Regression
   - ○ Random Forest
   - ○ Logistic Regression

2. In split= sample.split(Iris$Species, SplitRatio = 0.75), 75% of the dataset is _____
   - ○ Actual dataset
   - ○ Training set
   - ○ Test set
   - ○ Both training and test set

3. What is likely to happen when the following code is executed multiple times?
   Automobile$Model=factor (Automobile$Model,levels=c('BMW','Nissan'),labels=c(1,0))
   - ○ BMW and Nissan will remain the same
   - ○ BMW and Nissan will be replaced with 1 and 0 respectively
   - ○ BMW and Nissan will be replaced with 0 and 1 respectively
   - ○ BMW and Nissan will be replaced with NA

4. What is the syntax for prediction?
   - ○ y_pred()
   - ○ predict ()
   - ○ y.pred()
   - ○ ypredict ()

5. What is the range for split ratio?
   - ○ 0.1-0.9
   - ○ 0.01-0.9
   - ○ 0.1-0.99
   - ○ 0.01-0.99

6. plot(x,y,type='o',col='blue') What does type='o' mean?
   - ○ Plot points
   - ○ Plot lines
   - ○ Plot points and lines
   - ○ Color for line

7. What dataset you use when predicting?
   - ○ training set
   - ○ full data
   - ○ missing data
   - ○ test set

8. Who gave the machine learning definition with T, P and E?
   - ○ Tom Mitchell
   - ○ Mitchell Sam
   - ○ Tom cruise
   - ○ Mitchell McClenaghan

9. What is the function of a hyperplane?
   - ○ Classifies the datapoints
   - ○ Misclassifies the datapoints

- ○ classifies only the required data
- ○ does nothing

10. Logistic regression comes under which type of learning?
    - ○ Reinforcement learning
    - ○ Supervised Learning
    - ○ Unsupervised Learning
    - ○ None of the above

11. The outcome of Classification is_____?
    - ○ Discrete
    - ○ Continuous
    - ○ Can be discrete or continuous depending on the dataset
    - ○ Exponential

12. What does ctrl+L do?
    - ○ Execute the code
    - ○ Clears console
    - ○ Does nothing
    - ○ Clears code

13. How does Unsupervised learning work?
    - ○ Providing both input and output
    - ○ Providing only input
    - ○ Reward and Penalty system
    - ○ Doesn't work

14. What is a hidden layer in Artificial Neural network?
    - ○ Transforms the value from the input node to output node
    - ○ Transforms the value from the input node to output node adding some random weight
    - ○ Transforms the value from the output node to input node
    - ○ None of the above

15. Which of the following is not a search algorithm?
    - ○ Linear Search Algorithm
    - ○ Binary Search Algorithm
    - ○ Interpolation Search
    - ○ Convolutional Search

16. Which search algorithm works on the principle of divide and conquer?
    - ○ Binary Search Algorithm
    - ○ Jump Search
    - ○ Linear Search
    - ○ Convolutional Search

17. Which algorithm can be used to identify if the input image is a Human being or not a human being?
    - ○ Support Vector machine
    - ○ Logistic Regression
    - ○ Random Forest
    - ○ Multivariant Regression

18. Which of the following is not a subset of Neural network?
    - ○ Generative Adversarial Network
    - ○ Convolutional Neural Network

- ○ Recurrent Neural Network
- ○ Productive Adversarial Network

19. Debug the following code. training_set= subset (Iris, split=True)
    test_set= subset(Iris, split=False)
    - ○ training_set= subset (Iris, split==TRUE)
      test_set= subset(Iris, split==FALSE)
    - ○ training_set= subset (Iris, split=TRUE)
      test_set= subset(Iris, split=FALSE)
    - ○ training_set= set (Iris, split==TRUE)
      test_set= set(Iris, split==FALSE)
    - ○ No Error

20. What is the use of categorizing data?
    - ○ Categorize numbers into alphabets
    - ○ Categorize the data in ascending order
    - ○ Categorize the numbers in order
    - ○ Categorize alphabets to numbers

6 References

[1] Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of Statistical Learning - Data Mining, Inference, and Prediction. Berlin: Springer-Verlag.

[2] Mitchell, T. (1997). Machine Learning. New York: Mc Graw-Hill.

[3] Russel, S. and Norvig, P. (2003). Artifiical Intelligence: A Modern Approach. 2nd Edition. New York: Prentice-Hall

[4] Tan, P-N., Steinbach, M., and Kumar, V. (2004). Introduction to Data Mining. New York: Addison-Vesley