Introduction to Analytics
(Associate Analytics – I)

Syllabus

Unit I: Introduction to Analytics and R programming (NOS 2101)

Introduction to R, R Studio (GUI): R Windows Environment, introduction to various data types, Numeric, Character, date, data frame, array, matrix etc., Reading Datasets, Working with different file types .txt,.csv etc. Outliers, Combining Datasets in R, Functions and loops.

Manage your work to meet requirements (NOS 9001)

Understanding Learning objectives, Introduction to work & meeting requirements, Time Management, Work management & prioritization, Quality & Standards Adherence,

Unit II: Summarizing Data & Revisiting Probability (NOS 2101)

Summary Statistics - Summarizing data with R, Probability, Expected, Random, Bivariate Random variables, Probability distribution. Central Limit Theorem etc.

Work effectively with Colleagues (NOS 9002)

Introduction to work effectively, Team Work, Professionalism, Effective Communication skills, etc.

Unit III: SQL using R

Introduction to NoSQL, Connecting R to NoSQL databases. Excel and R integration with R connector.

Unit IV: Correlation and Regression Analysis (NOS 9001)

Regression Analysis, Assumptions of OLS Regression, Regression Modelling. Correlation, ANOVA, Forecasting, Heteroscedasticity, Autocorrelation, Introduction to Multiple Regression etc.

Unit V: Understand the Verticals - Engineering, Financial and others (NOS 9002)

Understanding systems viz. Engineering Design, Manufacturing, Smart Utilities, Production lines, Automotive, Technology etc.

Understanding Business problems related to various businesses

Requirements Gathering: Gathering all the data related to Business objective

Reference Books:

1. **Introduction to Probability and Statistics Using R**, ISBN: 978-0-557-24979-4, is a textbook written for an undergraduate course in probability and statistics.
2. **An Introduction to R**, by Venables and Smith and the R Development Core Team. http://www.r-project.org/, see Manuals.
3. Montgomery, Douglas C., and George C. Runger, **Applied statistics and probability for engineers.** John Wiley & Sons, 2010.
4. The ***Basic Concepts of Time Series Analysis***. http://anson.ucdavis.edu/~azari/sta137/AuNotes.pdf
5. **Time Series Analysis and Mining with R,**Yanchang Zhao.

UNIT I

UNIT-I: Introduction to Analytics and R programming (NOS 2101)
Introduction to R, R Studio (GUI): R Windows Environment, introduction to various data types, Numeric, Character, date, data frame, array, matrix etc., Reading Datasets, Working with different file types .txt,.csv etc. Outliers, Combining Datasets in R, Functions and loops.

Manage your work to meet requirements (NOS 9001)

Understanding Learning objectives, Introduction to work & meeting requirements, Time Management, Work management & prioritization, Quality & Standards Adherence.

UNIT-I: Introduction to Analytics and R programming (NOS 2101)

Topic No.	Name of the Topic
1.1	Introduction to R
1.2	R Studio (GUI):
1.3	Introduction to various data types: Numeric, Character ,date, data frame, array, matrix etc.,
1.4	Reading Datasets and Working with different file types . txt,.csv etc.
1.5	Outliers
1.6	Combining Datasets in R
1.7	Functions and Loops
2.10	**Provide Data/Information in Standard formats (NOS 9004)**

1.1. Introduction to R

What is R?

R is a flexible and powerful open-source implementation of the language S (for statistics) developed by John Chambers and others at Bell Labs.

Why R?

Five reasons to learn and use R:

- ✓ R is open source and completely free. R community members regularly contribute packages to increase R's functionality.
- ✓ R is as good as commercially available statistical packages like SPSS, SAS, and Minitab.
- ✓ R has extensive statistical and graphing capabilities. R provides hundreds of built-in statistical functions as well as its own built-in programming language.
- ✓ R is used in teaching and performing computational statistics. It is the language of choice for many academics who teach computational statistics.
- ✓ Getting help from the R user community is easy. There are readily available online tutorials, data sets, and discussion forums about R.

R uses:

- ✓ R combines aspects of functional and object-oriented programming.
- ✓ R can use in interactive mode
- ✓ It is an interpreted language rather than a compiled one.
- ✓ Finding and fixing mistakes is typically much easier in R than in many other languages.

R Features:-

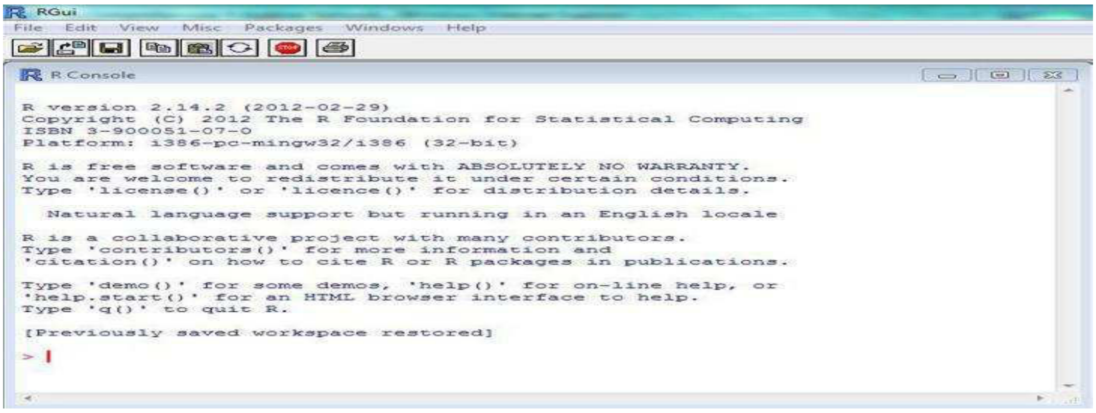
- ✓ Programming language for graphics and statistical computations
- ✓ Available freely under the GNU public license

- ✓ Used in data mining and statistical analysis
- ✓ Included time series analysis, linear and nonlinear modeling among others
- ✓ Very active community and package contributions
- ✓ Very little programming language knowledge necessary
- ✓ Can be downloaded from http://www.r-project.org/ **opensource**

What is CRAN?

CRAN abbreviates **Comprehensive R Archive Network** will provide binary files and follow the installation instructions and accepting all defaults. Download from *http://cran.r-project.org/* we can see the R Console window will be in the RGui (graphical user interface). Fig 1 is the sample

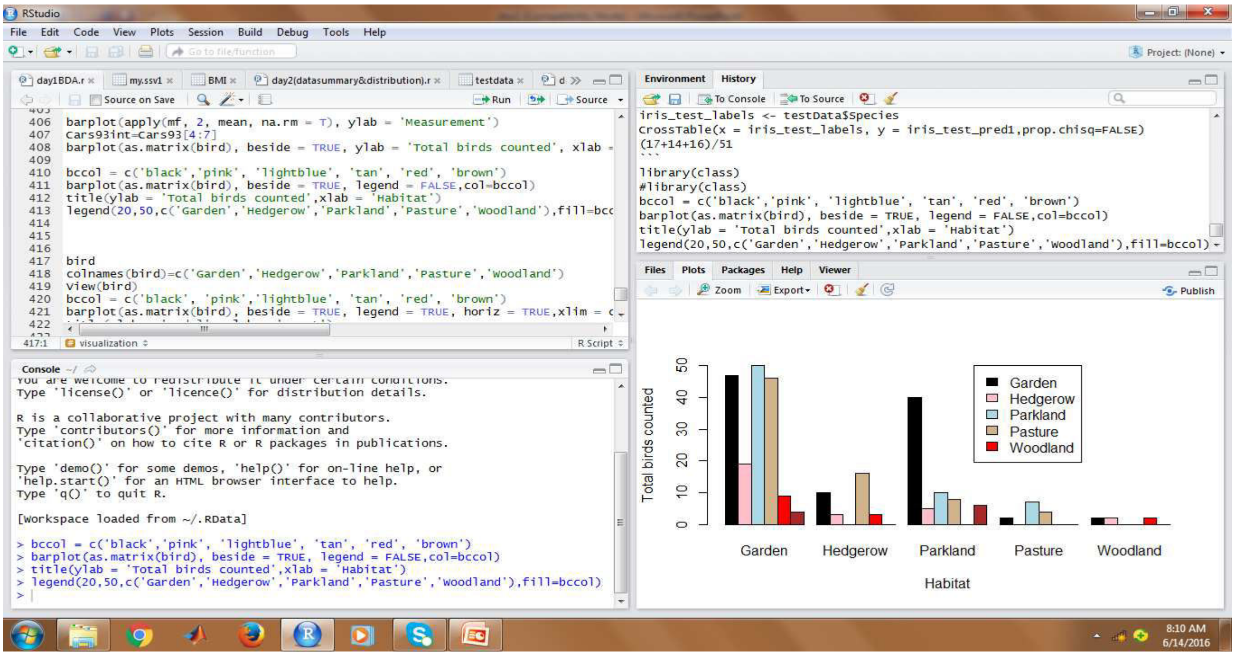
R GUI.



```
R version 2.14.2 (2012-02-29)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

Figure 1. R console

R Studio: R Studio is an Integrated Development Environment (IDE) for R Language with advanced and more user-friendly GUI. R Studio allows the user to run R in a more user-friendly environment. It is open-source (i.e. free) and available at **http://www.rstudio.com/**.



```
barplot(apply(mf, 2, mean, na.rm = T), ylab = 'Measurement')
Cars93int=Cars93[4:7]
barplot(as.matrix(bird), beside = TRUE, ylab = 'Total birds counted', xlab =

bccol = c('black','pink', 'lightblue', 'tan', 'red', 'brown')
barplot(as.matrix(bird), beside = TRUE, legend = FALSE,col=bccol)
title(ylab = 'Total birds counted',xlab = 'Habitat')
legend(20,50,c('Garden','Hedgerow','Parkland','Pasture','Woodland'),fill=bcc



bird
colnames(bird)=c('Garden','Hedgerow','Parkland','Pasture','Woodland')
View(bird)
bccol = c('black', 'pink','lightblue', 'tan', 'red', 'brown')
barplot(as.matrix(bird), beside = TRUE, legend = TRUE, horiz = TRUE,xlim = c
```

```
iris_test_labels <- testData$Species
CrossTable(x = iris_test_labels, y = iris_test_pred1,prop.chisq=FALSE)
(17+14+16)/51
```

library(class)
#library(class)
bccol = c('black','pink', 'lightblue', 'tan', 'red', 'brown')
barplot(as.matrix(bird), beside = TRUE, legend = FALSE,col=bccol)
title(ylab = 'Total birds counted',xlab = 'Habitat')
legend(20,50,c('Garden','Hedgerow','Parkland','Pasture','Woodland'),fill=bccol)
```

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> bccol = c('black','pink', 'lightblue', 'tan', 'red', 'brown')
> barplot(as.matrix(bird), beside = TRUE, legend = FALSE,col=bccol)
> title(ylab = 'Total birds counted',xlab = 'Habitat')
> legend(20,50,c('Garden','Hedgerow','Parkland','Pasture','Woodland'),fill=bccol)
> 
```

Figure 2. R studio GUI

Figure 2 shows the GUI of R Studio.The R Studio screen has four windows:

1. Console.
2. Workspace and history.
3. Files, plots, packages and help.
```

4. The R script(s) and data view.

The R script is where you keep a record of your work.

Create a new R script file:

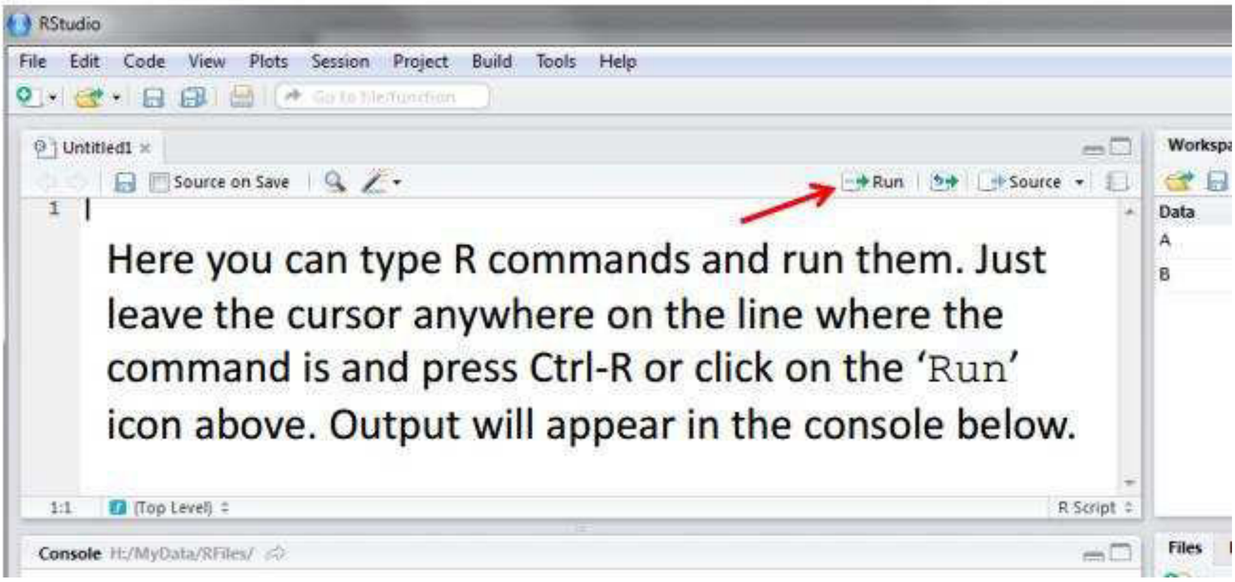
To create a new R script file:

1) File -> New -> R Script,

2) Click on the icon with the "+" sign and select "R Script"

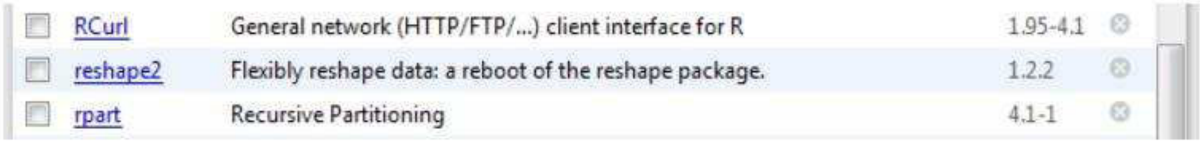
3) Use shortcut as: Ctrl+Shift+N.

Running the R commands on R Script file:



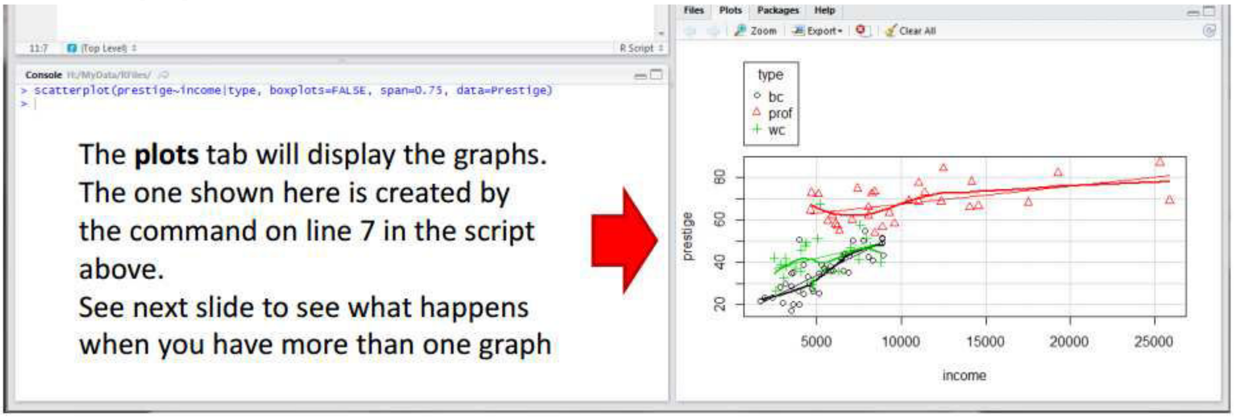
Here you can type R commands and run them. Just leave the cursor anywhere on the line where the command is and press Ctrl-R or click on the 'Run' icon above. Output will appear in the console below.

Installing Packages:



□	RCurl	General network (HTTP/FTP/...) client interface for R	1.95-4.1
□	reshape2	Flexibly reshape data: a reboot of the reshape package.	1.2.2
□	rpart	Recursive Partitioning	4.1-1

Plots to display:



```
> scatterplot(prestige~income|type, boxplots=FALSE, span=0.75, data=Prestige)
> |
```

The **plots** tab will display the graphs. The one shown here is created by the command on line 7 in the script above.

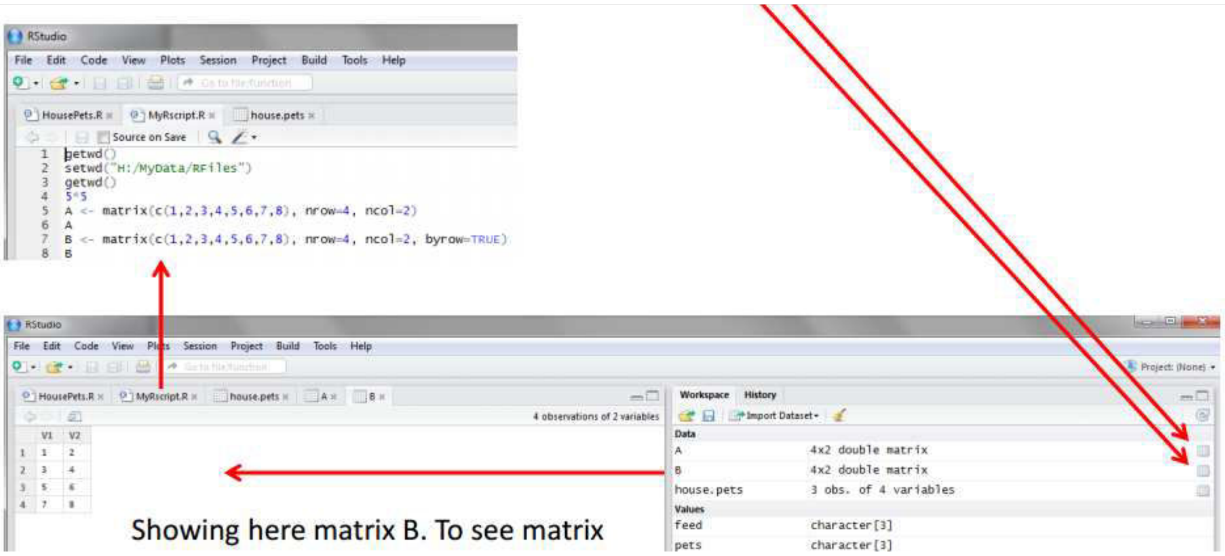
See next slide to see what happens when you have more than one graph

Console:

The console is where you can type commands and see output.

Workspace tab:

The workspace tab shows all the active objects (see next slide). The workspace tab stores any object, value, function or anything you create during your R session. In the example below, if you click on the dotted squares you can see the data on a screen to the left.



```
getwd()
setwd("H:/MyData/RFiles")
getwd()
5*5
A <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2)
A
B <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2, byrow=TRUE)
B
```

4 observations of 2 variables

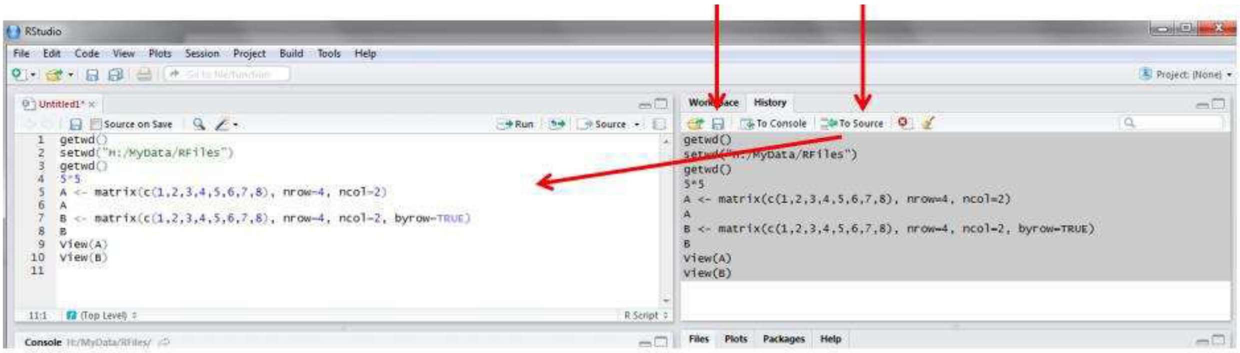
	V1	V2
1	1	2
2	3	4
3	5	6
4	7	8

```
Data
A              4x2 double matrix
B              4x2 double matrix
house.pets     3 obs. of 4 variables
Values
feed           character[3]
pets           character[3]
```

Showing here matrix B. To see matrix

History tab:

The history tab shows a list of commands used so far.The history tab keeps a record of all previous commands. It helps when testing and running processes. Here you can either save the whole list or you can select the commands you want and send them to an R script to keep track of your work. In this example, we select all and click on the "To Source" icon, a window on the left will open with the list of commands. Make sure to save the 'untitled1' file as an *.R script.



```
getwd()
setwd("H:/MyData/RFiles")
getwd()
5*5
A <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2)
A
B <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2, byrow=TRUE)
B
View(A)
View(B)
```

Files Tab:

The files tab shows all the files and folders in your default workspace as if you were on a PC/Mac window. The plots tab will show all your graphs. The packages tab will list a series of packages or add-ons needed to run certain processes.

Changing the working directory:

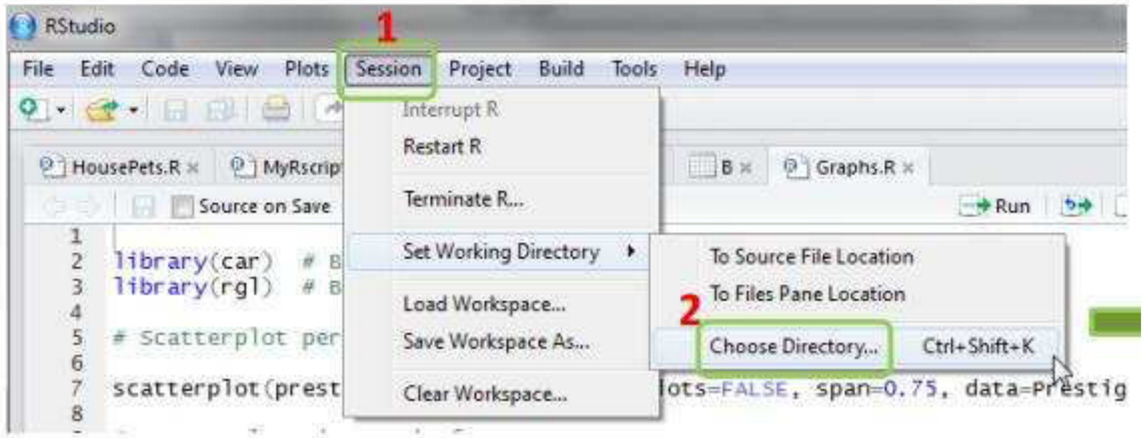
To Show the present working directory (wd)

>getwd()

C:/mydocuments #The default working directory is mydocuments

To change the working directory

>setwd("C:/myfolder/data")



```
library(car)   # B
library(rgl)   # B

# Scatterplot per

scatterplot(prest                      lots=FALSE, span=0.75, data=Prestig
```

First R program: Using R as calculator:

R commands can run in two ways:

1) Type at console and press enter to see the output. Output will get at console only in R studio.

2) Open new R Script file and write the command, keep the curser on the same line and press Ctrl+enter or click on Run. Then see the output at console along with command.

At console:

R as a calculator, typing commands directly into the R Console. Launch R and type the following code, pressing < Enter > after each command.

Type an *expression on console*.

R - Assignment Operators:

<- or = for assignment and == to test equality.

At the outer sider <- and = can be used similar. But we should be careful while using them in combined. In precise '<-' is prioritized than '=' . The operators <- and = assign into the environment in which they are evaluated. The operator <- can be used anywhere, whereas the operator = is only allowed at the top level (e.g., in the complete expression typed at the command prompt) or as one of the subexpressions in a braced list of expressions.

```
x <- y <- 5
x = y = 5
x = y <- 5
x <- y = 5
# Error in (x <- y) = 5 : could not find function "<-<-"
```

Example Exercise1:

```
> 2 * 2   ## Multiplication
[1] 4
> 2 / 2   ## Division
[1] 1
> 2 + 2   ## addition
[1] 4
> 2 – 2   ## subtraction
[1] 0
> 2 ^ 2   ## exponentiation
[1] 4
```

```
> q()      ## to quit
> y <- 3*exp(x)
> x <- 3*exp(x)
```

R expression:

At '>' (R-prompt) type the R expression and press enter

R output:

R labels each output value with a number in square brackets. As far as R is concerned, an individual number is a one-element vector. The [1] is simply the index of the first element of the vector.

Activity1: Calculate the following using R:
1. Log of 2
2.23 X 32
3. e3

Variable (Object) Names:

Certain variable names are reserved for particular purposes. Some reserved symbols are: **c q t C D F I T**

```
### meaning of c q t C D F I T
?      ## to see help document
?c    ## c means Combine Values into a Vector or List
?q    ## q means Terminate an R Session
?t     ## t means Matrix Transpose
?C    ## C means sets contrast for a factor
?D    ##  D means Symbolic and Algorithmic Derivatives of Simple Expressions
?F    ## F means logical vector Character strings
```

c("T", "TRUE", "True", "true") are regarded as true, c("F", "FALSE", "False", "false") as false, and all others as NA.

```
>F     ##[1] FALSE
?I     ##Inhibit Interpretation/Conversion of Objects
```

Working on variables:

Operators in R:

Table - Arithmetic operators		**Table. Logical operators**	
Operator	**Description**	**Operator**	**Description**
+	Addition	**<**	less than
-	Subtraction	**<=**	less than or equal to
*****	Multiplication	**>**	greater than
/	Division	**>=**	greater than or equal to
^ or **	Exponentiation	**==**	exactly equal to
x %% y	modulus (x mod y) 5%%2 is 1	**!=**	not equal to
x %/% y	integer division 5%/%2 is 2	**!x**	Not x
		x | y	x OR y
		x & y	x AND y
		isTRUE(x)	test if X is TRUE

Getting Help in R:

>help(**function**), or use the

>?**keyword** checks in all packages--- function shortcut

>??**keyword** checks in content of all packages

Either of above all will open the R documentation.

Comment Notation in R:

# is used to comment a line in R script.

List of Objects:

To see a listing of the objects in your workspace, you can use the **ls()** function. To get more detail, use **ls.str()**

```
> ls()
[1] "A" "acctdata" "address" "B" "b1"
[6] "balance" "c" "CareerSat" "chisquare" "colnames"
```

1.3. Introduction to various Data Types:

In Analytics the data is classified as Quantitative(numeric) and Qualitative(Character/Factor) on very broad level.

- Numeric Data: - It includes 0~9, "." and "- ve" sign.
- Character Data: - Everything except Numeric data type is Character.
  For Example, Names, Gender etc.

For Example, "1,2,3…" are Quantitative Data while "Good", "Bad" etc. are Qualitative Data. We can convert Qualitative Data into Quantitative Data using Ordinal Values. For Example, "Good" can be rated as 9 while "Average" can be rated as 5 and "Bad" can be rated as 0.

Table 1. List of Data types in R

Data Type		Verify

Logical	TRUE , FALSE	`v <- TRUE`
		`print(class(v))`
		`[1]  "logical"`

Numeric	12.3, 5, 999	`v <- 23.5`
		`print(class(v))`
		`[1] "numeric"`

Integer	2L, 34L, 0L	`v <- 2L`
		`print(class(v))`
		`[1] "integer"`

Complex	3 + 2i	`v <- 2+5i`
		`print(class(v))`
		`[1] "complex"`

Character	'a' , '"good", "TRUE", '23.4'	`v <- "TRUE"`
		`print(class(v))`
		`[1] "character"`

Raw	"Hello" is stored as 48 65 6c 6c 6f	`v <- charToRaw("Hello")`
		`print(class(v))`
		`[1] "raw"`

mode() or Class():

These are used to know the type of data object type assigned.

Example:

Assign several different objects to x, and check the mode (storage class) of each object.

# Declare variables of different types:

```
my_numeric <- 42
my_character <- "forty-two"
my_logical <- FALSE
```

# Check which type these variables have:

```
>class(my_numeric)
[1] "numeric"
> class(my_character)
[1] "character"
> class(my_logical)
[1] "logical"
```

Mode vs Class:

'mode' is a mutually exclusive classification of objects according to their basic structure. The 'atomic' modes are numeric, complex, charcter and logical. Recursive objects have modes such as 'list' or 'function' or a few others. An object has one and only one mode.

'class' is a property assigned to an object that determines how generic functions operate with it. It is not a mutually exclusive classification. If an object has no specific class assigned to it, such as a simple numeric vector, it's class is usually the same as its mode, by convention.

Changing the mode of an object is often called 'coercion'. The mode of an object can change without necessarily changing the class.

e.g.

```
> x <- 1:16
> mode(x)
[1] "numeric"
> dim(x) <- c(4,4)
> mode(x)
[1] "numeric"
> class(x)
[1] "matrix"
> is.numeric(x)
[1] TRUE
> mode(x) <- "character"
> mode(x)
[1] "character"
> class(x)
[1] "matrix"
```

```
However:
```

```
> x <- factor(x)
> class(x)
[1] "factor"
> mode(x)
[1] "numeric"
```

# Arithmetic operations on R objects

```
> x <- 2
> x
[1]  2
> x ^ x
```

```
[1]  4
> x ^ 2
[1]  4
> mode(x)    ## will returen the storage class of object
[1]  "numeric"
>  seq(1:10)   ## will create a vector of 1 to sequence numbers
[1] 1 2 3 4 5 6 7 8 9 10
>  x <- c(1:10)   ##vector of 1 to 10 digits
>  x
 [1]  1 2 3 4 5 6 7 8 9 10
> mode(x)
[1] "numeric"
> x <- c("Hello","world","!")
> mode(x)
[1] "character"
> x <- c(TRUE, TRUE, FALSE, FALSE, TRUE, FALSE, TRUE)
> mode(x)
[1] "logical"
> x <- list("R","12345",FALSE)
> x
[[1]]
[1] "R"
[[2]]
[1] "12345"
[[3]]
[1] FALSE
> mode(x)
[1] "list"
```

Create new variables using already available variables:

Example:

mydata$sum <- mydata$x1 + mydata$x2

New variable is created using two already available variables.

Modifying existing variable: Rename the existing variable by using *rename()* function.

For examples,

mydata<- rename(mydata, c(oldname="newname"))

1.3.1. Vectors:

Vector is the most common data structure in R. Vectors must be homogeneous i.e, the type of data in a given vector must all be the same. Vectors can be numeric, logical, or character. If a vector is mix data types then R forces (**coerces**, if you will) the data into one mode.

Creating a vector:

To create a vector , "concatenate" a list of numbers together to form a vector.

x < - c(1, 6, 4, 10, -2) ## c() to concatenate elements

my.vector<- 1:24 ## a numeric vector with 1 to 24 numbers

List of built-in functions to get useful summaries on vectors:

Example1:

> sum(x) ## sums the values in the vector

> length(x) ## produces the number of values in the vector, ie its length

> mean(x) ## the average (mean)

> var(x) ## the sample variance of the values in the vector

> sd(x) ## the sample standard deviation of the values in the vector (square root of the sample variance)

> max(x) ## the largest value in the vector

> min(x) ## the smallest number in the vector

```
> median(x) ## the sample median
> y < - sort(x) ## the values arranged in ascending order
```

Example2:

```
linkedin <- c(16, 9, 13, 5, 2, 17, 14)
> last <- tail(linkedin, 1)
> last
[1] 14
  > # Is last under 5 or above 10?
  > # Is last between 15 (exclusive) and 20 (inclusive)?
  > # Is last between 0 and 5 or between 10 and 15?
 > (last > 0 | last < 5)
[1] TRUE
> (last > 0 & last < 5)
[1] FALSE
> (last > 10 & last < 15)
[1] TRUE
```

Example3: Following are some other possibilities to create vectors

```
> x <- 1:10
> y <- seq(10) #Create a sequence
> z <- rep(1,10) #Create a repetitive pattern
> x
[1] 1 2 3 4 5 6 7 8 9 10
> y
[1] 1 2 3 4 5 6 7 8 9 10
> z
[1] 1 1 1 1 1 1 1 1 1 1
```

Adding elements to vector:

```
> x <- c(x, 11:15)
>x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Vector Arithmetic:

```
> x <- c(1:10)
> x
[1] 1 2 3 4 5 6 7 8 9 10
> y <- 10
> x + y
[1] 11 12 13 14 15 16 17 18 19 20
> 2 + 3 * x #Note the order of operations
[1] 5 8 11 14 17 20 23 26 29 32
> (2 + 3) * x #See the difference
[1] 5 10 15 20 25 30 35 40 45 50
> sqrt(x) #Square roots
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
[9] 3.000000 3.162278
> x %% 4 #This is the integer divide (modulo) operation
[1] 1 2 3 0 1 2 3 0 1 2
> y <- 3 + 2i #R does complex numbers
> re(y) #The real part of the complex number
[1] 3
> im(y) #The imaginary part of the complex number
[1] 2
> x * y
[1] 3+ 2i 6+ 4i 9+ 6i 12+ 8i 15 + 10i 18 + 12i 21 + 14i 24 + 16i 27 + 18i 30 + 20i
```

1.3.2. Matrices:

A matrix is a two-dimensional rectangular data set. Matrices must be heterogeneous i.e, the type of data in a given vector of all not in the same class.

Matrix can be created in three ways :

- matrix(): A vector input to the matrix function.
- Using rbind() and cbind() functions.
- Using dim() to the existing vector

Creating a matrix using matrix():

```
# Create a matrix.
M = matrix( c('a','a','b','c','b','a'), nrow=2,ncol=3,byrow = TRUE)
print(M)
     [,1] [,2] [,3]
[1,] "a"  "a"  "b"
[2,] "c"  "b"  "a"
```

Creating a matrix using rbind() or cbind():

First crate two vectors and then create a matrix using rbind() .It binds the two vectors data into two rows of matrix.

Example: To create a matrix havind the data as: 6,2 ,10 & 1, 3, -2

Step1: create two vectors as xr1,xr2

```
> xr1 <- c( 6, 2, 10)
> xr2 <- c(1, 3, -2)
> x <- rbind (xr1, xr2) ## binds the vectors into rows of a matrix(2X3)
> x
    [,1] [,2] [,3]
xr1    6    2   10
xr2    1    3   -2
```

First crate two vectors and then create a matrix using rbind() .It binds the two vectors data into two rows of matrix.

Example: To create a matrix havind the data as: 6,2 ,10 & 1, 3, -2

create two vectors as xr1,xr2

```
> y <- cbind(xr1, xr2) ## binds the same vectors into columns of a matrix(3X2)
> y
     xr1 xr2
[1,]   6   1
[2,]   2   3
[3,]  10  -2
```

Matrix operations:

```
> A <- matrix(c( 6, 1, 0, -3, -1, 2),3, 2, byrow = TRUE)
> B <- matrix(c( 4, 2, 0, 1, -5, -1),3, 2, byrow = TRUE)
>A (with output)
>B (with output)
> A + B
     [,1]  [,2]
[1,] 10    3
[2,]  0   -2
[3,] -6    1
> A - B
      [,1] [,2]
```

```
[1,]  2   -1
[2,]  0   -4
[3,]  4   3
> A * B # this is component-by-component multiplication, not matrix multiplication
     [,1] [,2]
[1,]  24  2
[2,]  0  -3
[3,]  5  -2
> t(A) ## Transpose of a matrix
[,1] [,2] [,3]
[1,] 6 0 -1
[2,] 1 -3 2
```

Alternative method to create a matrix using dim():

Create a vector and add the dimensions using the **dim ( )** function.It's especially useful if you have your data already in a vector.

Example: A vector with the numbers 1 through 24, like this:

```
>my.vector<- 1:24
```

You can easily convert that vector to an array exactly like my.array   simply by assigning the dimensions, like this:

```
> dim(my.vector) <- c(3,4)
```

1.3.3. Array:

Arrays can be of any number of dimensions. The array function takes a **dim** attribute which creates the required number of dimension. In the below example we create an array with two elements which are 3x3 matrices each. Creating an array:

```
>my.array< - array(1:24, dim=c(3,4,2))
```

In the above example, "my.array" is the name of the array we have given. There are 24 units in this array mentioned as "1:24" and are divided in three dimensions "(3, 4, 2)".

Alternative: with existing vector and using dim()

```
> my.vector<- 1:24
```

To convert my.vector vector to an array exactly like my.array simply by assigning the dimensions, like this:

```
> dim(my.vector) <- c(3,4,2)
```

Activity 2:Create an Array with name "MySales" with 30 observations using following methods:

1. Defining the dimensions of the array as 3, 5 and 2.
2. By using Vector method.

1.3.4. Lists:

A list is a R object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
>list1 <- list(c(2,5,3),21.3,sin)   # Create a list.
>print(list1)     # Print the list.
 [[1]]
[1] 2 5 3
```

```
[[2]]
[1] 21.3
[[3]]
function (x)  .Primitive("sin")
```

1.3.5. Data Frames:

Data frames are tabular data objects. Unlike a matrix in data frame each column can contain different modes of data. The first column can be numeric while the second column can be character and third column can be logical. It is a list of vectors of equal length. Data Frames are created using the data.frame() function. It displays data along with header information.

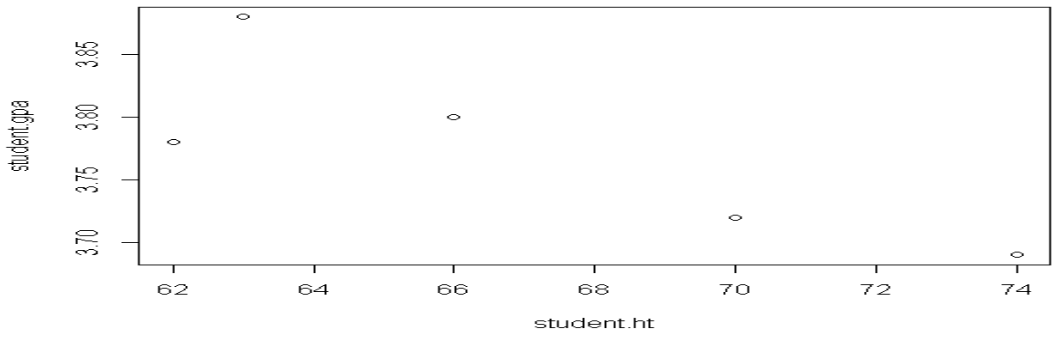
To retrieve data in a particular cell:

Enter its row and column coordinates in the single square bracket "[ ]" operator.

Example:

To retrieve the cell value from the first row, second column of mtcars.

```
>mtcars[1,2]
mtcars[row,column]
# Create the data frame.
>BMI <- data.frame(gender = c("Male", "Male","Female"), height = c(152, 171.5, 165), weight =
c(81,93, 78),Age =c(42,38,26))
>print(BMI)
      gender height weight Age
1   Male   152.0     81    42
2   Male   171.5     93    38
3  Female  165.0    78  26
Data1 :  Height GPA
 66        3.80
 62       3.78
 63       3.88
 70       3.72
 74       3.69
> student.ht <- c( 66, 62, 63, 70, 74)
> student.gpa < - c( 3.80, 3.78, 3.88, 3.72, 3.69)
> student.data1 < - data.frame(student.ht, student.gpa)
> student.data
      student.ht      student.gpa
1         66             3.80
2         62              3.78
3         63              3.88
4         70              3.72
5         74               3.69
> plot(student.ht, student.gpa)
```



1.3.6. Date:

Date() function is used to access the date and time in R

Sys.Date() : The current system date. This function returns a Date object.

class(Date)

A string in this format is treated as a character unless cast to a Date type.

>class("2010-06-16")

>class(as.Date("2010-06-16"))

You can also pass in dates in other formats and cast them as strings by specifying the format in use.

>as.Date('02/03/2004','%m/%d/%Y')

To format date information in a wide variety of string formats, use the strftime function.

>strftime(Sys.Date(),'%A: %B %d, %Y (Day %j of %Y)')

This returns the string "Tuesday: June 15, 2010 (Day 166 of 2010)"

Dates can be manipulated arithmetically.

Activity:

To return the next ten days…

seq(Sys.Date(),Sys.Date() + 10,1)

…or the last ten days…

seq(Sys.Date(),Sys.Date() – 10,-1)

1.4. Reading Datasets using R

We can import Datasets from various sources having various file types :

Example:

- **.csv or .txt format**
- **Big data tool – Impala**
- **CSV File**

The sample data can also be in comma separated values (CSV) format. Each cell inside such data file is separated by a special character, which usually is a comma, although other characters can be used as well. The first row of the data file should contain the column names instead of the actual data. Here is a sample of the expected format

```
Col1,Col2,Col3
100,a1,b1
200,a2,b2
300,a3,b3
```

After we copy and paste the data above in a file named "mydata.csv" with a text editor, we can read the data with the function read.csv.

In R data can read in two ways either from local disc or web.

From disc:

The data file location is known on local disc use: **read.csv() or read.table()** functions.

Path is not specific then use : **file.choose()**

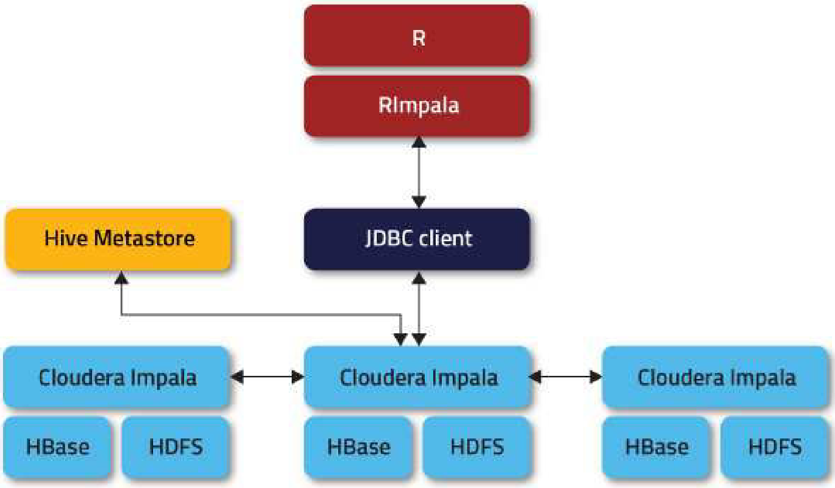
```
> mydata = read.csv("mydata.csv")  # read csv file
> mydata
```

From Web:

The URL of the data from web is pass to read.csv() or read.table() functions.

Big data tool – Impala:

- Cloudera 'Impala', which is a massively parallel processing (MPP) SQL query engine runs natively in Apache Hadoop.
- R package, RImpala, connects Impala to R.
- RImpala enables querying the data residing in HDFS and Apache HBase from R, which can be further processed as an R object using R functions.
- RImpala is now available for download from the Comprehensive R Archive Network (CRAN) under GNU General Public License (GPL3).
- This package is developed and maintained by MuSigma.



To install RImpala :

We use following code to install RImpala package.

```
>install. packages("RImpala")
```

Importing and Exporting CSV :

- Loading data – data(dataset_name*)*
- read and write functions
- getwd() and setwd(dir)
- read and write functions use full path name

Example:

read.csv (“C:/Rtutorials/Sampledata.csv”).

Writing dataset :write () function.

getwd() means get the working directory (wd) and
setwd() is used to set the working directory.

Activity 3:

Import a CSV file in R and check the output.

```
Name, Age, Sex
Shaan, 21, M
Ritu, 24, F
Raj, 31, M
```

1.5. Outliers:

- Outlier is a point or an observation that deviates significantly from the other observations.

Reasons for outliers: Due to experimental errors or “special circumstances”

Outlier detection tests to check for outliers.

There is no rigid mathematical definition of what constitutes an outlier; determining whether or not an observation is an outlier is ultimately a subjective exercise. There are various methods of outlier detection.[7][8][9][10] Some are graphical such as normal probability plots. Others are model-based. Box plots are a hybrid.

Outlier treatments are three types:

– **Retention :**

There is no rigid mathematical definition of what constitutes an outlier; determining whether or not an observation is an outlier is ultimately a subjective exercise. There are various methods of outlier detection.[7][8][9][10] Some are graphical such as normal probability plots. Others are model-based. Box plots are a hybrid.

- **Exclusion :**

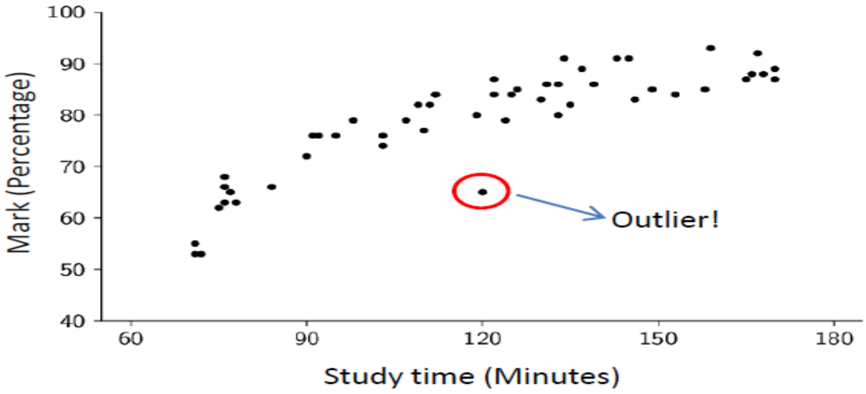
  Deletion of outlier data is a controversial practice frowned upon by many scientists and science instructors; while mathematical criteria provide an objective and quantitative method for data rejection, they do not make the practice more scientifically or methodologically sound, especially in small sets or where a normal distribution cannot be assumed. Rejection of outliers is more acceptable in areas of practice where the underlying model of the process being measured and the usual distribution of measurement error are confidently known. An outlier resulting from an instrument reading error may be excluded but it is desirable that the reading is at least verified.

- **Other treatment methods**

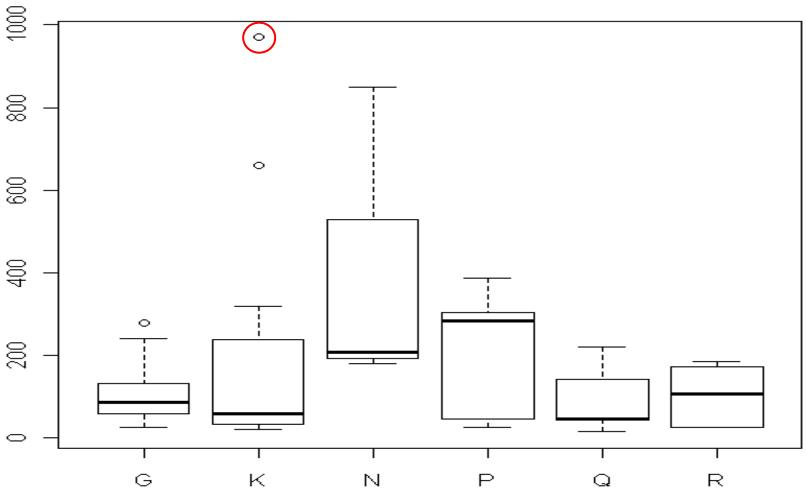
  OUTLIER package in R: to detect and treat outliers in Data.

  Outlier detection from graphical representation:
- Scatter plot and Box plot

Scatter Plot



Box Plot



The observations out of box are treated as outliers in data

Outliers and Missing Data treatment:

Missing Values

- In **R**, missing values are represented by the symbol **NA** (not available).
- Impossible values (e.g., dividing by zero) are represented by the symbol **NaN** (not a number) and R outputs the result for dividing by zero as 'Inf'(Infinity).
- **R** uses the same symbol for character and numeric data.
- To Test missing values: is.na () function

Example:

```
y <- c(1,2,3,NA)
is.na(y)
[1] FALSE FALSE FALSE TRUE
mean(y)##Arithmetic functions on missing values
[1] NA
x <- c(1,2,NA,3)
  mean(x) # returns NA
  mean(x, na.rm=TRUE) # returns 2
```

- To remove missing values :**na.omit() function.**
  newdata<- na.omit(y)
- Alternative method using na.rm=TRUE
  mean(x, na.rm=TRUE)

PMM approach to treat missing values:

- **PMM**-> Predictive Mean Matching (PMM) is a semi-parametric imputation approach.
- It is similar to the regression method except that for each missing value, it fills in a value randomly from among the observed donor values from an observation
- whose regression-predicted values are closest to the regression-predicted value for the missing value from the simulated regression model.
- **MICE Package -> Multiple Imputation by Chained Equations**
- MICE uses PMM to impute missing values in a dataset.

1.6. Combining Data sets in R

Merge(): To merge two data frames (datasets) horizontally. In most cases, we can join two data frames by one or more common key variables (i.e., an inner join).

To merge two data frames by ID:

total <- merge(data frameA, data frameB, by="ID")

To merge on more than one criteria :To merge two data frames by ID and Country:

total <- merge(data frameA,data frameB,by=c("ID","Country"))

To join two data frames (datasets) vertically : **rbind** function. The two data frames **must** have the same variables, but they do not have to be in the same order.

Example:

total <- rbind(data frameA, data frameB)

Plyr package: Tools for Splitting, Applying and Combining Data. We use rbind.fill() in plyr package in R. It binds or combines a list of data frames filling missing columns with NA.

Example: rbind.fill(mtcars[c("mpg", "wt")], mtcars[c("wt", "cyl")])

In this all the missing value will be filled with NA.

1.7. Function and Loops:

In R functions and loop structure are created using for,while along with conditional statements if-else.

1.7.1. "FOR" Loop:

For loop is used to repeat an action for every value in a vector. "for" loop in R :

for(i in values){... do something ...}

This for loop consists of the following parts:

The keyword **for**, followed by parentheses.An identifier between the parentheses. In this example, we use i, but that can be any object name you like.The keyword **in**, which follows the identifier.

Syntax:

for (val in sequence) { statement}

```
Example :To count the number of even numbers in a vector.
#For Loop Example 1:
```

```
x<-c(34,5,8,1,2,67,98)
count<-0
for(y in x)
{
  if(y%%2==0)
  {
    count = count+1;
    print(paste("y=",y))
  }
}
print(paste("count=",count))

##The above can be written in simple as:
x<-c(34,5,8,1,2,67,98)
count<-0
for(y in x) if(y%%2==0){ count = count+1;print(paste("y=",y))}
print(paste("count=",count))
```

Output:

```
[1] "y= 34"
[1] "y= 8"
[1] "y= 2"
[1] "y= 98"
[1] "count= 4"
```

```
#For Loop Example 2:
for(i in -5:3) if(i>0) print(paste(i," is +ve ")) else
  if(i<0) print(paste(i," is -ve")) else print(paste(i," is Zero"))
```

output:

```
[1] "-5  is -ve"
[1] "-4  is -ve"
[1] "-3  is -ve"
[1] "-2  is -ve"
[1] "-1  is -ve"
[1] "0  is Zero"
[1] "1  is +ve "
[1] "2  is +ve "
[1] "3  is +ve "
```

```
#For Loop Example 3:
x<-c(2,NA,5)
for(n in x) print(n*2)
```

output:

```
[1] 4
[1] NA
[1] 10
```

1.7.2. While loop

while (test_expression) { statement }

```
Example1:
i<-1
while(i<=10) i=i+4
i
```

output:

```
 [1] 13
```

```
Example2:
i<-1
while(i<=10) {print(paste("i= ",i));i=i+4 }
i
```

output:

```
[1] "i=  1"
[1] "i=  5"
```

```
[1] "i=  9"
[1] 13
```

```
Example3:
i<-1
while(T) if(i>10) break else i=i+4
i
```

output:

```
[1] 13
```

1.7.4 repeat:

Syntax: repeat {   statements   }

```
#Example:
i<-1
repeat { i<-i+4 ; if(i>10) break }
i
```

output:

```
[1] 13
```

1.7.5. IF-ELSE Function

```
 if ( test_expression1) {    statement1} else if ( test_expression2) {    statement2} else
if ( test_expression3) {   statement3} else   statement4
```

Example1:

```
x <- -5
y <- if(x > 0) 5 else 6
[1] 6
```

Example2:

```
x <- 0
if (x < 0)
{   print("Negative number")}
else if (x > 0)
{   print("Positive number")}
else
  print("Zero")
```

Output:

```
[1] Zero
```

Example3:

```
##ifelse is a function works like if-else
i<-1
ival<-ifelse(i>0,3,4)
ival
```

Output:

```
[1] 3
```

Example 4:

```
i<-0
if(i) i else 0
```

Output:

```
[1] 0
```

```
i<--4
if(i) i else 0
```

Output:

```
[1] -4
```

User defined functions:

To define functions in R using function()

One-Line Functions:

```
Example:
>log2 = function(x) log(x, base = 2)
> log2(64)
[1] 6
```

Using Default Values in Functions:

Functionname<-function(argument1,...){function body}

It returns by default the last statement return value.

Example1:

```
> sum<-function(i,j){ i+j }      ## function sum will take two integers and return the sum
of integers
> sum(2,3)   ## calling of function
[1] 5
manning = function(radius, gradient, coef=0.1125) (radius^(2/3)*gradient^0.5/coef)
> manning(radius = 1, gradient = 1/500)
[1] 0.3975232
```

Manage your work to meet requirements (NOS 9001)
Understanding Learning objectives, Introduction to work & meeting requirements, Time Management, Work management & prioritization, Quality & Standards Adherence.

Understanding Learning objectives:
The benefits of this course include:

- Efficient and Effective time management
- Efficient – Meeting timelines
- Effective – Meeting requirement for desired output
- Awareness of the SSC (Sector Skill Council) environment and time zone understanding
- Awareness of the SSC environment and importance of meeting timelines to handoffs

Review the course objectives listed above.
"To fulfil these objectives today, we'll be conducting a number of hands-on activities. Hopefully we can open up some good conversations and some of you can share your experiences so that we can make this session as interactive as possible. Your participation will be crucial to your learning experience and that of your peers here in the session today."
Question: Please share your thoughts on following?
A. Time is perishable – Cannot be created or recovered
B. Managing is only option – Prioritize

Importance of Time Management

The first part of this session discusses the following:

- "Plan better avoid wastage"
- Understanding the timelines of the deliverables. Receiving the hand off from upstream teams
- at right time is critical to start self contribution and ensure passing the deliverables to
- downstream team.
- It is important to value others' time as well to ensure overall organizational timelines are met
- Share the perspective of how important is time specifically in a global time zone mapping scenario

"Yes, I have room in my schedule to attend a Time Management Seminar...the day after I retire!"

Suggested Responses:

- Time management has to be looked at an organizational level and not just individual level
- These Aspects teach us how to build the blocks of time management.

Time Management Aspects
Prompt participants to come up with some aspects and relate them back to here.

- □ Planning and goal setting
- □ Managing yourself
- □ Dealing with other people
- □ Your time
- □ Getting results

The first 4 Interconnect and Interact to give the 5th one – Results

Differentiate between Urgent and Important task

- Assume importance as they demand immediate attention Important Task
- May become urgent if left undone
- Usually have a long term effect To judge importance vs. urgency, gauge tasks in terms of
- Impact of doing them
- Effect of not doing them

Main aim of prioritization is to avoid a crisis
We must
Time Management quadrants
Schedule our Priorities as opposed to Prioritizing our Schedule
1. Urgent and Important – Do Now
2. Not Urgent and Important – Schedule on your calendar
3. Urgent and Not Important – Delegate, Automate or Decline
4. Not Urgent Not Important – Delegate, Automate or Decline

Check Your Understanding

1. True or False? Time can be stored.
   a. True
   b. False

Suggested Responses:
False – Time once lost cannot be gotten back – hence important to plan time utilization properly

2. True or False? Time is perishable
   a. True
   b. False

Suggested Responses:
True – Time lost is lost for every – lost moments cannot be gotten back

3. True or False? Time management is required both at individual level and organizational level.
   a. True
   b. False

Suggested Responses:
True – plan for activities organizational level and also at individual level

4. True or False? Activities should be judged basis Urgency and Importance
   c. True
   d. False

Suggested Responses:
True – prioritization should be based on 2x2 matrix of urgency and importance

Team Exercise

Ask the participants to pick up the items listed below and place them in the Urgent/Important quadrant. Discuss the rationale of their thoughts and categorization.

List the items and ask participants to classify them as per the quadrant.

Discuss the rationale of their thoughts and categorization.

Categorize the below items in the Time Management Quadrant

1. Wildly important goal
2. Last minute assignments from boss
3. Busy work
4. Personal health
5. Pressing problems
6. Crises
7. Planning
8. Time wasters
9. Professional development
10. Win-win performance agreement
11. Too many objectives
12. Vital customer call
13. Major Deadlines
14. Unimportant pre scheduled meetings
15. Meaningless management reports
16. Coaching and mentoring team
17. Low priority email
18. Other people's minor issues
19. Workplace gossip
20. Exercise
21. Needless interruptions
22. Defining contribution
23. Aimless Internet surfing
24. Irrelevant phone calls

Suggested Answers:

Depends on rationale shared

1. Wildly important goal – Q1
2. Last minute assignments from boss – Q1
3. Busy work – Q4 – Consumes time however not pressing
4. Personal health – Q4 – requires planning and care not pressing
5. Pressing problems – Q1 – has to be solved immediately
6. Crises – Q1 – have to tended to immediately
7. Planning – Q2 – Important but not urgent; should be done before crisis
8. Time wasters – Q4
9. Professional development – Q2
10. Win-win performance agreement – Q2 – Expectation setting part of planning
11. Too many objectives – Q3 – Prioritize further to establish which are important and pressing
12. Vital customer call – Q1 – Customer centricity
13. Major Deadlines – Q1
14. Unimportant pre scheduled meetings – Q3

15. Meaningless management reports – Q3 – Prioritize further to establish which are important and pressing
16. Coaching and mentoring team – Q2
17. Low priority email – Q3 – Prioritize further to establish which are important and pressing
18. Other people’s minor issues – Q3 – May not be urgent but important for team building
19. Workplace gossip – Q4 – Non value add; occasionally creates negativity

Exercise – Q4 –
Important for health and personal wellbeing. To be done in spare and leisure time.
Cannot be ignored.

21. Needless interruptions – Q3
22. Defining contribution – Q2
23. Aimless Internet surfing – Q4
24. Irrelevant phone calls – Q4 – Reserve and avoid

Summary

- It is important to manage time.
- To manage time one must:
- Prioritize
- Define Urgency
- Define Importance

Work Management and Prioritization

Preparing morning tea is a good example. Define time, no of family members, preparation required at night and then in the morning. Perfect execution to ensure good morning tea !!! with family.

Gather responses.

Start the session by connecting the course content to the candidate responses.

Work Management

Six steps for expectation setting with the stakeholders

1. Describe the jobs in terms of major outcomes and link to the organization's need The first step in expectation setting is to describe the job to the employees. Employees need to feel there is a greater value to what they do. We need to feel out individual performance has an impact on the organization's mission.

Answer this question: My work is the key to ensuring the organization's success because…

While completing the answer link it to

- Job Description
- Team and Organization's need
- Performance Criteria

2. Share expectations in terms of work style While setting expectation, it's not only important to talk about the "what we do" but also on "how we expect to do it". What are the ground rules for communication at the organization?

Sample ground rules

- Always let your tam know where are the problems. Even if you have a solution, no one likes surprises.
- Share concerns openly and look for solutions
- If you see your colleagues doing something well, tell them. If you see them doing something poorly, tell them.

Sample work style questions

- Do you like to think about issues by discussing them in a meeting or having quite time alone?
- How do you prefer to plan your day?

3. Maximize Performance -

Identify what is required to complete the work: Supervisor needs / Employee needs. Set input as well as output expectations.

In order to ensure employees are performing at their best, the supervisor needs to provide not only the resource (time, infrastructure, desk, recognition etc.) but also the right levels of direction (telling how to do the task) and support (engaging with employees about the task).

4. Establish priorities.

Establish thresh holds and crisis plan Use the time quadrant to establish priorities. Refer to earlier session.

5. Revalidate understanding.

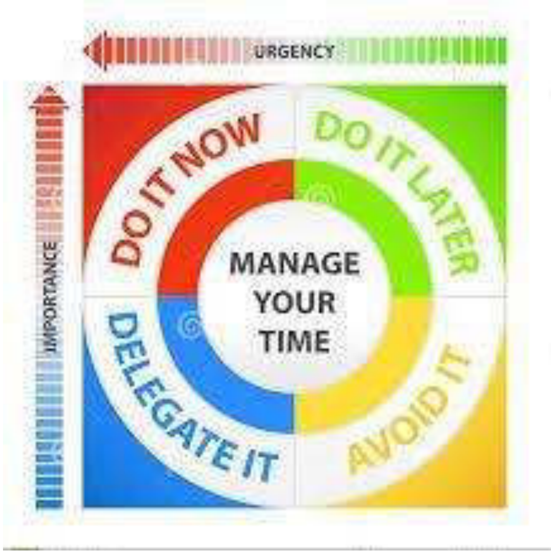
Create documentation and communication plan to establish all discussion

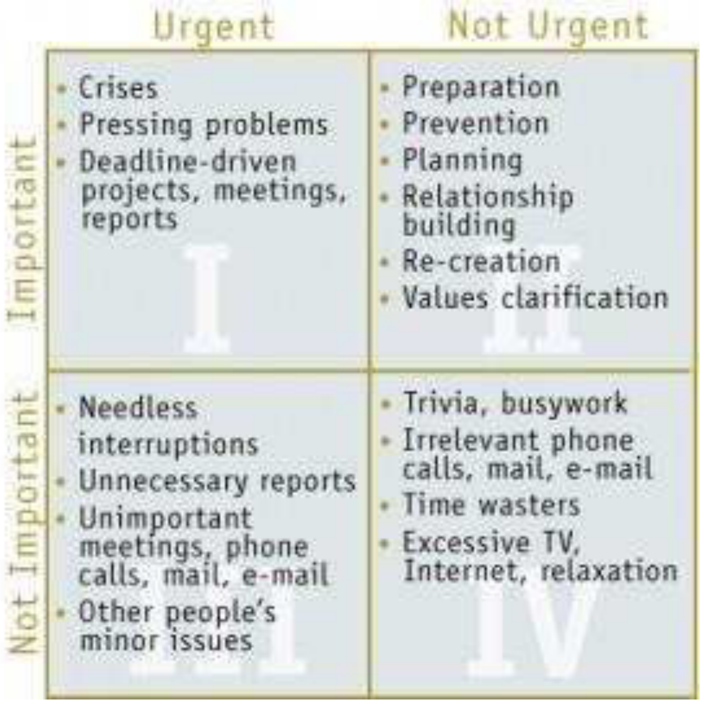
When you are having a conversation about expectations with stakeholders, you're covering lot of details so you'll need to review to make sure you both have a common understanding of the commitments you have made.

6. Establish progress check

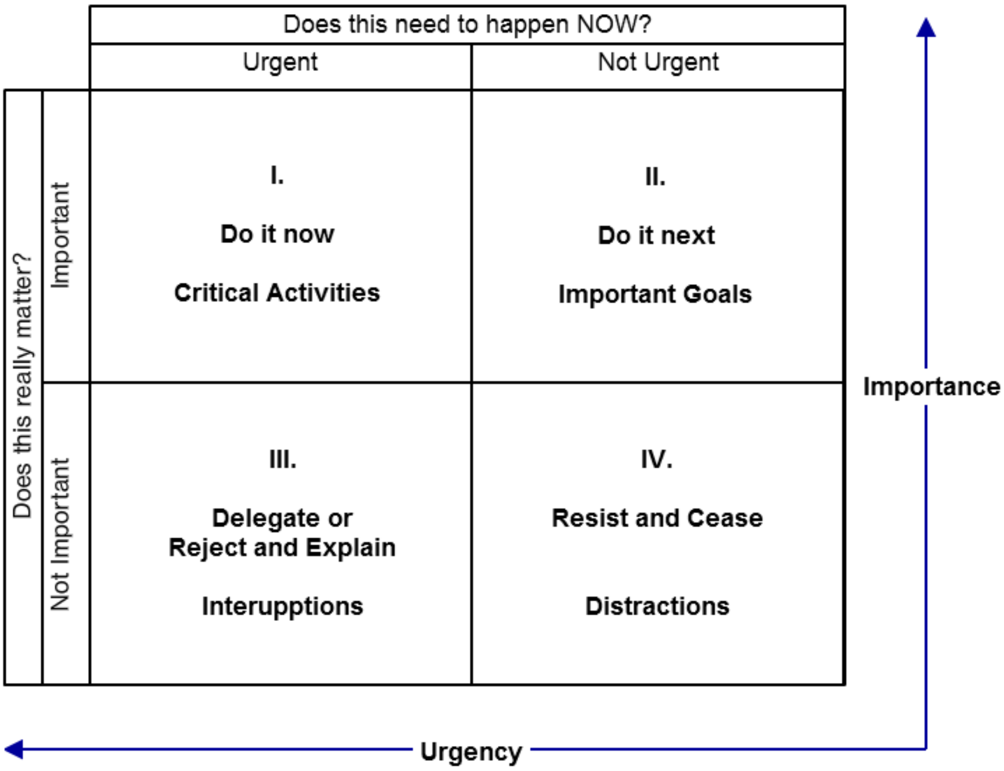
No matter how careful you have been in setting expectations, you'll want to follow up since there will be questions as work progresses.

Schedule an early progress check to get things started the right way, and agreed on scheduled/unscheduled further checks. Acknowledge good performance and point your ways to improve.





	Urgent	Not Urgent
Important	• Crises • Pressing problems • Deadline-driven projects, meetings, reports I	• Preparation • Prevention • Planning • Relationship building • Re-creation • Values clarification II
Not Important	• Needless interruptions • Unnecessary reports • Unimportant meetings, phone calls, mail, e-mail • Other people's minor issues III	• Trivia, busywork • Irrelevant phone calls, mail, e-mail • Time wasters • Excessive TV, Internet, relaxation IV



		Does this need to happen NOW?	
		Urgent	Not Urgent
Does this really matter?	Important	**I.** **Do it now** **Critical Activities**	**II.** **Do it next** **Important Goals**
	Not Important	**III.** **Delegate or Reject and Explain** **Interupptions**	**IV.** **Resist and Cease** **Distractions**

***** End of Unit 1 *****

Introduction to Analytics

(Associate Analytics – I)

UNIT II

Summarizing Data & Revisiting Probability (NOS 2101)

Summary Statistics - Summarizing data with R, Probability, Expected, Random, Bivariate Random variables, Probability distribution. Central Limit Theorem etc.

Work effectively with Colleagues (NOS 9002)

Introduction to work effectively, Team Work, Professionalism, Effective Communication skills, etc.

S.No	Content
2.1	Summary Statistics - Summarizing data with R
2.2	Probability
2.3	Expected
2.4	Random Variables
2.5	Bivariate Random variables
2.6	Probability distribution
2.7	Central Limit Theorem etc.
2.8	**Work effectively with Colleagues (NOS 9002)**

2.1. Summary Statistics - Summarizing data with R:

Example1:

```
> grass
rich    graze
1   12   mow
2  15    mow
3   17   mow
4  11    mow
5  15    mow
6   8    unmow
7   9   unmow
8   7   unmow
9   9   unmow
```

a) summary():

It gives the summary statistics of data object in terms of min, max,1st Quartile and 3rd Quartile mean/median values.

```
> x<-c(1,2,3,4,5,6,7,8,9,10,11,12)
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.75    6.50    6.50    9.25   12.00
> summary(grass)
Rich  graze
Min. : 7.00  mow :5  1st Qu  : 9.00
 unmow:4 Median :11.00   Mean :11.44
3rd Qu.:15.00  Max. :17.00
> summary(graze)
Length    Class        Mode
   9         character    character
```

```
> summary(grass$graze)
mow unmow
   5      4
```

b) str():

It gives the structure of data object in terms of class of object, No. of observations and each variable class and sample data.

Example2:

```
> 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 ...
```

```
> str(grass)
'data.frame': 9 obs. of 2 variables:
$ rich : int 12 15 17 11 15 8 9 7 9
$ graze: Factor w/ 2 levels "mow","unmow": 1 1 1 1 1 2 2
```

c) Tail():

It gives the last 6 observations of the given data object.

Example3: > tail(iris)

```
> tail(mtcars)
                mpg cyl  disp  hp drat    wt qsec vs am gear carb
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
> tail(HairEyeColor,2)
[1] 7 8
> tail(state.x77,2)
          Population Income Illiteracy Life Exp Murder HS Grad Frost  Area
Wisconsin       4589   4468        0.7    72.48    3.0    54.5   149 54464
Wyoming          376   4566        0.6    70.29    6.9    62.9   173 97203
```

```
> tail(grass)
  rich graze
4   11 mow
5   15 mow
6   8 unmow
7   9 unmow
8   7 unmow
9   9 unmow
```

d) **Head():**

It displays the top 6 observations from dataset

Example:

```
> head(iris)
```

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

```
>head(grass)
   rich graze
1  12 mow
2  15 mow
3   17 mow
4   11 mow
 5   15 mow
 6   8 unmow
```

e) Names():

It returns the coloum names

```
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
```

```
>names(grass)
graze  rich
```

f) nrow():

It returns the number of observations in the given dataset.

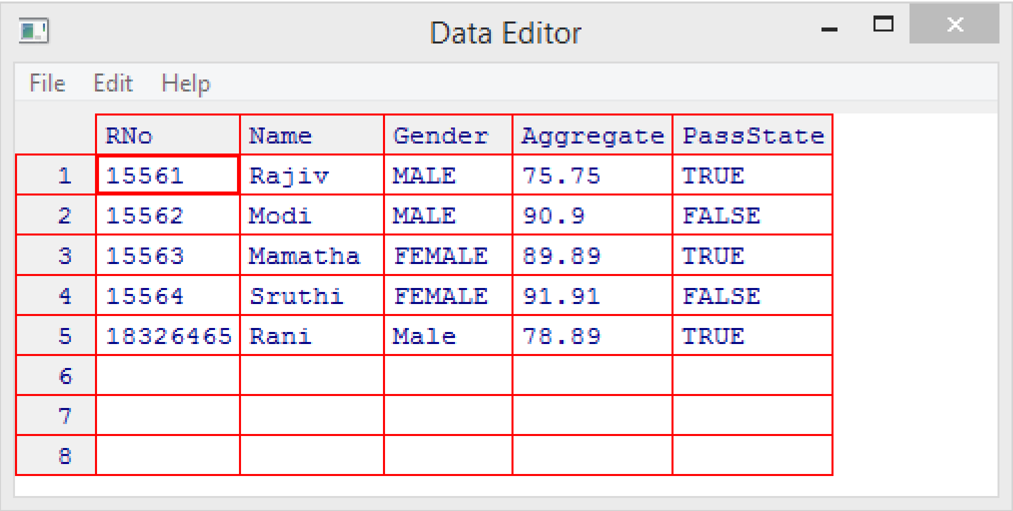
```
> dim(mtcars)
[1] 32 11
> nrow(mtcars)
[1] 32
> ncol(mtcars)
[1] 11
```

```
>nrow(iris)
9
```

g) fix(iris):

To fix the data in the given dataset.

```
> fix(mydFrame1)
```



	RNo	Name	Gender	Aggregate	PassState
1	15561	Rajiv	MALE	75.75	TRUE
2	15562	Modi	MALE	90.9	FALSE
3	15563	Mamatha	FEMALE	89.89	TRUE
4	15564	Sruthi	FEMALE	91.91	FALSE
5	18326465	Rani	Male	78.89	TRUE
6					
7					
8					

h) With():

To replace $along with attribute names

i) Aggregate():

To get the summary statistic of specific column with respect to different levels in the class attribute.

aggregate(x ~ y, data, mean)

Here x is numeric and y is factor type

```
>aggregate(rich~graze, grass, mean)
       graze  rich
     1   mow    14.00
2              unmow  8.25
```

j) Subset ():

To subset the data based on condition.

subset (data, x>7, select=c(x,y))

x is one of variable in data

select: to get the subset in specified order.

```
>subset(grass, rich>7, select=c(graze,rich))
    graze  rich
 1  mow    12
2   mow    15
3   mow    17
4   mow     11
 5  mow     15
 6  unmow  8
 7 unmow    9
 9  unmow   9
```

Lab activity:

A researcher wants to understand the data collected by him about 3 species of flowers.

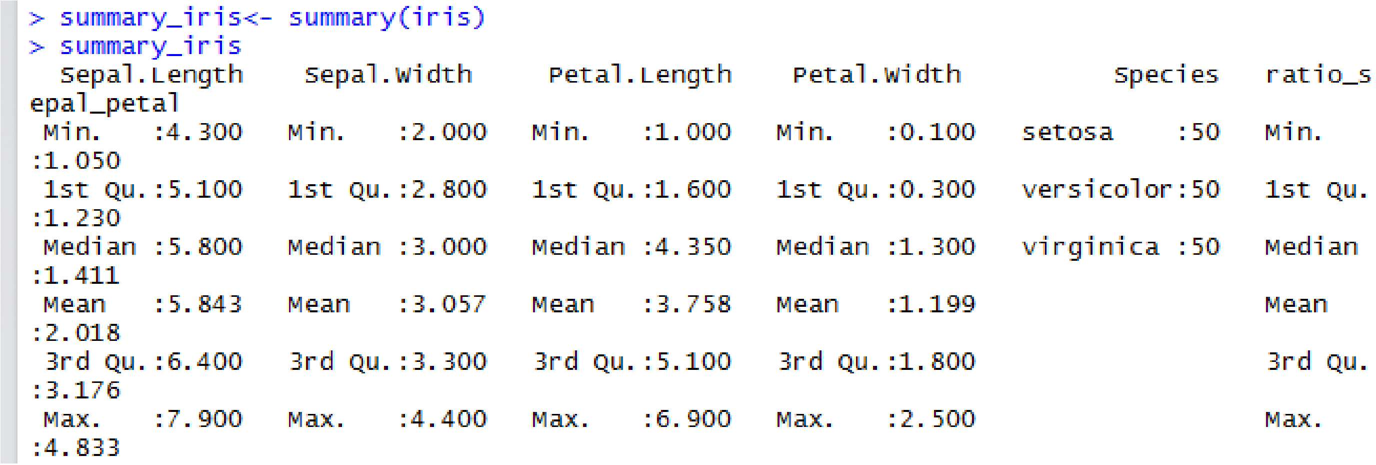
He wants the following:-

1. **The summary of 150 flower data including Sepal Length, Sepal Width, Petal Length and Petal Width. He also wants the summary of Sepal Length vs petal length.**

Solution:

To summarize data in R Studio we use majorly two functions Summary and Aggregate.

Using Summary command:



```
> summary_iris<- summary(iris)
> summary_iris
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species     ratio_s
epal_petal
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50   Min.   
:1.050  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50   1st Qu.
:1.230  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50   Median 
:1.411  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                   Mean   
:2.018  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                   3rd Qu.
:3.176  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                   Max.   
:4.833  
```

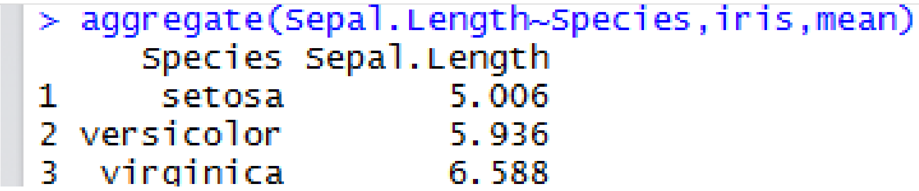
We get Min, Max, 1st Quartile, 3rd Quartile, Median, Mean as an output of summary() command.

2. **He wants to understand the mean Petal Length of each species.**

Solution:

For getting detailed output of one or more functions we use aggregate() command.

Using Aggregate () command:



```
> aggregate(Sepal.Length~Species,iris,mean)
     Species Sepal.Length
1     setosa        5.006
2 versicolor        5.936
3  virginica        6.588
```

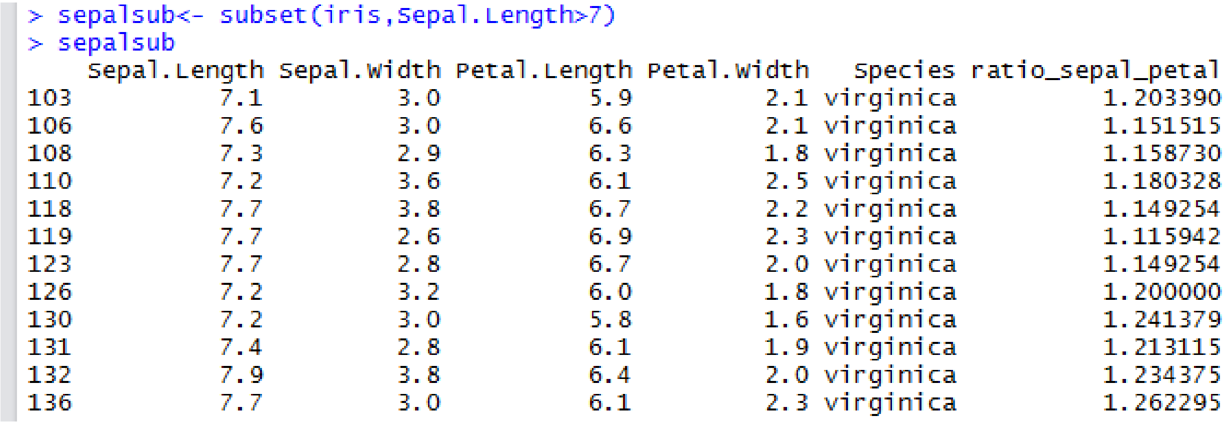
3. He wants to segregate the data of flowers having Sepal length greater than 7.

In the above example, we have calculated the mean sepal length of different species. Similarly we can calculate other functions also like frequency, median, summation etc.

For more details in terms of argument of Aggregate () command we use? Aggregate command to get help.

We also use subset () function to form subsets of data.

Using subset () command:

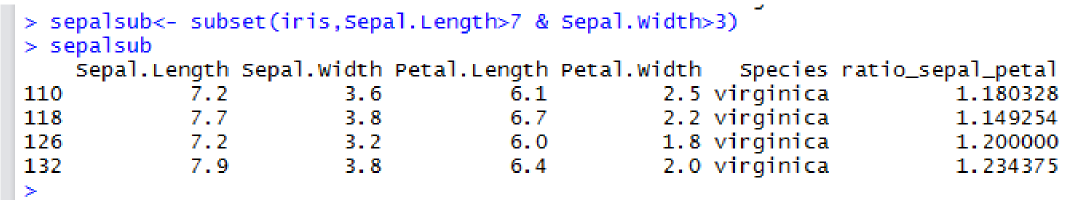


```
> sepalsub<- subset(iris,Sepal.Length>7)
> sepalsub
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species ratio_sepal_petal
103          7.1         3.0          5.9         2.1 virginica          1.203390
106          7.6         3.0          6.6         2.1 virginica          1.151515
108          7.3         2.9          6.3         1.8 virginica          1.158730
110          7.2         3.6          6.1         2.5 virginica          1.180328
118          7.7         3.8          6.7         2.2 virginica          1.149254
119          7.7         2.6          6.9         2.3 virginica          1.115942
123          7.7         2.8          6.7         2.0 virginica          1.149254
126          7.2         3.2          6.0         1.8 virginica          1.200000
130          7.2         3.0          5.8         1.6 virginica          1.241379
131          7.4         2.8          6.1         1.9 virginica          1.213115
132          7.9         3.8          6.4         2.0 virginica          1.234375
136          7.7         3.0          6.1         2.3 virginica          1.262295
```

4. He wants to segregate the data of flowers having Sepal length greater than 7 and Sepal width greater than 3 simultaneously.

Solution:

When we have to use more than 1 condition then we use & as shown below

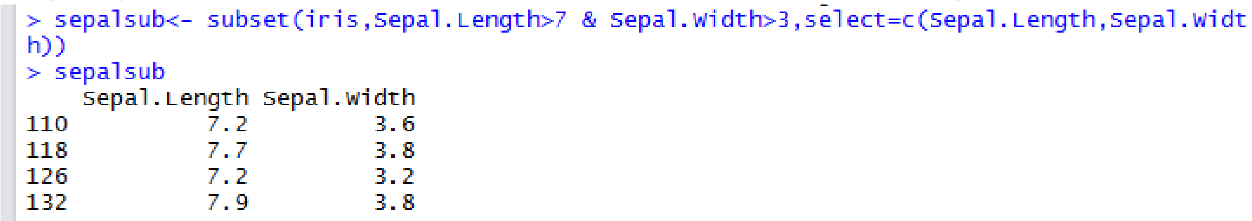


```
> sepalsub<- subset(iris,Sepal.Length>7 & Sepal.Width>3)
> sepalsub
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species ratio_sepal_petal
110          7.2         3.6          6.1         2.5 virginica          1.180328
118          7.7         3.8          6.7         2.2 virginica          1.149254
126          7.2         3.2          6.0         1.8 virginica          1.200000
132          7.9         3.8          6.4         2.0 virginica          1.234375
>
```

5. He wants to view 1st 7 rows of data .

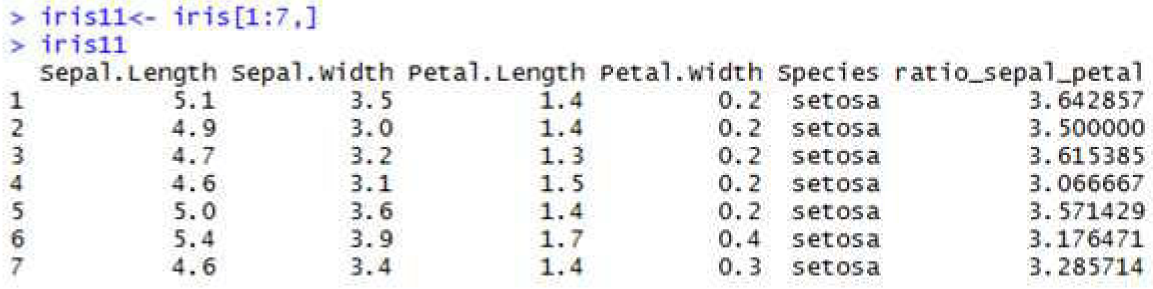
Solution:

For getting only few columns of requirement we use select () command in the argument:



```
> sepalsub<- subset(iris,Sepal.Length>7 & Sepal.Width>3,select=c(Sepal.Length,Sepal.Widt
h))
> sepalsub
    Sepal.Length Sepal.Width
110          7.2         3.6
118          7.7         3.8
126          7.2         3.2
132          7.9         3.8
```

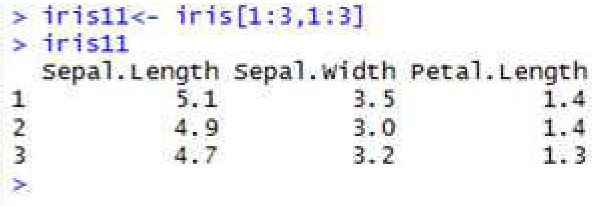
For subsetting data without ant condition just based on rows and columns we use square brackets .



```
> iris11<- iris[1:7,]
> iris11
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species ratio_sepal_petal
1          5.1         3.5          1.4         0.2  setosa          3.642857
2          4.9         3.0          1.4         0.2  setosa          3.500000
3          4.7         3.2          1.3         0.2  setosa          3.615385
4          4.6         3.1          1.5         0.2  setosa          3.066667
5          5.0         3.6          1.4         0.2  setosa          3.571429
6          5.4         3.9          1.7         0.4  setosa          3.176471
7          4.6         3.4          1.4         0.3  setosa          3.285714
```

6. He wants to view 1st 3 rows and 1st 3 columns of data.

Solution:



```
> iris11<- iris[1:3,1:3]
> iris11
  Sepal.Length Sepal.Width Petal.Length
1          5.1         3.5          1.4
2          4.9         3.0          1.4
3          4.7         3.2          1.3
>
```

2.2 Basics of Probability

We shall introduce some of the basic concepts of probability theory by defining some terminology relating to *random experiments* (i.e., experiments whose outcomes are not predictable).

2.2.1. Terminology

Def. Outcome

The end result of an experiment. For example, if the experiment consists of throwing a die H or T, the outcome would be anyone of the six faces, F1,F2,F3,F4,F5,F6.

Def. Random experiment:

If an '**experiment**' is conducted for number of times, under essentially identical conditions, which has a set of all possible outcomes associated with it, if the result is not certain and is any one of the several possible outcomes is known as **Random Experiment.** In Simple an experiment whose outcomes are not known in advance.

Ex: Throwing a fair die, tossing a honest coin, measuring the noise voltage at the terminals of a resistor, etc.

Def. Sample space

The sample space of a random experiment is a mathematical abstraction that represent all possible outcomes of the experiment. We denote the sample space by$S$

Ex: In a random experiment of tossing 2 coins, S = {HH, HT, TH, TT}.

In the case a die, S = {1,2,3,4,5,6}

Each outcome of the experiment is represented by a point in S and is called a sample point. We use s (with or without a subscript), to denote a sample point. An event on the sample space is represented by an appropriate collection of sample point(s).

Def: Equally Likely Events:

Events are said to be equally likely when there is no reason to expect anyone of them rather than anyone of the others.

Def. Exhaustive Events

All possible events in any trial are known as Exhaustive events.

Ex:

In tossing a coin, there are two exhaustive elementary events, viz. Head and Tail.

In drawing 3 balls out 9 in a box, there are 9_{C_3} (9C3) exhaustive elementary events.

Def. Mutually exclusive (disjoint) events

Two events A and B are said to be mutually exclusive if they have no common elements (or outcomes).Hence if A and B are mutually exclusive, they cannot occur together. i.e., if the happening of any one of the events in a trial excludes the happening of any of others. (Two or more of the events can't happen simultaneously in the same trial.)

Ex: In tossing coin occurrence of the outcome 'Head' excludes the occurrence of 'Tail'.

Def. Classical Definition of Probability:

In a random experiment, let there be n mutually exclusive and equally likely elementary events. Let E be an event of the experiment. If m events are favorable to E, then the probability of E (Chance of occurrence of E) is defined as

$$p(E) = \frac{m}{n} = \frac{No.of Events Favourable To E}{Total No.of Events}$$

Note:

1. $0 \le \frac{m}{n} \le 1$
2. $0 \le p(E) \le 1$ and $0 \le p(\bar{E}) \le 1$

Random Variables – Distribution function:

A random variable, aleatory variable or stochastic variable is a variable whose value is subject to variations due to chance (i.e. randomness, in a mathematical sense).

Def: A ***random variable***, usually written X, is a variable whose possible values are numerical outcomes of a random phenomenon

Let S be the sample of the random experiment. Random variable is a function whose domain is the set of outcomes w∈S and whose range is R, the set of real numbers. The random variable assigns a real value X(w) such that

1. The set {w/X(w)≤x} is an event for every x ∈ R, for which a probability is defined. This condition is called Measurability.
2. The probabilities of the events {w/X(w)=∞} and {w/X(w)=-∞} are equal to zero. i.e., p(X=∞) = p(X=-∞) = 0.
3. For A⊂S, there corresponds a set T⊂R called the image of A. Also for every T⊂R there exists in S the inverse image $X^{-1}(T) = \{w \in S \mid X(w) \in T\}$

In simple A random variable is a **real-valued function** defined on the points of a sample space.

Random variables are two broad categories:

- Random variable with discrete values
- Bivariate Random Variable

1. **Discrete Random Variable:** A random variable X which can take only a finite number of discrete values in an interval of domain is called a discrete random variable. In other words, if the random variable takes the values only on the set {0, 1,2,3,4,…n} is called Discrete Random variable.
   Ex: The printing mistakes in each page of a book, the number of tephone calls received by the receptionist are the examples of Discrete Random Variables.
   Thus to each outcome of S of a random experiment there corresponds a real number X(s) which is defined for each point of the sample S.
2. **Continuous Random Variable:** A Random variable X which can take values continuously i.e., which takes all possible values in a given interval is called a continuous random variable.
   Ex: the height, age and weight of individuals are the examples of continuous random variable
3. **Bivariate Random Variable:**

Bivariate Random Variables are those variables having only 2 possible outcomes.

Ex: Flip of coin(two outcomes: head/tail).

Probability distribution : Which describes how the values of a random variable are distributed. The probability distribution for a random variable X gives the possible values for X, and the probabilities associated with each possible value (i.e., the likelihood that the values will occur) The methods used to specify discrete prob. distributions are similar to (but slightly different from) those used to specify continuous prob. distributions.

Binomial distribution: The collection of all possible outcomes of a sequence of coin tossing

Normal distribution: The means of sufficiently large samples of a data population

Note: The characteristics of these theoretical distributions are well understood, they can be used to make Statistical inferences on the entire data population as a whole.

Example: Probability of ace of Diamond in a pack of 52 cards when 1 card is pulled out at random.

"At Random" means that there is no biased treatment

No. of Ace of Diamond in a pack = S = 1

Total no of possible outcomes = Total no. of cards in pack = 52

Probability of positive outcome = S/P = 1/52

That is we have 1.92% chance that we will get positive outcome.

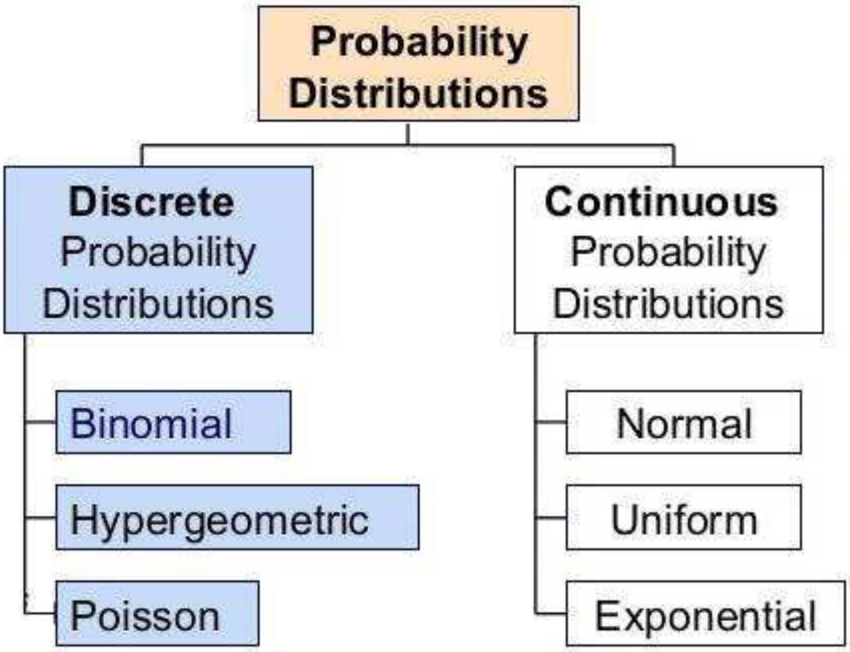
2.3.Expected value:

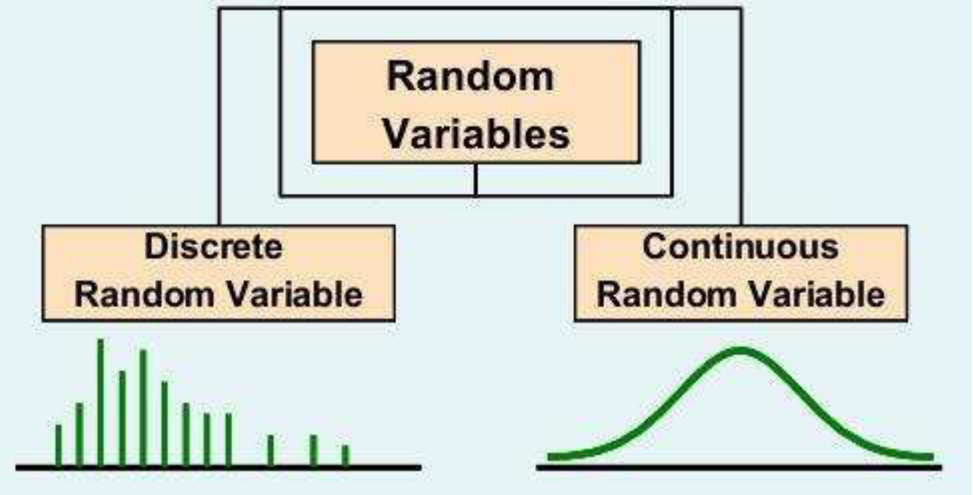
The expected value of a random variable is the long-run average value of repetitions of the experiment it represents.

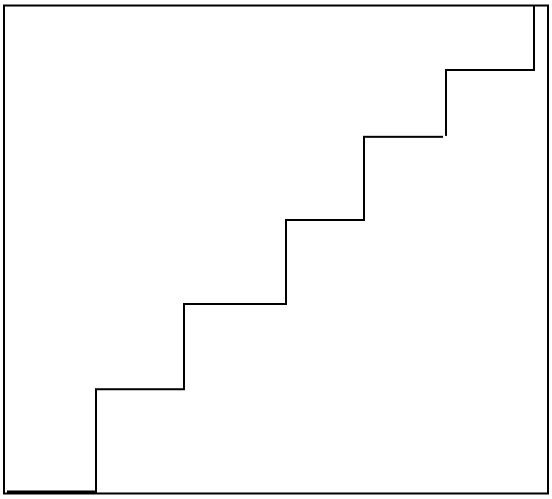
Example:

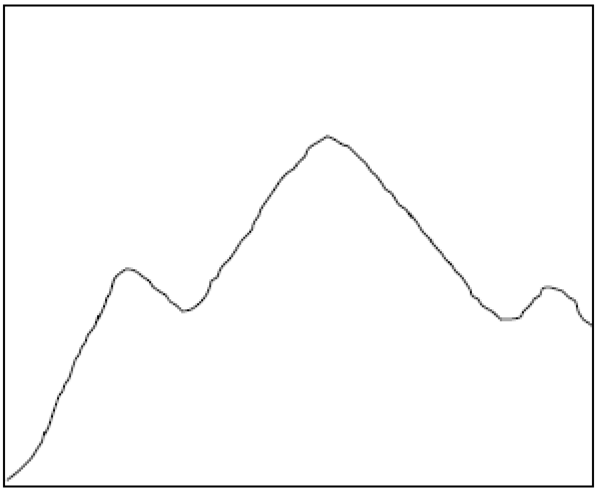
The expected value of a dice roll is 3.5 means the average of an extremely large number of dice rolls is practically always nearly equal to 3.5. Expected value is also known as the expectation, mathematical expectation, EV, mean, or first moment.

- **Expected value of a discrete random variable** is the probability-weighted average of all possible values
- **Continuous random variables are** the sum replaced by an integral and the probabilities by probability densities.


Probability Distributions
Discrete Probability Distributions
Continuous Probability Distributions
Binomial
Hypergeometric
Poisson
Normal
Uniform
Exponential







2.7.Probability Distribution Function (PDF):

It defines probability of outcomes based on certain conditions. Based on Conditions, there are majorly 5 types PDFs.

Types of Probability Distribution:

- Binomial Distribution
- Poisson Distribution
- Continuous Uniform Distribution
- Exponential Distribution
- Normal Distribution
- Chi-squared Distribution
- Student t Distribution
- F Distribution

Binomial Distribution

The **binomial distribution** is a discrete probability distribution. It describes the outcome of *n* independent trials in an experiment. Each trial is assumed to have only two outcomes, either success or failure. If the probability of a successful trial is *p*, then the probability of having *x* successful outcomes in an experiment of *n* independent trials is as follows.

$$f(x) = (n_{c_x})\, p^x (1-p)^{(n-x)} \qquad \textit{Where } x = 0, 1, 2, \; . \; . \; . \; , n$$

Problem

Ex: Find the probability of getting 3 doublets when a pair of fair dice are thrown for 10 times.

Solution

n=no. of trials=10,
p=probability of success i.e., getting a doublet = 6/36=1/6
q=probability of failure=1-p=1-(1/6)=5/6
r=no. of successes expected=3

$$P(x=3) = (n_{c_x})\, p^x (1-p)^{(n-x)} = (n_{c_r})\, p^r (q)^{(n-r)}$$

$$= (10_{c_3})\, p^3 (q)^{(10-3)}$$

$$= (10_{c_3}) \left(\frac{1}{6}\right)^3 \left(\frac{1}{6}\right)^{(10-3)} = 0.1550454$$

This can be computed in R as:

```
> choose(10,3)*((1/6)^3*(5/6)^7)  # Choose (10,3) is 10C3
[1] 0.1550454
```

```
This binomial distribution can be found using the formula in R as
> dbinom(3,size=10,prob=(1/6))
[1] 0.1550454
```

Problem: From the above problem find the probability of getting 3 or lesser doublets.

Solution:

```
>choose(10,0)*((1/6)^0*(5/6)^10  +
                 choose(10,1)*((1/6)^1*(5/6)^9) +
                                 choose(10,2)*((1/6)^2*(5/6)^8) +
                                               choose(10,3)*((1/6)^3*(5/6)^7))
[1] 0.9302722
```

This can be obtained using cumulative binomial distribution function as

```
>pbinom(3,size=10,prob=(1/6),lower=T)
       [1] 0.9302722

>pbinom(3,size=10,prob=(1/6),lower=F) #probability of getting 4 or more doublets
       [1] 0.06972784

Note:> 0.9302722+ 0.06972784 = 1
```

Problem2

Suppose there are twelve multiple choice questions in an English class quiz. Each question has five possible answers, and only one of them is correct. Find the probability of having four or less correct answers if a student attempts to answer every question at random.

Solution

Since only one out of five possible answers is correct, the probability of answering a question correctly by random is 1/5=0.2. We can find the probability of having exactly 4 correct answers by random attempts as follows.

dbinom(x,size,prob)

x: no. of successful outcomes (favourable)

size: n no.of independent trials

prob: probability of successful trial p

```
> dbinom(4, size=12, prob=0.2)
[1] 0.1329
```

To find the probability of having four or less correct answers by random attempts, we apply the function dbinom with *x* = 0,…,4.

```
> dbinom (0, size=12, prob=0.2) +
        dbinom (1, size=12, prob=0.2) +
                dbinom(2, size=12, prob=0.2) +
                        dbinom(3, size=12, prob=0.2) +
                                dbinom(4, size=12, prob=0.2)
[1] 0.92744
```

Alternatively, we can use the cumulative probability function for binomial distribution pbinom.

```
> pbinom (4, size=12, prob=0.2)
[1] 0.92744
```

Answer:

The probability of four or less questions answered correctly by random in a twelve question multiple choice quiz is 92.7%.

`Note:` `dbinom` gives the density, `pbinom` gives the cumulative distribution function, `qbinom` gives the quantile function and `rbinom` generates random deviates. If `size` is not an integer, `NaN` is returned.

Poisson Distribution

The **Poisson distribution** is the probability distribution of independent event occurrences in an interval. If λ is the mean occurrence per interval, then the probability of having x occurrences within a given interval is:

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad \text{Where } x = 0, 1, 2, \ldots$$

Problem

If there are twelve cars crossing a bridge per minute on average, find the probability of having seventeen or more cars crossing the bridge in a particular minute.

Solution

The probability of having *sixteen or less* cars crossing the bridge in a particular minute is given by the function ppois.

```
> ppois(16, lambda=12)   # lower tail
[1] 0.89871
```

Hence the probability of having seventeen or more cars crossing the bridge in a minute is in the *upper tail* of the probability density function.

```
> ppois(16, lambda=12, lower=FALSE)   # upper tail
[1] 0.10129
```

Similarly we can find the following:

```
> rpois(10, lambda=12)
 [1] 17 10  8 22  5 10 12 12  7 12
> dpois(16, lambda=12)
[1] 0.05429334
```

Answer

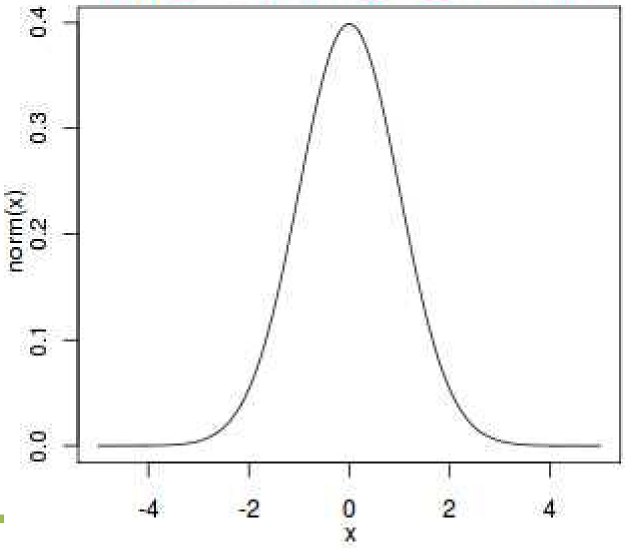
If there are twelve cars crossing a bridge per minute on average, the probability of having seventeen or more cars crossing the bridge in a particular minute is 10.1%.

Normal Distribution

The **normal distribution** is defined by the following probability density function, where μ is the population mean and σ^2 is the variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Figure 1: Normal Distribution



If a random variable X follows the normal distribution, then we write:

$X \sim N(\mu, \sigma^2)$

In particular, the normal distribution with $\mu = 0$ and $\sigma = 1$ is called the *standard normal distribution*, and is denoted as $N(0,1)$. It can be graphed as follows.

Figure 1 shows the normal distribution of sample data. The shape of a normal curve is highly dependent on the standard deviation.

Importance of Normal Distribution:

- Normal distribution is a continuous distribution that is “bell-shaped”.
- Data are often assumed to be normal.
- Normal distributions can estimate probabilities over a ***continuous interval of data values***.

Properties:

The normal distribution f(x), with any mean μ and any positive deviation σ, has the following properties:

- It is symmetric around the point x = μ, which is at the same time the mode, the median and the mean of the distribution.
- It is unimodal: its first derivative is positive for x < μ, negative for x > μ, and zero only at x = μ.
- Its density has two inflection points (where the second derivative of is zero and changes sign), located one standard deviation away from the mean as x = μ − σ and x = μ + σ.
- Its density is log-concave.
- Its density is infinitely differentiable, indeed super smooth of order 2.

Its second derivative f′′(x) is equal to its derivative with respect to its variance σ2.

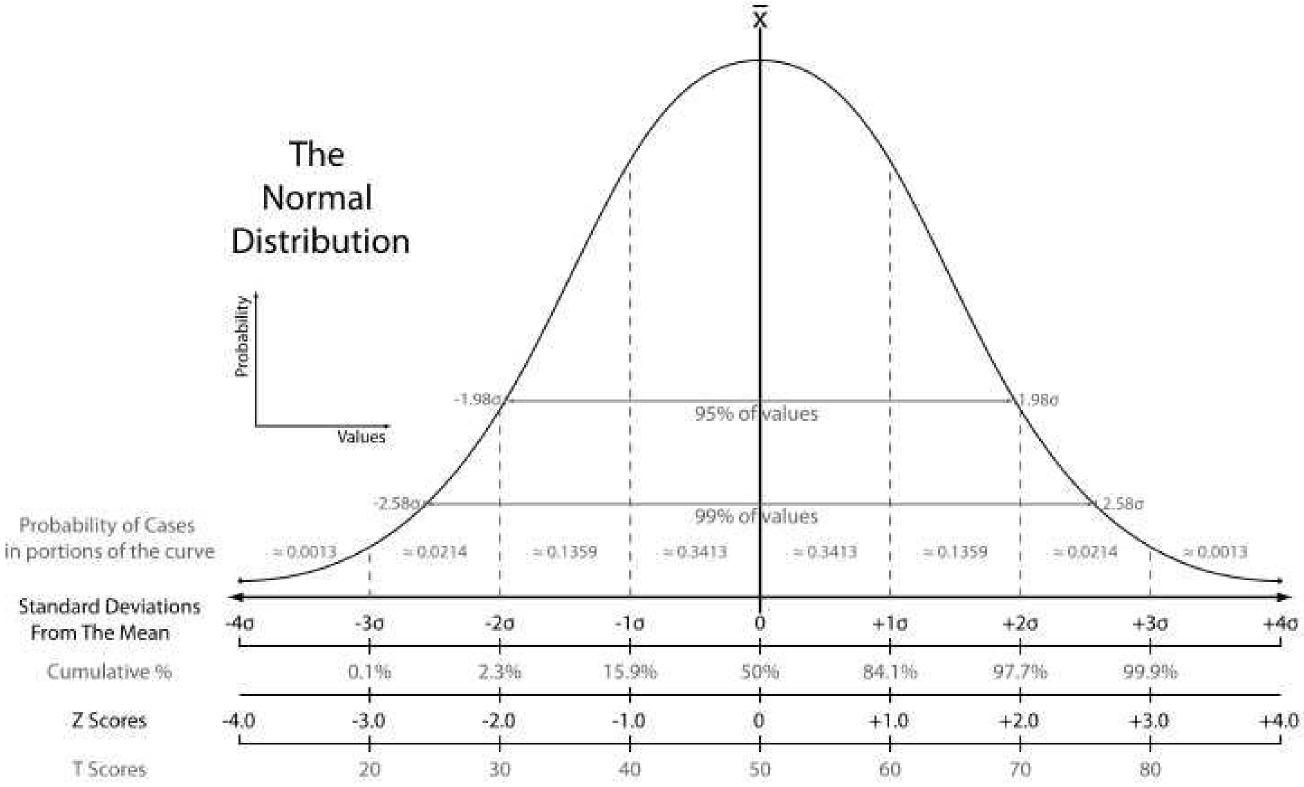


Figure 2: A normal distribution with Mean=0 and Standard deviation = 1

Normal Distribution in R:

Description:

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.

The normal distribution is important because of the **Central Limit Theorem**, which states that the population of all possible samples of size *n* from a population with mean *μ* and variance σ^2approaches a normal distribution with mean *μ* and σ^2/n when *n* approaches infinity.

Problem exam fits a normal distribution. Furthermore, the mean test score is 72, and the standard deviation is 15.2. What is the percentage of students scoring 84 or more in the exam?

Solution

We apply the function pnorm of the normal distribution with mean 72 and standard deviation 15.2. Since we are looking for the percentage of students scoring higher than 84, we are interested in the *upper tail* of the normal distribution.

```
> pnorm(84, mean=72, sd=15.2, lower.tail=FALSE)
[1] 0.21492
```

Answer

The percentage of students scoring 84 or more in the college entrance exam is 21.5%.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
 rnorm(n, mean = 0, sd = 1)
```

Arguments

- **x, q vector of quantiles.**
- **P vector of probabilities.**
- **N number of observations. If length(n) > 1, the length is taken to be the number required.**
- **Mean vector of means.**
- **Sd vector of standard deviations.**
- **log, log. P logical; if TRUE, probabilities p are given as log(p).**
- **lower.tail logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.**
- **rnorm(n, mean = 0, sd = 1) as default**

The Central Limit Theorem

The central limit theorem and the law of large numbers are the two *fundamental theorems* of probability. Roughly, the central limit theorem states that the distribution of the sum (or average) of a large number of independent, identically distributed variables will be approximately normal, regardless of the underlying distribution. The importance of the central limit theorem is hard to overstate; indeed it is the reason that many statistical procedures work.

The CLT says that if you take many repeated samples from a population, and calculate the *averages* or *sum* of each one, the collection of those averages will be normally distributed… and it doesn't matter what the shape of the source distribution is!

Lab Activity:

To generates 20 numbers with a mean of 5 and a standard deviation of 1:

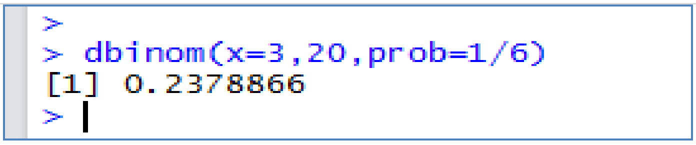
```
> rnorm(20, mean = 5, sd = 1)
[1] 5.610090 5.042731 5.120978 4.582450 5.015839 3.577376 5.159308 6.496983
[9] 3.071729 6.187525 5.027074 3.517274 4.393562 3.866088 4.533490 6.021554
[17] 5.359491 5.265780 3.817124 5.855315
> pnorm(5, mean = 5, sd = 1)
[1] 0.5
> qnorm(0.5, 5, 1)
[1] 5
> dnorm(c(4,5,6), mean = 5, sd = 1)
[1] 0.2419707 0.3989423 0.2419707
```

Lab Activity 3: Probability Theories:

1. If you throw a dice 20 times then what is the probability that you get following results:

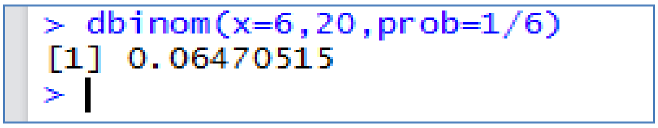
a. 3 sixes

Solution:



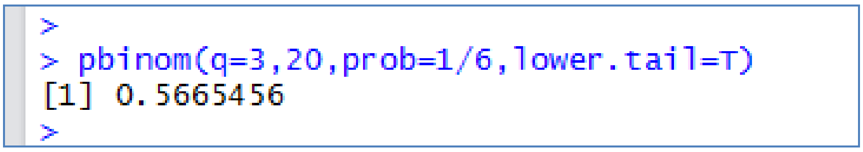
```
>
> dbinom(x=3,20,prob=1/6)
[1] 0.2378866
> |
```

b. 6 sixes



```
> dbinom(x=6,20,prob=1/6)
[1] 0.06470515
> |
```

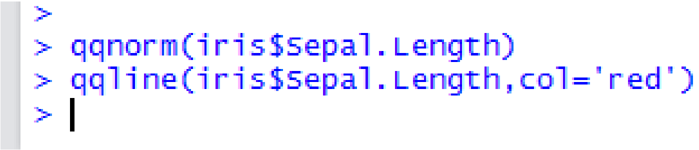
c. 1,2 and 3 sixes



```
>
> pbinom(q=3,20,prob=1/6,lower.tail=T)
[1] 0.5665456
>
```

2. In Iris data set check whether Sepal Length is normally distributed or not.

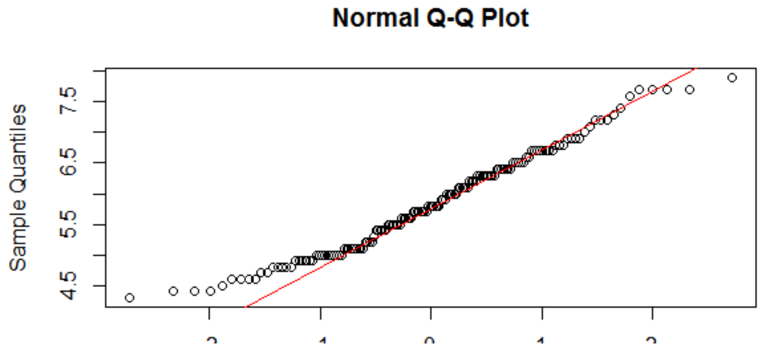
Use : To find if the Sepal Length is normally distributed or not we use 2 commands- qqnorm() &qqline().



```
>
> qqnorm(iris$Sepal.Length)
> qqline(iris$Sepal.Length,col='red')
> |
```

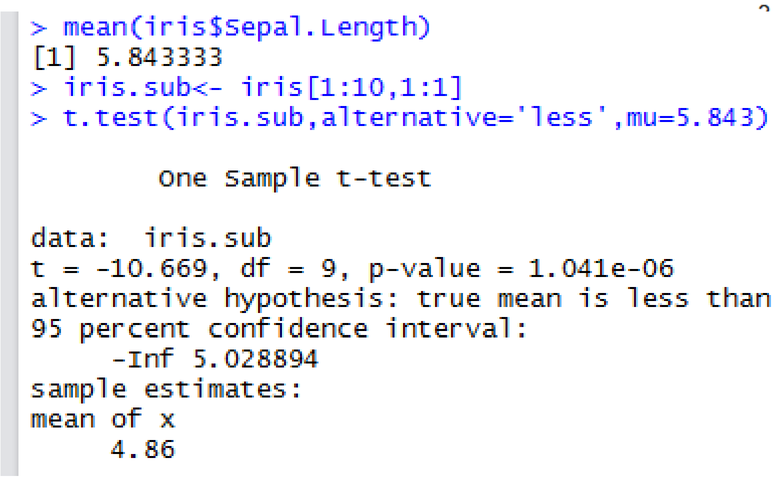
The qqnorm() shows the actual distribution of data while qqline() shows the line on which data would lie if the data is normally distributed. The deviation of plot from line shows that data is not normally distributed.

Figure3: Normal distribution of iris$ Sepal Length



3. Prove that population mean of Sepal length is different from mean of 1st 10 data significantly.

T-Test of sample subset of Iris data set.



```
> mean(iris$Sepal.Length)
[1] 5.843333
> iris.sub<- iris[1:10,1:1]
> t.test(iris.sub,alternative='less',mu=5.843)

        One Sample t-test

data:  iris.sub
t = -10.669, df = 9, p-value = 1.041e-06
alternative hypothesis: true mean is less than 5.843
95 percent confidence interval:
     -Inf 5.028894
sample estimates:
mean of x
     4.86
```

Here p-value is much less than 0.05. So we reject the null hypothesis and we accept the alternate hypothesis which says that mean of sample is less than the population mean.

$\mu_s < \mu_p$

Also sample mean is 4.86 and degree of freedom if 9 which is sample size -1.

Similarly we can do two sided test by writing alternative= “two sided”. And also paired sample t-test by using paired=TRUE as the part of argument.

Work effectively with Colleagues (NOS 9002)

Introduction to work effectively, Team Work, Professionalism, Effective Communication skills, etc.

Refer Students Hand Book and the ppt issued.

Introduction to work effectively

Team Work

1. Ways to Be More Effective at Work
   a) Trim Your Task List
   b) Swap Your To-Do List for a Schedule
   c) Stop While You're Still On a Roll
   d) Stay Organized
   e) Make Bad Habits More Difficult to Indulge(Spoil).
   f) Prioritize
   g) Tackle Your Most Important Tasks First
   h) Plan Tomorrow Tonight
   i) Use Idle Time to Knock Out Admin Tasks
   j) Schedule Meetings With Yourself
   k) Change Your Self-Talk
   l) Communicate and Clarify
   m) Find Ways to Do More of the Work You Enjoy

Team Work

1. What is team work?
2. How is it more advantageous?

A team comprises a group of people linked in a common purpose.

- Teams are especially appropriate for conducting tasks that are high in complexity and have many interdependent subtasks.
- Coming together is a beginning, keeping together is progress and working together is success.
- A team is a number of people associated together in work or activity.
- In a good team members create an environment that allows everyone to go beyond their limitation.

Why do we need teamwork –to make the organization profitable.

Team work vs. Individual work

- Team Work:
- Work Agree on goals/milestones
- Establish tasks to be completed
- Communicate / monitor progress
- Solve Problem
- Interpret Results
- Agree completion of projects

Individual work

- Work on tasks
- Work on new / revised tasks

Team Development

- Team building is any activity that builds and strengthens the team as a team.

Team building fundamentals

- Clear Expectations – Vision/Mission
- Context – Background – Why participation in Teams?
- Commitment – dedication – Service as valuable to Organization & Own
- Competence – Capability – Knowledge
- Charter – agreement – Assigned area of responsibility
- Control – Freedom & Limitations
- Collaboration – Team work
- Communication

- Creative Innovation
- Consequences – Accountable for rewards
- Coordination
- Cultural Change

Roles of team member

- Communicate
- Don't Blame Others
- Support Group Member's Ideas
- No Bragging(Arrogant) – No Full of yourself
- Listen Actively
- Get Involved
- Coach, Don't Demonstrate
- Provide Constructive Criticism
- Try To Be Positive
- Value Your Group's Ideas

Team Work: Pros and Cons

Summary:

- A team comprises a group of people linked in a common purpose.
- · Team work is essential to the success of every organization. In a good team, members create an environment that allows everyone to go beyond their limitation.
- · Some of the fundamentals on which a team is built are: Collaboration, Clear Expectations and Commitment

Professionalism

- ➢ Professionalism is the competence or set of skills that are expected from a professional.
- ➢ Professionalism determines how a person is perceived by his employer, co-workers, and casual contacts.
- ➢ How long does it take for someone to form an opinion about you?
- ➢ Studies have proved that it just takes six seconds for a person to form an opinion about another person.

How does someone form an opinion about you?

- ➢ Eye Contact – Maintaining eye contact with a person or the audience says that you are confident. It says that you are someone who can be trusted and hence can maintain contact with you.
- ➢ Handshake – Grasp the other person’s hand firmly and shake it a few times. This shows that you are enthusiastic.
- ➢ Posture – Stand straight but not rigid, this will showcase that you are receptive and not very rigid in your thoughts.
- ➢ Clothing – Appropriate clothing says that you are a leader with a winning potential.

How to exhibit professionalism?

- ➢ Empathy (compassion)
- ➢ Positive Attitude
- ➢ Teamwork
- ➢ Professional Language
- ➢ Knowledge
- ➢ Punctual
- ➢ Confident
- ➢ Emotionally stable

Grooming

What are the colours that one can opt for work wear?

- A good rule of thumb is to have your pants, skirts and blazers in neutral colours. Neutrals are not only restricted to grey brown and off white - you can also take advantage of the beautiful navies, forest greens, burgundies, tans and caramel tones around.
- Pair these neutrals with blouses, scarves or other accessories in accent colours - ruby red, purple, teal blue, soft metallic and pinks are some examples.

Things to remember

- Wear neat clothes at work which are well ironed and do not stink.
- Ensure that the shoes are polished and the socks are clean
- Cut your nails on a regular basis and ensure that your hair is in place.
- Women should avoid wearing revealing clothes at work.
- Remember that the way one presents oneself plays a major role in the professional world

Effective Communication

- Effective communication is a mutual understanding of the message.
- Effective communication is essential to workplace effectiveness
- The purpose of building communication skills is to achieve greater understanding and meaning between people and to build a climate of trust, openness, and support.
- A big part of working well with other people is communicating effectively.

Things to remember

- Wear neat clothes at work which are well ironed and do not stink.
- Ensure that the shoes are polished and the socks are clean
- Cut your nails on a regular basis and ensure that your hair is in place.
- Women should avoid wearing revealing clothes at work.
- Remember that the way one presents oneself plays a major role in the professional world

Effective Communication

- Effective communication is a mutual understanding of the message.
- Effective communication is essential to workplace effectiveness
- The purpose of building communication skills is to achieve greater understanding and meaning between people and to build a climate of trust, openness, and support.
- A big part of working well with other people is communicating effectively.

- Sometimes we just don’t realize how critical effective communication is to getting the job done.

What is Effective Communication?

- We cannot not communicate.
- **The question is: Are we communicating what we intend to communicate?**
- **Does the message we send match the message the other person receives?**
- Impression = Expression
- Real communication or understanding happens only when the receiver’s impression matches what the sender intended through his or her expression.
- So the goal of effective communication is a mutual understanding of the message.

Forms of Communication

1. Verbal communication
2. Non verbal communication
3. Written communication

Verbal Communication :

- Verbal communication refers to the use of sounds and language to relay a message
- It serves as a vehicle for expressing desires, ideas and concepts and is vital to the processes of learning and teaching.
- verbal communication acts as the primary tool for expression between two or more people

Types of verbal communication

- Interpersonal communication and public speaking are the two basic types of verbal communication.
- Whereas public speaking involves one or more people delivering a message to a group
- interpersonal communication generally refers to a two-way exchange that involves both talking and listening.

Forms of non verbal communication

1. Ambulation is the way one walks
2. Touching is possibly the most powerful nonverbal communication form.
3. Eye contact is used to size up the trustworthiness of another.
4. 4.Posturing can constitute a set of potential signals that communicate how a person is experiencing the environment
5. Tics are involuntary nervous spasms that can be a key to indicate one is being threatened.
6. Sub-vocals are the non-words one says, such as "ugh" or "um." They are used when one is trying to find the right word..
7. Distancing is a person's psychological space. If this space is invaded, one can become somewhat tense, alert, or "jammed up.
8. Tics are involuntary nervous spasms that can be a key to indicate one is being threatened.
9. Sub-vocals are the non-words one says, such as "ugh" or "um." They are used when one is trying to find the right word..
10. Distancing is a person's psychological space. If this space is invaded, one can become somewhat tense, alert, or "jammed up.
11. Gesturing carries a great deal of meaning between people, but different gestures can mean different things to the sender and the receiver. This is especially true between cultures. Still, gestures are used to emphasize our words and to attempt to clarify our meaning.
12. Vocalism is the way a message is packaged and determines the signal that is given to another person. For example, the message, "I trust you," can have many meanings. "I trust you" could imply that someone else does not. "I trust you" could imply strong sincerity. "I trust you" could imply that the sender does not trust others.

Written Communication

- ➢ Written communication involves any type of message that makes use of the written word. Written communication is the most important and the most effective of any mode of business communication

- ➢ Examples of written communications generally used with clients or other businesses include email, Internet websites, letters, proposals, telegrams, faxes, postcards, contracts, advertisements, brochures, and news releases

Advantages and disadvantages of written communication:

Advantages

- ➢ Creates permanent record
- ➢ Allows to store information for future reference
- ➢ Easily distributed
- ➢ All recipients receive the same information
- ➢ Written communication helps in laying down apparent principles, policies and rules for running on an organization.
- ➢ It is a permanent means of communication. Thus, it is useful where record maintenance is required.
- ➢ Written communication is more precise and explicit
- ➢ Effective written communication develops and enhances organization's image
- ➢ It provides ready records and references
- ➢ Written communication is more precise and explicit.
- ➢ Effective written communication develops and enhances an organization's image
- ➢ Necessary for legal and binding documents

Disadvantages of Written Communication

- ➢ Written communication does not save upon the costs. It costs huge in terms of stationery and the manpower employed in writing/typing and delivering letters.
- ➢ Also, if the receivers of the written message are separated by distance and if they need to clear their doubts, the response is not spontaneous.
- ➢ - Written communication is time-consuming as the feedback is not immediate. The encoding and sending of message takes time.

2 – Ensuring Connectivity

- The content that comprises a piece of writing should reflect fluency and should be connected through a logical flow of thought, in order to prevent misinterpretation and catch the attention of the reader.
- Moreover, care should be taken to ensure that the flow is not brought about through a forced/deliberate use of connectives, as this make the piece extremely uninteresting and artificial.

3 – Steering Clear of Short Form

- ➢ People may not be aware of the meaning of various short forms and may thus find it difficult to interpret them.
- ➢ Moreover, short forms can at time be culture specific or even organization specific and may thus unnecessarily complicate the communication.

4 – Importance of Grammar, Spelling and Punctuation

- Improper grammar can at worst cause miscommunication and at least results in unwanted humour and should be thus avoided. So too, spellings can create the same effect or can even reflect a careless attitude on part of the sender.
- Finally, effective use of punctuations facilitates reading and interpretation and can in rare cases even prevent a completely different meaning, which can result in miscommunication

5 – Sensitivity to the Audience

- ➢ One needs to be aware of and sensitive to the emotions, need and nature of the audience in choosing the vocabulary, content, illustrations, formats and medium of communication, as a discomfort in the audience would hamper rather than facilitate communication.

6 – Importance of Creativity

- ➢ In order to hold the readers' attention one needs to be creative to break the tedium of writing and prevent monotony from creeping in.
- ➢ This is especially true in the case of all detailed writing that seeks to hold the readers' attention.

7 – Avoidance Excessive use of Jargons

- ➢ Excessive use of jargon(slang/terminology)

can put off a reader, who may not read further, as, unlike a captive audience, the choice of whether to participate in the communication rests considerably with the reader.

Go through the Facilitators Guide for Objective Questions/True or False Qns..They will be given for weekly tests.

***** End of Unit-2 *****

Unit III: SQL using R

3.1. Introduction to NoSQL:

Define Nosql Database:

NoSQL is originally referring to "non SQL" or "non-relational" and also called "Not only SQL" to emphasize that they may support SQL-like query languages. The RDBMS database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. NoSQL databases are increasingly used in big data and real-time web applications.

Benefits of NoSQL Database:

No SQL databases are more scalable and provide superior performance. The NoSQL data model addresses several issues that the relational model is not designed to address:

- Large volumes of structured, semi-structured, and unstructured data
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture

Classification of NoSQL databases based on data model: A basic classification based on data model, with examples:

- **Document**: Clusterpoint, Apache CouchDB, Couchbase, DocumentDB, HyperDex, Lotus Notes, MarkLogic, **MongoDB**, OrientDB, Qizx
- **Key-value**: CouchDB, Oracle NoSQL Database, Dynamo, FoundationDB, HyperDex, MemcacheDB, Redis, Riak, FairCom c-treeACE, Aerospike, OrientDB, MUMPS
- **Graph**: Allegro, Neo4J, InfiniteGraph, OrientDB, Virtuoso, Stardog
- **Multi-model**: OrientDB, FoundationDB, ArangoDB, Alchemy Database, CortexDB

Differences between SQL database and NoSQL Database:

		SQL Databases	**NOSQL Databases**
1.	**Types**	One type (SQL database) with minor variations	Many different types including key-value stores, document databases, wide-column stores, and graph databases
2.	**Development History**	Developed in 1970s to deal with first wave of data storage applications	Developed in 2000s to deal with limitations of SQL databases, particularly concerning scale, replication and unstructured data storage
3.	**Examples**	MySQL, Postgres, Oracle Database	MongoDB, Cassandra, HBase, Neo4j

4.	**Data Storage Model**	Individual records (e.g., "employees") are stored as rows in tables, with each column storing a specific piece of data about that record (e.g., "manager," "date hired," etc.), much like a spreadsheet. Separate data types are stored in separate tables, and then joined together when more complex queries are executed. For example, "offices" might be stored in one table, and "employees" in another. When a user wants to find the work address of an employee, the database engine joins the "employee" and "office" tables together to get all the information necessary.	Varies based on database type. For example: • key-value stores function similarly to SQL databases, but have only two columns ("key" and "value"), with more complex information sometimes stored within the "value" columns. • Document databases do away with the table-and-row model altogether, storing all relevant data together in single "document" in **JSON, XML**, or another format, which can nest values hierarchically.
5.	**Schemas**	Structure and data types are fixed in advance. To store information about a new data item, the entire database must be altered, during which time the database must be taken offline.	Typically dynamic. Records can add new information on the fly, and unlike SQL table rows, dissimilar data can be stored together as necessary. For some databases (e.g., wide-column stores), it is somewhat more challenging to add new fields dynamically.
6.	**Scaling**	Vertically, meaning a single server must be made increasingly powerful in order to deal with increased demand. It is possible to spread SQL databases over many servers, but significant additional engineering is generally required.	Horizontally, meaning that to add capacity, a database administrator can simply add more commodity servers or cloud instances. The database automatically spreads data across servers as necessary.
7.	**Development Model**	Mix of open-source (e.g., Postgres, MySQL) and closed source (e.g., Oracle Database)	Open-source
8.	**Supports Transactions**	Yes, updates can be configured to complete entirely or not at all	In certain circumstances and at certain levels (e.g., document level vs. database level)

9.	**Data Manipulation**	Specific language using Select, Insert, and Update statements, e.g. SELECT fields FROM table WHERE…	Through object-oriented APIs
10.	**Consistency**	Can be configured for strong consistency	Depends on product. Some provide strong consistency (e.g., MongoDB) whereas others offer eventual consistency (e.g., Cassandra)

Unit III: SQL using R

3.2. Connecting R to NoSQL databases

Lab Activity: No SQL Example: R script to access a XML file:

Step1: Install Packages plyr,XML

Step2: Take xml file url

Step3: create XML Internal Document type object in R using **xmlParse()**

Step4 :Convert xml object to list by using **xmlToList()**

Step5**:** convert list object to data frame by using **ldply(xl, data.frame)**

```
install.packages("XML")
install.packages("plyr")
> fileurl<-"http://www.w3schools.com/xml/simple.xml"
> doc<-xmlParse(fileurl,useInternalNodes=TRUE)
> class(doc)
[1] "XMLInternalDocument" "XMLAbstractDocument"
> doc
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories>
  </food>
  <food>
> xl<-xmlToList(doc)
> class(xl)
[1] "list"
> xl
$food
$food$name
[1] "Belgian Waffles"
$food$price
[1] "$5.95"
$food$description
```

```
[1] "Two of our famous Belgian Waffles with plenty of real maple syrup"
$food$calories
[1] "650"
$food
> data<-ldply(xl, data.frame)
> head(data)
   .id                        name price
1 food             Belgian Waffles $5.95
2 food   Strawberry Belgian Waffles $7.95
3 food Berry-Berry Belgian Waffles $8.95
4 food                French Toast $4.50
5 food         Homestyle Breakfast $6.95

Description:
1   Two of our famous Belgian Waffles with plenty of real maple syrup
2   Light Belgian waffles covered with strawberries and whipped cream
3   Light Belgian waffles covered with an assortment of fresh berries and whipped cream
4   Thick slices made from our homemade sourdough bread
5   Two eggs, bacon or sausage, toast, and our ever-popular hash browns
    calories
1     650
2     900
3     900
4     600
5     950
```

3.3. Excel and R integration with R connector

Different approaches in R to connect with Excel to perform read write and execute activities:

3.3.1. Read Excel spreadsheet in R:

Multiple Packages are available to access Excel sheet from R

1. **gdata:** This package requires you to install additional Perl libraries on Windows platforms but it's very powerful.

   ```
   require(gdata)
   myDf <- read.xls ("myfile.xlsx"), sheet = 1, header = TRUE)
   ```

2. **XLConnect:** It might be slow for large dataset but very powerful otherwise.

   ```
   require (XLConnect)
   ```

```
wb <- loadWorkbook("myfile.xlsx")
myDf <- readWorksheet(wb, sheet = "Sheet1", header = TRUE)
```

3. **xlsx:** This package requires JRM to install. This is suitable for java supported envirments.Prefer the read.xlsx2() over read.xlsx(), it’s significantly faster for large dataset.

```
require(xlsx)
read.xlsx2("myfile.xlsx", sheetName = "Sheet1")
```

Lab activity : example R script:

```
install.packages("rjava")
install.packages("xlsx")
require(xlsx)
> read.xlsx2("myfile.xlsx", sheetName = "Sheet1")
```

```
Sno   Sname   Marks   Attendance   Contactno      Mailid
1      sri      45        45          988776655     s@mymail.com
2     vas      78        78         435465768     v@mymail.com
3     toni     34        46        -117845119     s@mymail.com
4     mac      90        89        -671156006     v@mymail.com
5     ros       25        23        -1224466893   s@mymail.com
```

xlsReadWrite: Available for Windows only. It’s rather fast but doesn’t support. **.xlsx** files which is a serious drawback. It has been removed from CRAN lately.

read.table(“clipboard”): It allows to copy data from Excel and read it directly in R. This is the quick and dirty R/Excel interaction but it’s very useful in some cases.

```
myDf<- read.table("clipboard")
```

3.3.2. Read R output in Excel:

First create a csv output from an R **data.frame** then read this file in Excel. There is one function that you need to know it’s **write.table**. You might also want to consider: **write.csv** which uses “.” for the decimal point and a comma for the separator and **write.csv2** which uses a comma for the decimal point and a semicolon for the separator.

```
x <- cbind(rnorm(20),runif(20))
colnames(x) <- c("A","B")
write.table(x,"your_path",sep=",",row.names=FALSE)
```

3.3.3. Execute R code in VBA:

RExcel is from my perspective the best suited tool but there is at least one alternative. You can run a batch file within the VBA code. If R.exe is in your PATH, the general syntax for the batch file (.bat)
R CMD BATCH [options] myRScript.R

Here’s an example of how to integrate the batch file above within your VBA code.

3.3.4. Execute R code from an Excel spreadsheet

Rexcel is the only tool I know for the task. Generally speaking once you installed RExcel you insert the excel code within a cell and execute from RExcel spreadsheet menu. See the RExcel references below for an example.

a) Execute VBA code in R

This is something I came across but I never tested it myself. This is a two steps process. First write a VBscript wrapper that calls the VBA code. Second run the VBscript in R with the system or shell functions. The method is described in full details here.

b) Fully integrate R and Excel

RExcel is a project developped by Thomas Baier and Erich Neuwirth, “making R accessible from Excel and allowing to use Excel as a frontend to R”. It allows communication in both directions: Excel to R and R to Excel and covers most of what is described above and more. I’m not going to put any example of RExcel use here as the topic is largely covered elsewhere but I will show you where to find the relevant information. There is a wiki for installing RExcel and an excellent tutorial available here. I also recommend the following two documents: RExcel – Using R from within Excel and High-Level Interface Between R and Excel. They both give an in-depth view of RExcel capabilities.

Introduction to Analytics (Associate Analytics – I)

UNIT IV: Correlation and Regression Analysis (NOS 9001)

4.1. Regression Analysis:

Regression modeling or analysis is a statistical process for estimating the relationships among variables. The main focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors').The value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

The general mathematical equation for a linear regression is −

```
y = mx + c
```

Following is the description of the parameters used −

y is the response variable.

x is the predictor variable.

m(slope) and c(intercept) are constants which are called the coefficients.

In R , lm () function to do simple regression modeling.

The simple linear equation Y=mX+C , intercept "C" and the slope "m" . The below plot shows the linear regression.

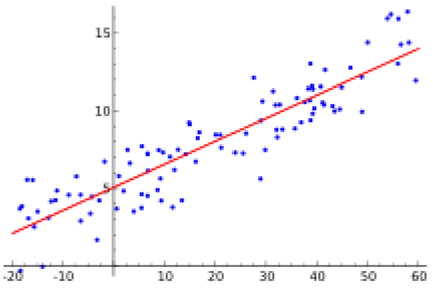


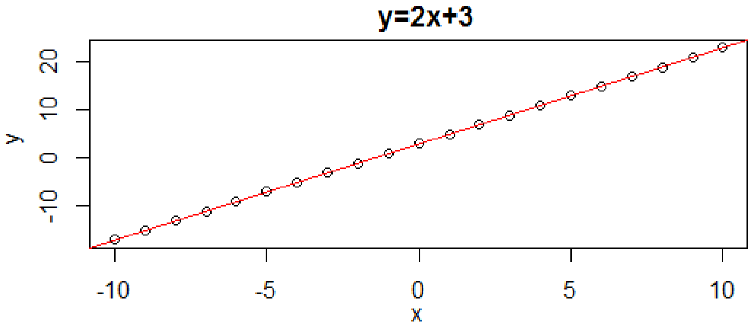
Figure Linear Regression plot

Example:

#Let us consider the equation y=mx+c with the sample values m=2, c=3

#Hence y=2x+3 will takes the following values

```
> x<-c(-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10)
> y<-c(-17,-15,-13,-11,-9,-7,-5,-3,-1,1,3,5,7,9,11,13,15,17,19,21,23)
  # OR You can take y as #  > y<- 2*x+3
> relationxy<-lm(y~x)
> relationxy

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
          3            2
> plot(x,y)
> abline(relationxy,col='red')
> title(main="y=2x+3")
```



Example: (Case study #1)

Lab activity: Linear Regression for finding the relation between petal length and petal width in IRIS dataset:

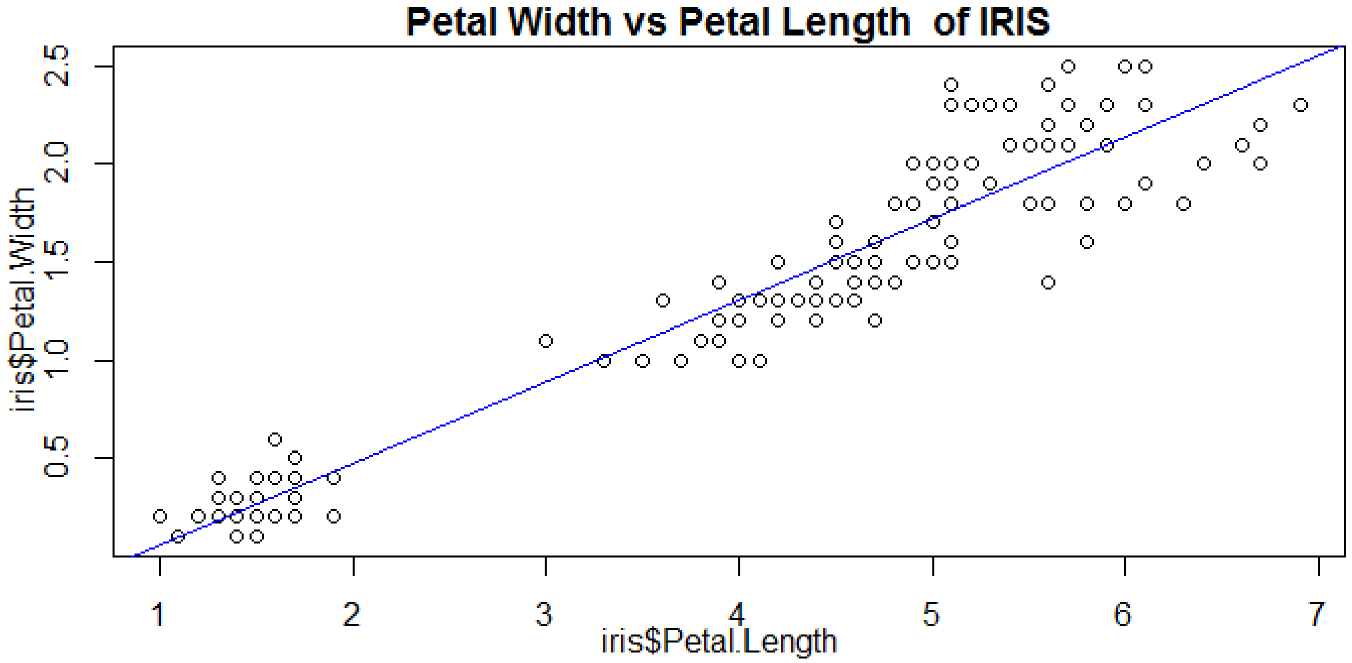
```
> fit <- lm(iris$Petal.Length ~ iris$Petal.Width)
> fit
Call:
lm(formula = iris$Petal.Length ~ iris$Petal.Width)
Coefficients:
     (Intercept)         iris$Petal.Width
           1.084                    2.230
```

We get the intercept “C” and the slope “m” of the equation – Y=mX+C. Here m=2.230 and C=1.084 now we found the linear equation between petal length and petal width is

iris$Petal.Length=2.230* iris$Petal.Width+1.084

We can observe the plot and line below as

```
> plot(iris$Petal.Length,iris$Petal.Width)
> abline(lm(iris$Petal.Width~iris$Petal.Length ),col="blue")
> title(main="Petal Width vs Petal Length  of IRIS")
```



Example: (Case study #2)

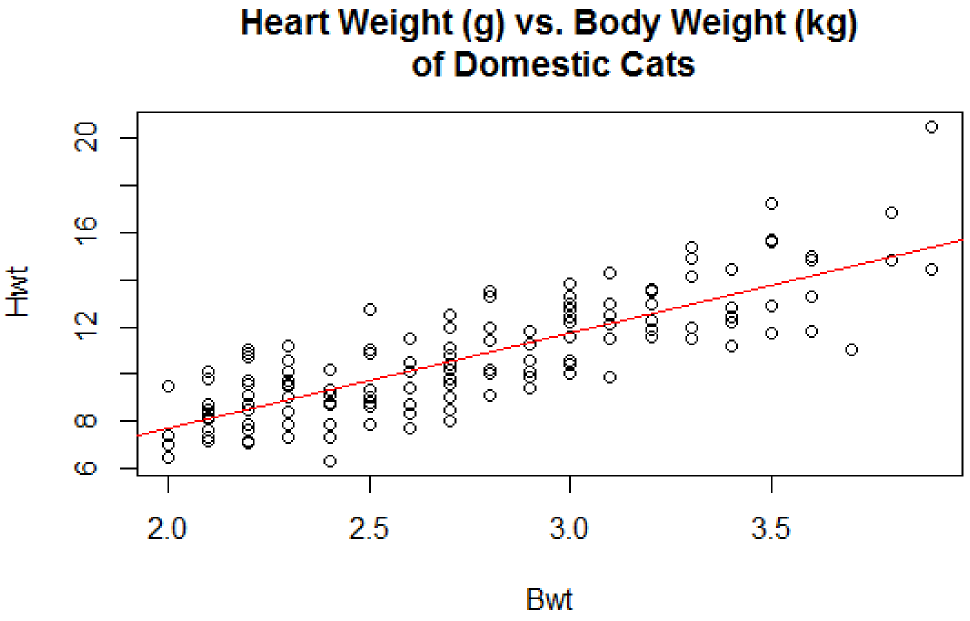
Relation between Heart weight and body weight of cats

- Generate a simple linear regression equation in two variables of ***cats*** dataset. The two variables are Heart Weight and Body Weight of the cats being examined in the research.
- Also find out if there is any relation between Heart Weight and Body Weight.
- Now check if Heart weight is affected by any other factor or variable.
- Find out how Heart Weight is affected by Body Weight and Sex together using Multiple Regression.

```
> library(MASS)   #  MASS Contains cats  #
> data(cats)    # data() was originally intended to allow users to load datasets from packages for use in their examples and
as such it loaded the datasets into the work space. That need has been almost entirely superseded by lazy-loading of datasets.
> str(cats)
      'data.frame':    144 obs. of  3 variables:
       $ Sex: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
       $ Bwt: num  2 2 2 2.1 2.1 2.1 2.1 2.1 2.1 2.1 ...
       $ Hwt: num  7 7.4 9.5 7.2 7.3 7.6 8.1 8.2 8.3 8.5 ...

> summary(cats)
       Sex          Bwt              Hwt
       F:47   Min.   :2.000   Min.   : 6.30
       M:97   1st Qu.:2.300   1st Qu.: 8.95
              Median :2.700   Median :10.10
              Mean   :2.724   Mean   :10.63
              3rd Qu.:3.025   3rd Qu.:12.12
              Max.   :3.900   Max.   :20.50
```

"Bwt" is the body weight in kilograms, "Hwt" is the heart weight in grams, and "Sex" should be obvious. There are no missing values in any of the variables, so we are ready to begin by looking at a scatterplot.

```
> attach(cats)     # This works better rather using data(cats)
> plot(Bwt, Hwt)
> title(main="Heart Weight (g) vs. Body Weight (kg)\n of Domestic Cats")
> lmout<-lm(Hwt~Bwt)
> abline(lmout,col='red')
> lmout
Call:
lm(formula = Hwt ~ Bwt)
Coefficients:
(Intercept)          Bwt
    -0.3567       4.0341
```

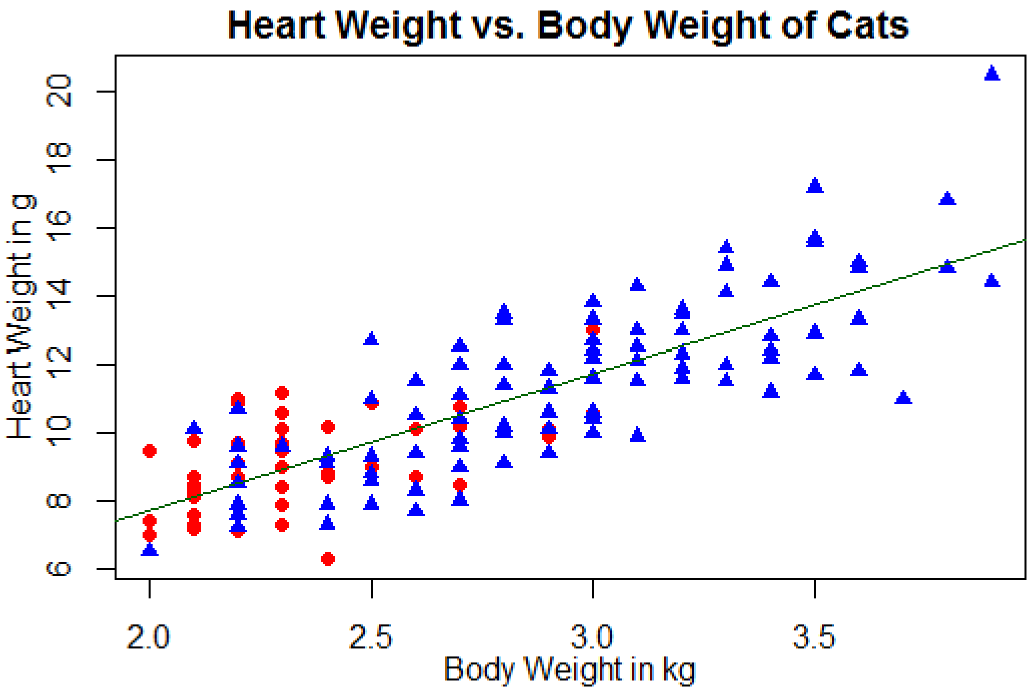


The plot() function gives a scatterplot whenever you feed it two numeric variables. The first variable listed will be plotted on the horizontal axis. A formula interface can also be used, in which case the response variable should come before the tilde and the variable to be plotted on the horizontal axis after. (Close the graphics window before doing this, because the output will look exactly the same.)

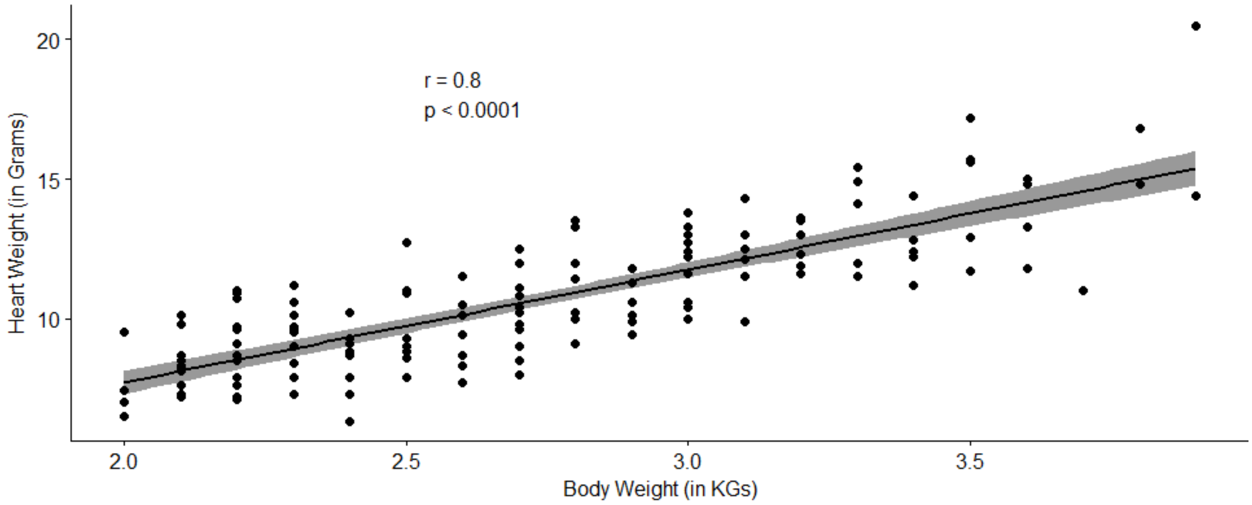
The scatterplot shows a fairly strong and reasonably linear relationship between the two variables. A Pearson product-moment correlation coefficient can be calculated using the cor() function (which will fail if there are missing values).

For a more revealing scatterplot, try this.

```
> with(cats, plot(Bwt, Hwt, type="n", las=.5, xlab="Body Weight in kg",
       ylab="Heart Weight in g",main="Heart Weight vs. Body Weight of Cats"))
> with(cats, points(Bwt[Sex=="F"], Hwt[Sex=="F"], pch=16, col="red"))
> with(cats, points(Bwt[Sex=="M"], Hwt[Sex=="M"], pch=17, col="blue"))
> abline(lm.out,col="dark green")
```



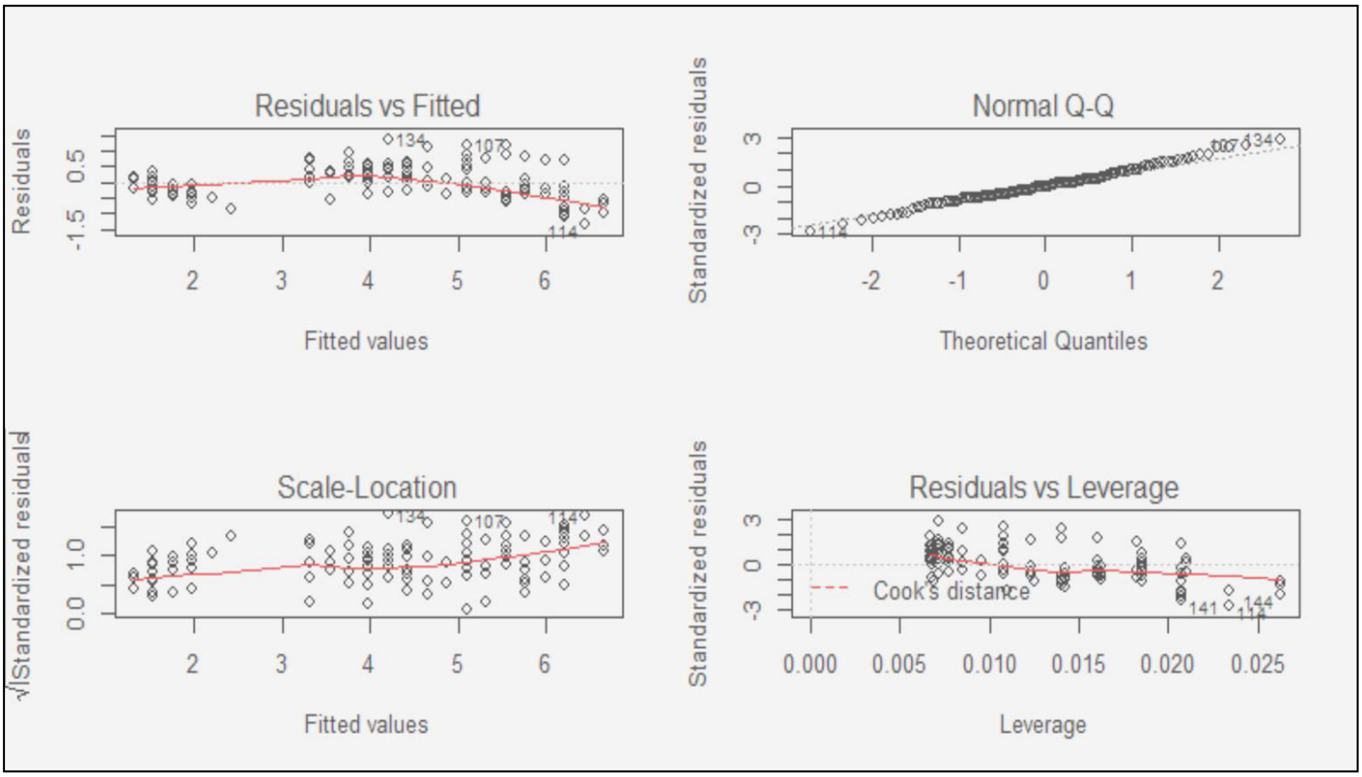
```
#Another Example with additional features
ggscatter(cats, x = "Bwt", y = "Hwt", add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson", xlab = "Body Weight (in KGs)",
          ylab = "Heart Weight (in Grams")
```



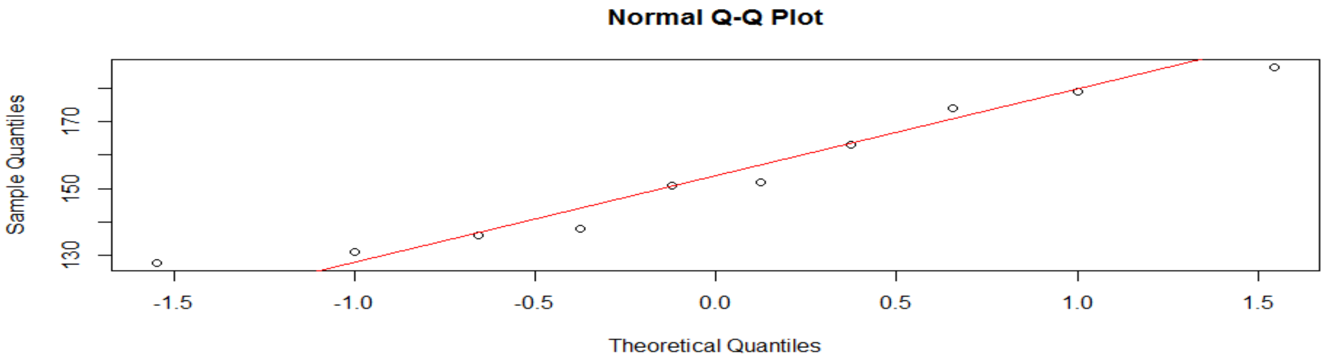
Visualization of fit data:

The fit information displays four charts: Residuals vs. Fitted, Normal Q-Q, Scale-Location, and Residuals vs. Leverage.

Below are the various graphs representing values of regression.



```
# The Normal Q-Q (quantile-quantile Normal Plot) for given vector of data items ca
n be drawn as/:
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131) # with single data set
> qqnorm(x)
> qqline(x,col='red')
```



4.2. Assumptions of OLS Regression

Ordinary least squares (OLS) Method:

Ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model, with the goal of minimizing the differences between the observed responses and the predicted responses by the linear approximation of the data.

Assumptions of regression modeling:

For both simple linear and multiple regressions where the common assumptions are

a) The model is linear in the coefficients of the predictor with an additive random error term

b) The random error terms are

- Normally distributed with 0 mean and
- A variance that doesn't change as the values of the predictor covariates change.

4.3. Regression Modeling:

Regression modeling or analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors').

- **Understand influence of changes in dependent variable:**
  More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables, i.e the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the regression function which can be described by a probability distribution.
- **Estimation of continuous response variables:**
  Regression may refer specifically to the estimation of continuous response variables, as opposed to the discrete response variables used in classification. The case of a continuous output variable may be more specifically referred to as metric regression to distinguish it from related problems.
- **Regression analysis uses:**
  It is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable; for example, correlation does not imply causation.
- **Parametric and non-parametric regression:**
  Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data.
- **Nonparametric regression** refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.
- **Performance of regression analysis :**
  The performance of regression analysis methods in practice depends on the form of the data generating process, and how it relates to the regression approach being used. Since the true form of the data-generating process is generally not known, regression analysis often depends to some extent on making assumptions about this process. These assumptions are sometimes testable if a sufficient quantity of data is available. Regression models for prediction are often useful even when the assumptions are moderately violated, although they may not perform optimally.

4.3.1Regression residuals:

The residual of an observed value is the difference between the observed value and the estimated value of the quantity of interest. Because a linear regression model is not always appropriate for the data, assess the appropriateness of the model by defining residuals and examining residual plots.

Residuals:

The difference between the observed value of the dependent variable (y) and the predicted value ($\hat{y}$) is called the **residual** (e). Each data point has one residual.

Residual = Observed value - Predicted value

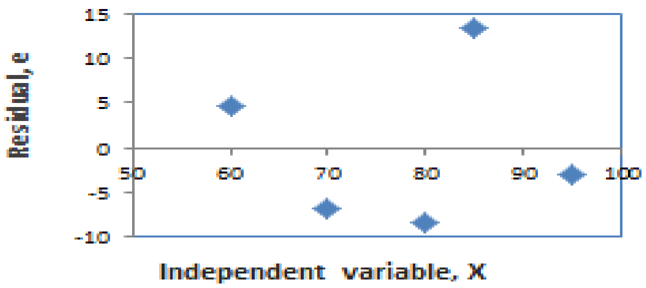
$e = y - \hat{y}$

x	60	70	80	85	95
y	70	65	70	95	85
ŷ	65.411	71.849	78.288	81.507	87.945
e	4.589	-6.849	-8.288	13.493	-2.945

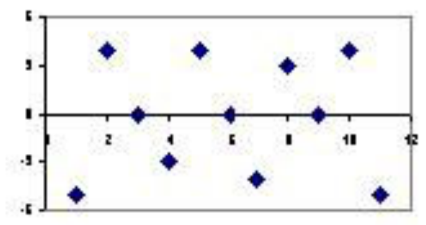
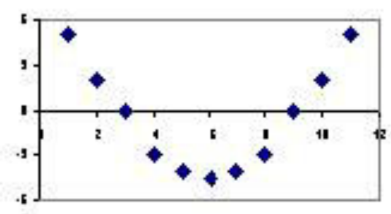
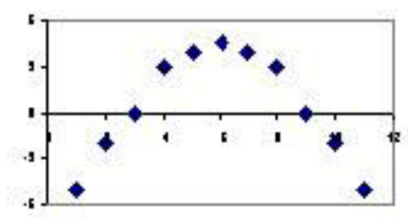
Both the sum and the mean of the residuals are equal to zero. That is, Σ e = 0 and e = 0. The above table shows inputs and outputs from a simple linear regression analysis.

Residual Plots:

A **residual plot** is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.



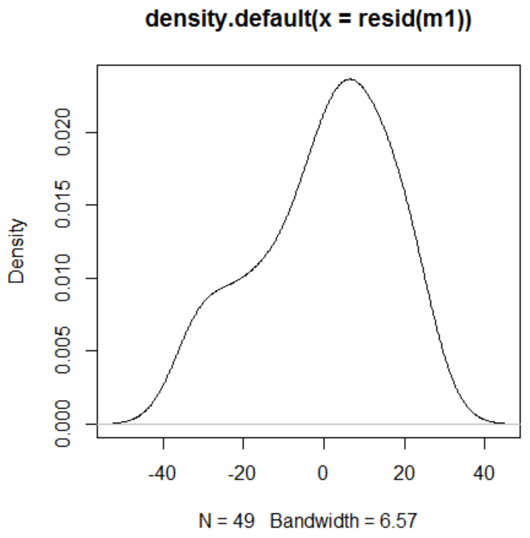
The chart on the right displays the residual (e) and independent variable (X) as a residual plot. The residual plot shows a fairly random pattern - the first residual is positive, the next two are negative, the fourth is positive, and the last residual is negative. This random pattern indicates that a linear model provides a decent fit to the data. Below, the residual plots show three typical patterns. The first plot shows a random pattern, indicating a good fit for a linear model. The other plot patterns are non-random (U-shaped and inverted U), suggesting a better fit for a non-linear model.

		
Random pattern	Non-random: U-shaped	Non-random: Inverted U

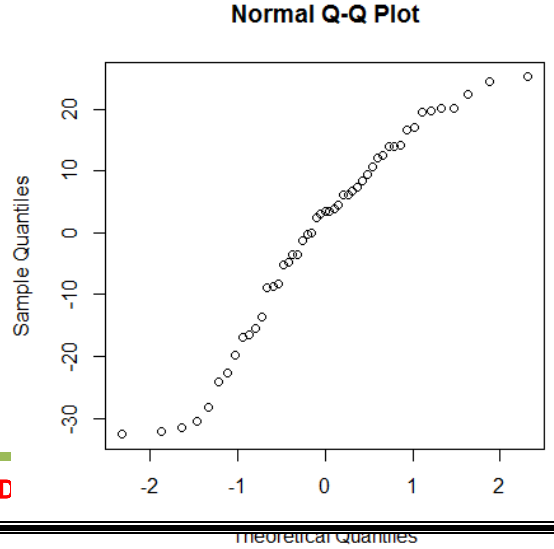
How to find Residuals and plot them?

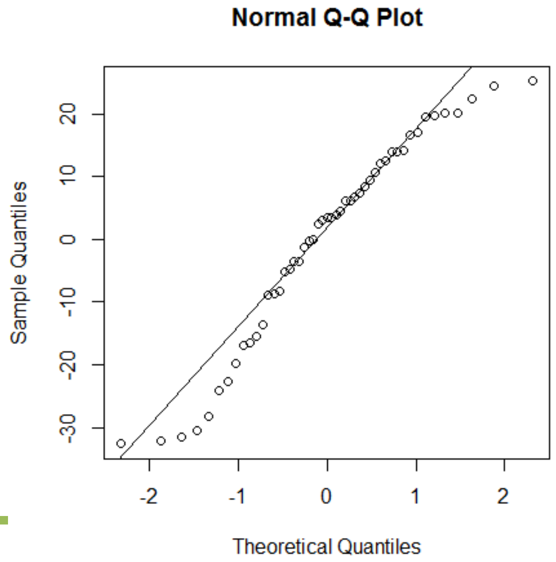
```
##Finding Residuals examples
x=c(21,34,6,47,10,49,23,32,12,16,29,49,28,8,57,9,31,10,21,26,31,52,21,8,18,5,18,26,
    27,26,32,2,59,58,19,14,16,9,23,28,34,70,69,54,39,9,21,54,26)
y = c(47,76,33,78,62,78,33,64,83,67,61,85,46,53,55,71,59,41,82,56,39,89,31,43,
    29,55, 81,82,82,85,59,74,80,88,29,58,71,60,86,91,72,89,80,84,54,71,75,84,79)

m1 <- lm(y~x)  #Create a linear model
resid(m1) #List of residuals
> resid(m1) #List of residuals
```

```
             1            2            3            4            5            6
7            8
-15.56228406   7.44941046 -22.65270081   3.46110498   4.50474366   2.53982722 -30.
48356182  -3.62931177
             9           10           11           12           13           14
15           16
 24.58346589   6.74091036  -5.24739512   9.53982722 -19.78675624  -3.57397858 -24.
14528385  13.96538254
            17           18           19           20           21           22
23           24
 -8.16867289 -16.49525634  19.43771594  -8.86547847 -28.16867289  12.15791057 -31.
56228406 -13.57397858
            25           26           27           28           29           30
31           32
-32.18036741  -0.19206193  19.81963259  17.13452153  16.67388265  20.13452153  -8.
62931177  20.18985472
            33           34           35           36           37           38
39           40
 -0.06656161   8.39407727 -32.64100629  -1.33781187  10.74091036   2.96538254  22.
51643818  25.21324376
            41           42           43           44           45           46
47           48
  3.44941046   3.86641067  -4.67295044   6.23663280 -16.85378395  13.96538254  12.
43771594   6.23663280
            49
 14.13452153
> plot(density(resid(m1))) #A density plot
```



```
> qqnorm(resid(m1)) # A quantile normal plot -
good for checking normality
> qqline(resid(m1))
```





4.4.Correlation:

The Correlation is a measure of association between two variables. Correlations are Positive and negative which are ranging between +1 and -1.

Positive correlation (0 to +1) example: Earning and expenditure

Negative correlation (-1 to 0) example : Speed and time

In R , correlation between x and y is by using cor(x,y) function.

Example:

```
> cor(iris$Petal.Length,iris$Petal.Width)
[1] 0.9628654
```

Means Petal Length and Petal Width are very strongly correlated.

Correlation Coefficient: correlation coefficient is indicated with r. The r value

- ✓ +1 : Perfectly positive
- ✓ -1 : Perfectly negative
- ✓ 0 – 0.2 : No or very weak association
- ✓ 0.2 – 0.4 : Weak association
- ✓ 0.4 – 0.6 : Moderate association
- ✓ 0.6 – 0.8 : Strong association
- ✓ 0.8 – 1.0 : Very strong to perfect association

```
> cor(Petal.Length, Petal.Width)
       Error in is.data.frame(y) : object 'Petal.Width' not found
> attach(iris)
> cor(Petal.Length, Petal.Width)
       [1] 0.9628654
> cor(Sepal.Length,Sepal.Width)
       [1] -0.1175698
> cor(Petal.Length,Sepal.Length)
       [1] 0.8717538
> cor(Petal.Length,Sepal.Width)
       [1] -0.4284401
> cor(Petal.Width,Sepal.Length)
       [1] 0.8179411
```

```
> iris1<-iris[1:4]
> iris1
   Sepal.Length Sepal.Width Petal.Length Petal.Width
1           5.1         3.5          1.4         0.2
2           4.9         3.0          1.4         0.2
3           4.7         3.2          1.3         0.2
4           4.6         3.1          1.5         0.2
5           5.0         3.6          1.4         0.2
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
.
.
.
```

```
[CONTD …]     # we have a total of 150 records in iris1, observe in iris 1 only first four columns are considered, and species is left out to consider numeric data
> cor(iris1)
                 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.test(Petal.Length,Petal.Width)
        Pearson's product-moment correlation
      data:  Petal.Length and Petal.Width
      t = 43.387, df = 148, p-value < 2.2e-16
      alternative hypothesis: true correlation is not equal to 0
      95 percent confidence interval:
       0.9490525 0.9729853
      sample estimates:
            cor
      0.9628654    # which is the same value with cor() computation

> with(cats, cor.test(Bwt, Hwt))       # cor.test(~Bwt + Hwt, data=cats) also works
    Pearson's product-moment correlation    #^ Know More at the end of this unit..
      data:  Bwt and Hwt
      t = 16.1194, df = 142, p-value < 2.2e-16
      alternative hypothesis: true correlation is not equal to 0
      95 percent confidence interval:
       0.7375682 0.8552122
      sample estimates:
            cor
      0.8041274
```

Output Explanation:

The first line Pearson’s Product-Moment Correlation (PPMC) (or Pearson correlation coefficient, for short) is a measure of the strength of a linear association between two variables and is denoted by *r*. Basically, a Pearson product-moment correlation attempts to draw a line of best fit through the data of two variables, and the Pearson correlation coefficient, *r*, indicates how far away all these data points are to this line of best fit (i.e., how well the data points fit this new model/line of best fit).

The Second Line the information about the data that we have considered for the correlation test.

On third line, you have the t-test for H0(Hypothesis Test HO & HA): As you can see, t is very large and p-value is very, very tiny, so you can reject the nil and be (almost) sure that the correlation on the population is not zero. P-value is just a yes/no kind of thing. Lower p-value does not mean stronger correlation. If p-value is <0.05 we can be reasonably certain that the two data columns are correlated, but a p-value of 0.01 does not mean the data is more strongly correlated than a set with p-value of 0.03. The t-statistic is used to calculate the p-value. The p value that's greater than 0.05 failed to reject the alternative hypothesis and the conclusion is that there is no significant correlation, namely accept the null hypothesis where correlation is equal to zero. Forth is the statement derived from alternative hypothesis test.

The fifth & sixth lines have very important information that the confidence interval at 95% for the correlation on the population. You can see that is between 0.73 and 0.86, so you can be (almost) sure that according to Cohen, you have a strong correlation on your population.

The last line is the ***correlation coefficient*** value ‘r’ obtained from the test.

```
> with(cats, cor.test(Bwt, Hwt, alternative="greater", conf.level=.8))

              Pearson's product-moment correlation

      data:  Bwt and Hwt
      t = 16.1194, df = 142, p-value < 2.2e-16
      alternative hypothesis: true correlation is greater than 0
      80 percent confidence interval:
       0.7776141 1.0000000
      sample estimates:
```

```
      cor
0.8041274
```

There is also a formula interface for cor.test(), but it's tricky. Both variables should be listed after the tilde. (Order of the variables doesn't matter in either the cor() or the cor.test() function.)

Covariance

The **covariance** of two variables *x* and *y* in a data set measures how the two are linearly related. A positive covariance would indicate a positive linear relationship between the variables, and a negative covariance would indicate the opposite.

The **sample covariance** is defined in terms of the sample means as:

$$\text{Covariance is } \sigma_{x,y} = \frac{\sum(x-\overline{x})(y-\overline{y})}{N}$$

$$\text{Correlation Coefficient is } \rho_{x,y} = \frac{\sigma_{x,y}}{\sigma_x \sigma_y} \, where -1 \le \rho \le +1$$

In R, the covariance of x and y is by using cov(x, y) function.

Example: Covariance of petal length and petal width of iris dataset.

> cov(iris$Petal.Length,iris$Petal.Width)

[1] 1.295609

It means between Petal Length and Petal Width, the positive linear relationship existed.

Problem

Find the covariance of eruption duration and waiting time in the data set faithful. Observe if there is any linear relationship between the two variables.

Solution

We apply the cov function to compute the covariance of eruptions and waiting.

```
> duration = faithful$eruptions    # eruption durations
> waiting = faithful$waiting       # the waiting period
> cov(duration, waiting)           # apply the cov function
[1] 13.978
```

Answer

The covariance of eruption duration and waiting time is about 14. It indicates a positive linear relationship between the two variables.

The sample correlation coefficient is defined by the following formula, where sx and sy are the sample standard deviations, and cov(x,y) is the sample covariance.

$$r = \mathrm{cov}(x, y)\frac{\mathrm{cov}(x, y)}{\sqrt{S_x^2 S_y^2}}$$

4.5. ANOVA:

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences among group means and their associated procedures (such as "variation" among and between groups).

In the ANOVA setting, the observed variance in a particular variable is partitioned into components attributable to different sources of variation.

In its simplest form, ANOVA provides a statistical test of whether or not the means of several groups are equal, and therefore generalizes the t-test to more than two groups.

ANOVAs are useful for comparing (testing) three or more means (groups or variables) for statistical significance.

In R, to perform ANOVA test the built in function is anova()

Compute analysis of variance (or deviance) tables for one or more fitted model objects.

anova(object, ...)

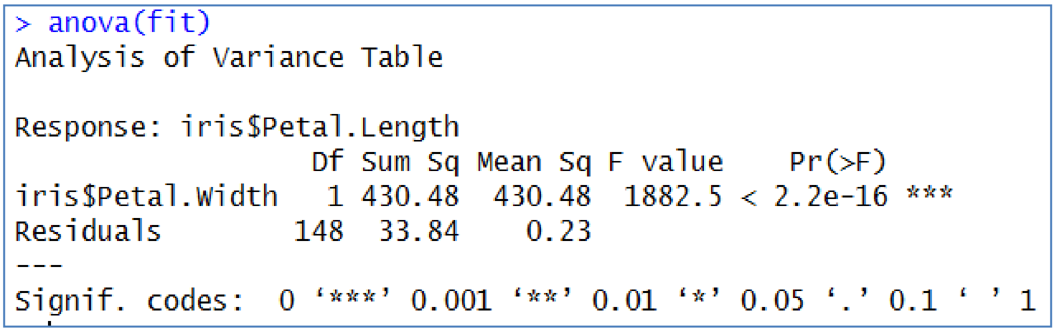
Object : An object containing the results returned by a model fitting function (e.g., lm or glm).

Return value: an object of class anova includes:

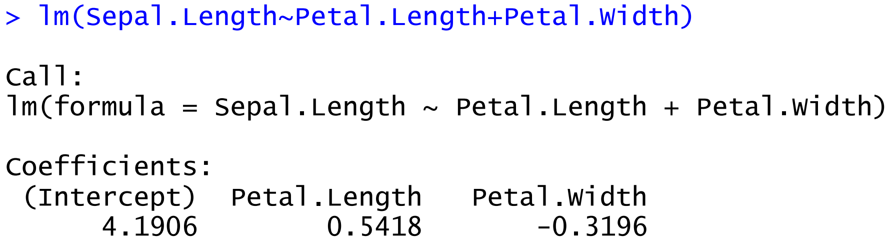
Analysis-of-variance

Analysis-of-deviance tables.

Warning: The comparison between two or more models will only be valid if they are fitted to the same dataset. This may be a problem if there are missing values and R's default of na.action = na.omit is used.



```
> anova(fit)
Analysis of Variance Table

Response: iris$Petal.Length
                  Df Sum Sq Mean Sq F value    Pr(>F)    
iris$Petal.Width   1 430.48  430.48  1882.5 < 2.2e-16 ***
Residuals        148  33.84    0.23                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> lm(Sepal.Length~Petal.Length+Petal.Width)

Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width)

Coefficients:
 (Intercept)  Petal.Length   Petal.Width  
      4.1906        0.5418       -0.3196  
```

4.6. Forecasting

Forecasting is the process of making predictions of the future based on past and present data and analysis of trends. A commonplace example might be estimation of some variable of interest at some specified future date.

Usage can differ between areas of application:

Example:

In hydrology, the terms "forecast" and "forecasting" are sometimes reserved for estimates of values at certain specific future times, The term "prediction" is used for more general estimates, such as the number of times floods will occur over a long period.

4.7. Heteroscedasticity:

A collection of random variables is heteroscedastic, if there are sub-populations that have different variability’s from others. Here "variability" could be quantified by the variance or any other measure of statistical dispersion. Thus heteroscedasticity is the absence of homoscedasticity.

The existence of heteroscedasticity is a major concern in the application of regression analysis, including the analysis of variance, as it can invalidate statistical tests of significance that assume that the modeling errors are uncorrelated and uniform—hence that their variances do not vary with the effects being modeled. For instance, while the ordinary least squares estimator is still unbiased in the presence of heteroscedasticity, it is inefficient because the true variance and covariance are underestimated. Similarly, in testing for differences between sub-populations using a location test, some standard tests assume that variances within groups are equal.

4.8. Autocorrelation:

Autocorrelation, also known as serial correlation or cross-autocorrelation, is the cross-correlation of a signal with itself at different points in time. It is the similarity between observations as a function of the time lag between them. It is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal obscured by noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. It is often used in signal processing for analyzing functions or series of values, such as time domain signals. In statistics, the autocorrelation of a random process describes the correlation between values of the process at different times, as a function of the two times or of the time lag.

Multicollinearity (also collinearity) is a phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, meaning that one can be linearly predicted from the others with a substantial degree of accuracy.

4.9.Introduction to Multiple Regression:

The multiple regression is the relationship between several independent or predictor variables and a dependent or criterion variable.

For example: A real estate agent might record for each listing the size of the house (in square feet), the number of bedrooms, the average income in the respective neighborhood according to census data, and a subjective rating of appeal of the house. Once this information has been compiled for various houses it would be interesting to see whether and how these measures relate to the price for which a house is sold. For example, you might learn that the number of bedrooms is a better predictor of the price for which a house sells in a particular neighborhood than how "pretty" the house is (subjective rating).

Lab Activity:

Multiple Regression model on Iris Data set

Step1: Subset the numeric data from iris dataset

Step2: Find the correlation among all variables

Step3: Find the formula based on highly correlated variables

Step4: call the glm()

```
>iris1<-iris[1:4]
> cor(iris1)
             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
```

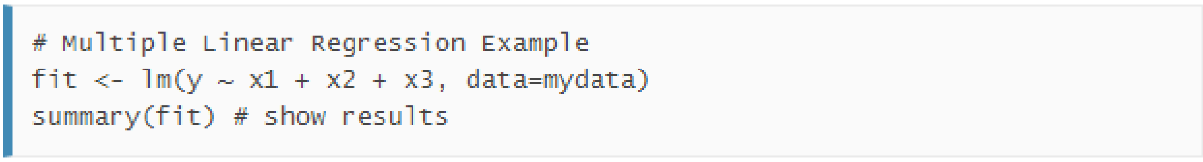
Hence (sepal length, petal length),(petal width, sepal length) and (petal length ,petal width) are showing high correlation.

```
> glm(formula=iris$Petal.Length~iris$Petal.Width+iris$Sepal.Length)

Call:  glm(formula = iris$Petal.Length ~ iris$Petal.Width + iris$Sepal.Length)

Coefficients:
      (Intercept)   iris$Petal.Width  iris$Sepal.Length
          -1.5071             1.7481             0.5423

Degrees of Freedom: 149 Total (i.e. Null); 147 Residual
Null Deviance:          464.3
Residual Deviance: 23.9     AIC: 158.2
```

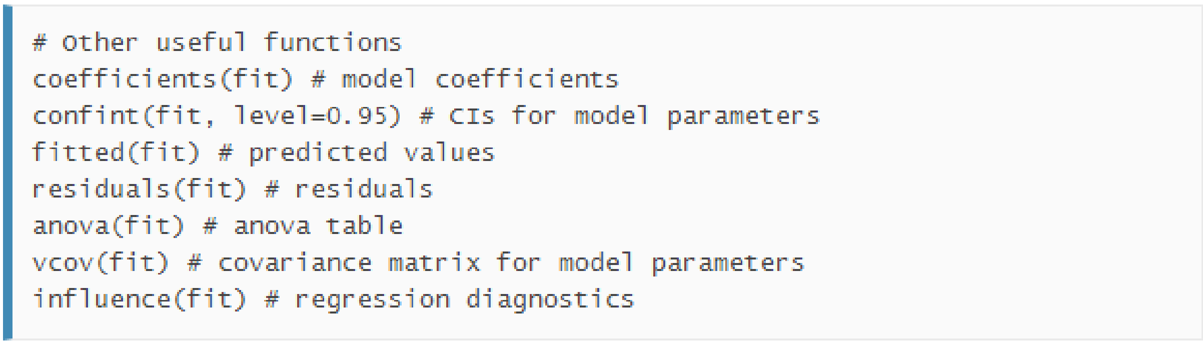
Hence the formula found from model is

iris$Petal.Length = 1.7481*iris$Petal.Width + 0.5423*iris$Sepal.Length-1.5071

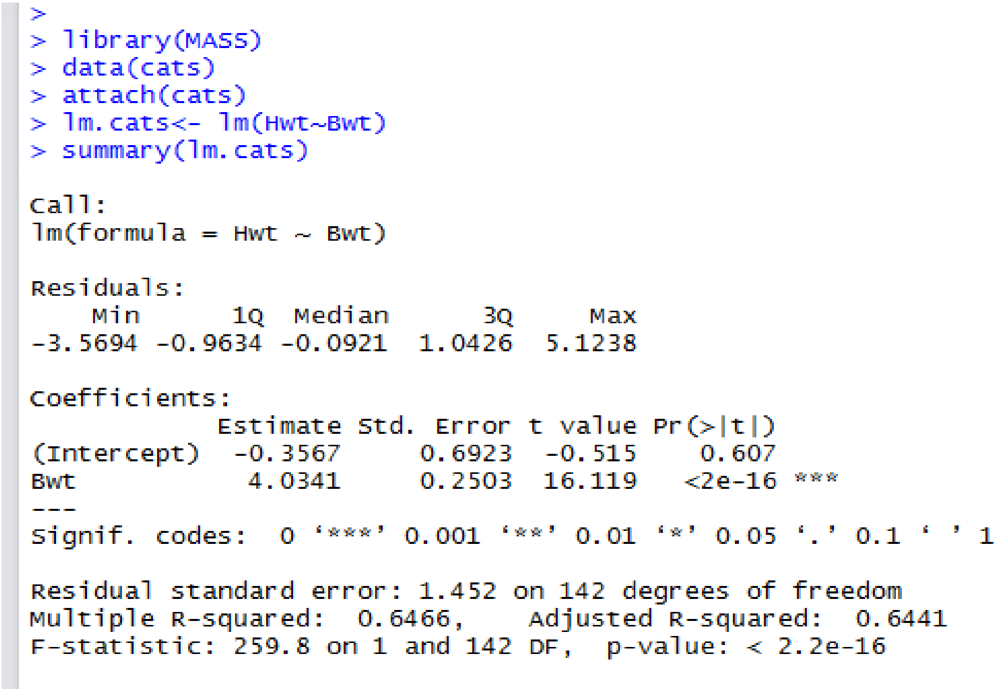
study2:Multiple linear regression on MS application data:



```
# Multiple Linear Regression Example
fit <- lm(y ~ x1 + x2 + x3, data=mydata)
summary(fit) # show results
```



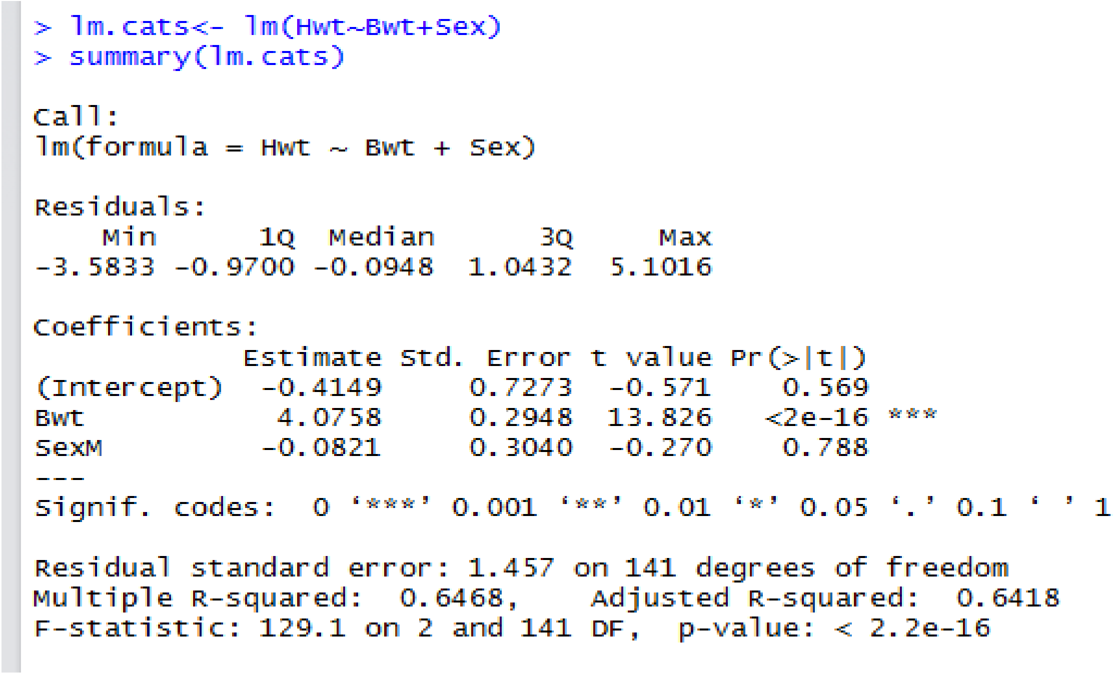
```
# Other useful functions
coefficients(fit) # model coefficients
confint(fit, level=0.95) # CIs for model parameters
fitted(fit) # predicted values
residuals(fit) # residuals
anova(fit) # anova table
vcov(fit) # covariance matrix for model parameters
influence(fit) # regression diagnostics
```



```
>
> library(MASS)
> data(cats)
> attach(cats)
> lm.cats<- lm(Hwt~Bwt)
> summary(lm.cats)

Call:
lm(formula = Hwt ~ Bwt)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5694 -0.9634 -0.0921  1.0426  5.1238 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.3567     0.6923  -0.515    0.607    
Bwt           4.0341     0.2503  16.119   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.452 on 142 degrees of freedom
Multiple R-squared:  0.6466,	Adjusted R-squared:  0.6441 
F-statistic: 259.8 on 1 and 142 DF,  p-value: < 2.2e-16
```

P Value is less than 0.05 which means we reject null hypothesis. Degree of Freedom is 142.
For other examples use the link: http://www.ats.ucla.edu/stat/r/dae/rreg.htm
Also refer to the book: - Practical Regression and Anova using R
Now to create a linear model of effect of Body Weight and Sex on Heart Weight we use multiple regression modeling.

Case study 3: Multiple linear regression on cats dataset:



```
> lm.cats<- lm(Hwt~Bwt+Sex)
> summary(lm.cats)

Call:
lm(formula = Hwt ~ Bwt + Sex)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5833 -0.9700 -0.0948  1.0432  5.1016 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.4149     0.7273  -0.571    0.569    
Bwt           4.0758     0.2948  13.826   <2e-16 ***
SexM         -0.0821     0.3040  -0.270    0.788    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.457 on 141 degrees of freedom
Multiple R-squared:  0.6468,	Adjusted R-squared:  0.6418 
F-statistic: 129.1 on 2 and 141 DF,  p-value: < 2.2e-16
```

So we can say that 65% variation in Heart Weight can be explained by the model.

The equation becomes y=4.07x-0.08y-0.41

Dummy Variables:

In regression analysis, a dummy variable (also known as an indicator variable, design variable, Boolean indicator, categorical variable, binary variable, or qualitative variable) is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome. Dummy variables are used as devices to sort data into mutually exclusive categories (such as smoker/non-smoker, etc.).

In other words, Dummy variables are "proxy" variables or numeric stand-ins for qualitative facts in a regression model. In regression analysis, the dependent variables may be influenced not only by quantitative variables (income, output, prices, etc.), but also by qualitative variables (gender, religion, geographic region, etc.). A dummy independent variable (also called a dummy explanatory variable) which for some observation has a value of 0 will cause that variable's coefficient to have no role in influencing the dependent variable, while when the dummy takes on a value 1 its coefficient acts to alter the intercept.

Example:

Suppose Gender is one of the qualitative variables relevant to a regression. Then, female and male would be the categories included under the Gender variable. If female is arbitrarily assigned the value of 1, then male would get the value 0. Then the intercept (the value of the dependent variable if all other explanatory variables hypothetically took on the value zero) would be the constant term for males but would be the constant term plus the coefficient of the gender dummy in the case of females.

***** End of Unit-4 *****

Pearson's product-moment correlation

What is Pearson Correlation?

Correlation between sets of data is a measure of how well they are related. The most common measure of correlation in stats is the Pearson Correlation. The full name is the Pearson Product Moment Correlation or PPMC. It shows the linear relationship between two sets of data. In simple terms, it answers the question, Can I draw a line graph to represent the data? Two letters are used to represent the Pearson correlation: Greek letter rho (ρ) for a population and the letter "r" for a sample.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

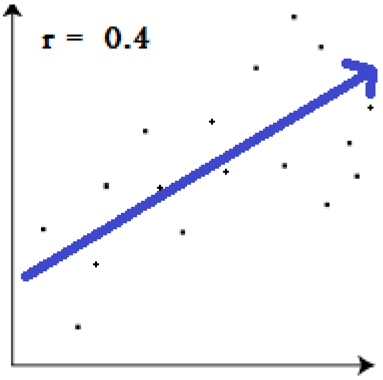
What are the Possible Values for the Pearson Correlation?

The results will be between -1 and 1. You will very rarely see 0, -1 or 1. You'll get a number somewhere in between those values. The closer the value of r gets to zero, the greater the variation the data points are around the line of best fit.

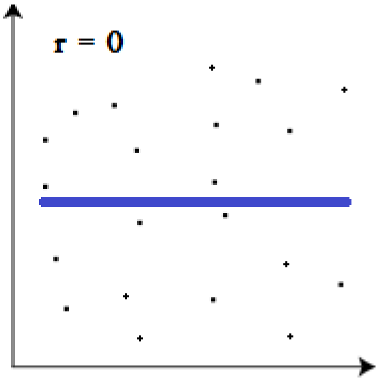
High correlation: .5 to 1.0 or -0.5 to 1.0.

Medium correlation: .3 to .5 or -0.3 to .5.

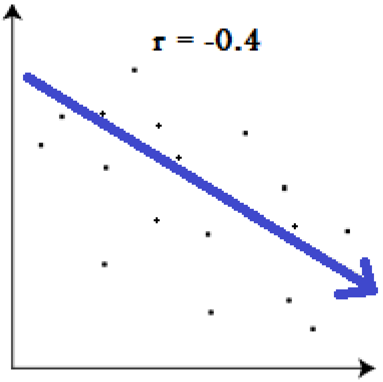
Low correlation: .1 to .3 or -0.1 to -0.3.



Positive Correlation



No correlation



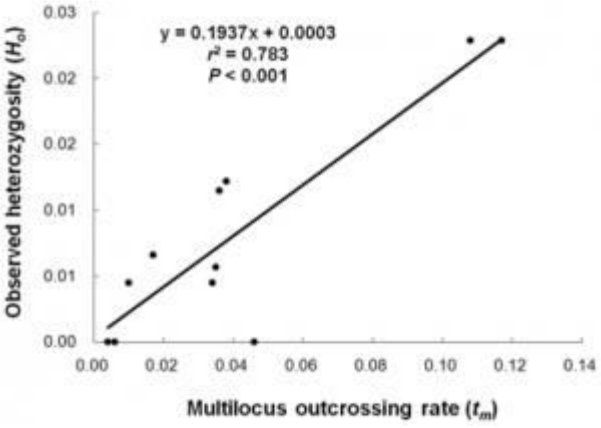
Negative

Potential problems with Pearson correlation.

The PPMC is not able to tell the difference between dependent and independent variables. For example, if you are trying to find the correlation between a high calorie diet and diabetes, you might find a high correlation of .8. However, you could also work out the correlation coefficient formula with the variables switched around. In other words, you could say that diabetes causes a high calorie diet. That obviously makes no sense. Therefore, as a researcher you have to be aware of the data you are plugging in. In addition, the PPMC will not give you any information about the slope of the line; It only tells you whether there is a relationship.

Real Life Example

Pearson correlation is used in thousands of real life situations. For example, scientists in China wanted to know if there was a relationship between how weedy rice populations are different genetically. The goal was to find out the evolutionary potential of the rice. Pearson's correlation between the two groups was analyzed. It showed a positive Pearson Product Moment correlation of between 0.783 and 0.895 for weedy rice populations. This figure is quite high, which suggested a fairly strong relationship.



If you're interested in seeing more examples of PPMC, you can find several studies on the National Institute of Health's Openi website, which shows result on studies as varied as breast cyst imaging to the role that carbohydrates play in weight loss.

Correlation Coefficient

The **correlation coefficient** of two variables in a data set equals to their covariance divided by the product of their individual standard deviations. It is a normalized measurement of how the two are linearly related.

Formally, the **sample correlation coefficient** is defined by the following formula, where s_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

Similarly, the **population correlation coefficient** is defined as follows, where σ_x and σ_y are the population standard deviations, and σ_{xy} is the population covariance.

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

If the correlation coefficient is close to 1, it would indicate that the variables are positively linearly related and the scatter plot falls almost along a straight line with positive slope. For -1, it indicates that the variables are negatively linearly related and the scatter plot almost falls along a straight line with negative slope. And for zero, it would indicate a weak linear relationship between the variables.

Problem

Find the correlation coefficient of eruption duration and waiting time in the data set faithful. Observe if there is any linear relationship between the variables.

Solution

We apply the cor function to compute the correlation coefficient of eruptions and waiting.

```
> duration = faithful$eruptions    # eruption durations
> waiting = faithful$waiting       # the waiting period
> cor(duration, waiting)           # apply the cor function
[1] 0.90081
```

Answer

The correlation coefficient of eruption duration and waiting time is 0.90081. Since it is rather close to 1, we can conclude that the variables are positively linearly related.

Introduction to Analytics (Associate Analytics – I)

UNIT V

Understand the Verticals - Engineering, Financial and others (NOS 9002)

Understand the Verticals _ Engineering, Financial and others (NOS 9002)

Understanding systems viz. Engineering Design, Manufacturing, Smart Utilities, Production lines, Automotive, Technology etc. Understanding Business problems related to various businesses.

Requirements Gathering

Gathering all the data related to Business objective

S. No	Content
5.1	Understanding systems viz. Engineering Design, Manufacturing, Smart Utilities, Production lines, Automotive, Technology etc
5.2	Understanding Business problems related to various businesses.
5.3	Requirements Gathering Gathering all the data related to Business objective

5.1.1 Engineering Design:

The engineering design process is a series of steps that engineers follow to come up with a solution to a problem. Many times the solution involves designing a product (like a machine or computer code) that meets certain criteria and/or accomplishes a certain task.

The engineering design process is a methodical series of steps that engineers use in creating functional products and processes. The process is highly iterative - parts of the process often need to be repeated many times before production phase can be entered - though the part(s) that get iterated and the number of such cycles in any given project can be highly variable.

Engineering Design Process describes the following stages:

1) Research
2) Conceptualization
3) Feasibility assessment
4) Establishing Design requirements
5) Preliminary design
6) Detailed design
7) Production planning and tool design, and
8) Production.

1. Research:

Research is a careful and detailed study into a specific problem, concern, or issue using the scientific method Research can be about anything, and we hear about all different types of research in the news. Cancer research has 'Breakthrough Cancer-Killing Treatment Has No Side Effects in Mice,' and 'Baby Born with HIV Cured.' Each of these began with an issue or a problem (such as cancer or HIV), and they had a question, like, 'Does medication X reduce cancerous tissue or HIV infections?'

But all I've said so far is what research has done (sort of like saying baking leads to apple pie; it doesn't really tell you anything other than the two are connected). To begin researching something, you have to have a problem, concern, or issue that has turned into a question. These can come from observing the world, prior research, professional literature, or from peers. Research really begins with the right question, because your question must be answerable. Questions like, 'How can I cure cancer?' aren't really answerable with a study. It's too vague and not testable.

3. Conceptualization:

Conceptualization is mental process of organizing one"s observations and experiences into meaningful and coherent wholes.

In research, conceptualization produces an agreed upon meaning for a concept for the purposes of research. Different meaning for a concept for the purposes of research. Different researchers may conceptualize a concept slightly differently. „ Conceptualization describes the indicators we'll use to measure Conceptualization describes the indicators we'll use to measure the concept and the different aspects of the concept.

3. Feasibility assessment:

Feasibility studies are almost always conducted where large sums are at stake. Also called feasibility analysis. In order to ensure the manufacturing facility to make a new item the engineers launched a feasibility study to determine the actual steps required to build the product.

4. Design Requirements :

The product/component to be analysed is characterised in terms of: functional requirements, objective of the materials selection process, constraints imposed by the requirements of the application, plus the free variable, which is usually one of the geometric dimensions of the product/component, such as thickness, which enables the constraints to be satisfied and the objective function to be maximised or minimised, depending on the application. Hence, the design requirement of the part/component is defined in terms of function, objective, constraints

5. Preliminary Design:

The preliminary design, or high-level design includes , often bridges a gap between design conception and detailed design, particularly in cases where the level of conceptualization achieved during ideation is not sufficient for full evaluation. So in this task, the overall system configuration is defined, and schematics, diagrams, and layouts of the project may provide early project configuration. This notably varies a lot by field, industry, and product. During detailed design and optimization, the parameters of the part being created will change, but the preliminary design focuses on creating the general framework to build the project on.

6. Detailed Design:

Detailed Design phase, which may consist of procurement of materials as well. This phase further elaborates each aspect of the project/product by complete description through solid modelling, drawings as well as specifications.

7. Production planning and tool design:

The production planning and tool design consists of planning how to mass-produce the product and which tools should be used in the manufacturing process. Tasks to complete in this step include selecting materials, selection of the production processes, determination of the sequence of operations, and selection of tools such as jigs, fixtures, metal cutting and metal or plastics forming tools. This task also involves additional prototype testing iterations to ensure the mass-produced version meets qualification testing standards.

8. Production:

Production is a process of workers combining various material inputs and immaterial inputs (plans, know-how) in order to make something for consumption (the output). It is the act of creating output, a good or service which has value and contributes to the utility of individuals.

5.1.2. Manufacturing:

Manufacturing is the production of goods for use or sale using labour and machines, tools, chemical and biological processing, or formulation. The term may refer to a range of human activity, from handicraft to high tech, but is most commonly applied to industrial production, in which raw materials are transformed into finished goods on a large scale. Such finished goods may be used for manufacturing other, more complex products, such as aircraft, household appliances or automobiles, or sold to wholesalers, who in turn sell them to retailers, who then sell them to end users – the "consumers".

Manufacturing takes turns under all types of economic systems. In a free market economy, manufacturing is usually directed toward the mass production of products for sale to consumers at a profit. In a collectivist economy, manufacturing is more frequently directed by the state to supply a centrally planned economy. In mixed market economies, manufacturing occurs under some degree of government regulation.

Modern manufacturing includes all intermediate processes required for the production and integration of a product's components. Some industries, such as semiconductor and steel manufacturers use the term fabrication instead.

The manufacturing sector is closely connected with engineering and industrial design. Examples of major manufacturers in North America include General Motors Corporation, General Electric, Procter & Gamble, General Dynamics, Boeing, Pfizer, and Precision Cast parts. Examples in Europe include Volkswagen Group, Siemens, and Michelin. Examples in Asia include Sony, Huawei, Lenovo, Toyota, Samsung, and Bridgestone.

5.1.3 Smart Utilities:

S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology; often written as SMART) is a monitoring system included in computer hard disk drives (HDDs) and solid-state drives (SSDs) that detects and reports on various indicators of drive reliability, with the intent of enabling the expectation of hardware failures.

When S.M.A.R.T. data indicates a possible forthcoming drive failure, software running on the host system may notify the user so stored data can be copied to another storage device, preventing data loss, and the failing drive can be replaced.

Smart Utility Systems is the leading provider of Software-as-a-Service (SaaS) solutions for Customer Engagement, Mobile Workforce, and Big Data Analytics to the Energy and Utility sector. We help utilities improve their operational efficiency and maximize revenue realization, through mobile and cloud technologies.

5.1.4 Production lines:

Production lines is an arrangement in a factory in which a thing being manufactured is passed through a set linear sequence of mechanical or manual operations. A production line is a set of sequential operations established in a factory whereby materials are put through a refining process to produce an end-product that is suitable for onward consumption; or components are assembled to make a finished article.

5.1.5 Automotive:

The automotive industry is a wide range of companies and organizations involved in the design, development, manufacturing, marketing, and selling of motor vehicles,[1] some of them are called automakers. It is one of the world's most important economic sectors by revenue. The automotive industry does not include industries will be dedicated to the maintenance of automobiles following delivery to the end-user (maybe), such as automobile repair shops and motor fuel filling stations.

5.1.6 Technology:

The application of scientific knowledge for practical purposes, especially in industry. Technology can be the knowledge of techniques, processes, and the like, or it can be embedded in machines which can be operated without detailed knowledge of their workings. The human species' use of technology began with the conversion of natural resources into simple tools. The prehistoric discovery of how to control fire and the later Neolithic Revolution increased the available sources of food and the invention of the wheel helped humans to travel in and control their environment. Developments in historic times, including the printing press, the telephone, and the Internet, have lessened physical barriers to communication and allowed humans to interact freely on a global scale. The steady progress of military technology has brought weapons of ever-increasing destructive power, from clubs to nuclear weapons.

Technology has many effects. It has helped develop more advanced economies (including today's global economy) and has allowed the rise of a leisure class. Many technological processes produce unwanted by-products known as pollution and deplete natural resources to the detriment of Earth's environment. Various implementations of technology influence the values of a society and new technology often raises new ethical questions. Examples include the rise of the notion of efficiency in terms of human productivity, and the challenges of bioethics.

Philosophical debates have arisen over the use of technology, with disagreements over whether technology improves the human condition or worsens it. Neo-Luddism, anarcho-primitivism, and similar reactionary movements criticise the pervasiveness of technology in the modern world, arguing that it harms the environment and alienates people; proponents of ideologies such as transhumanism and techno-progressivism view continued technological progress as beneficial to society and the human condition.

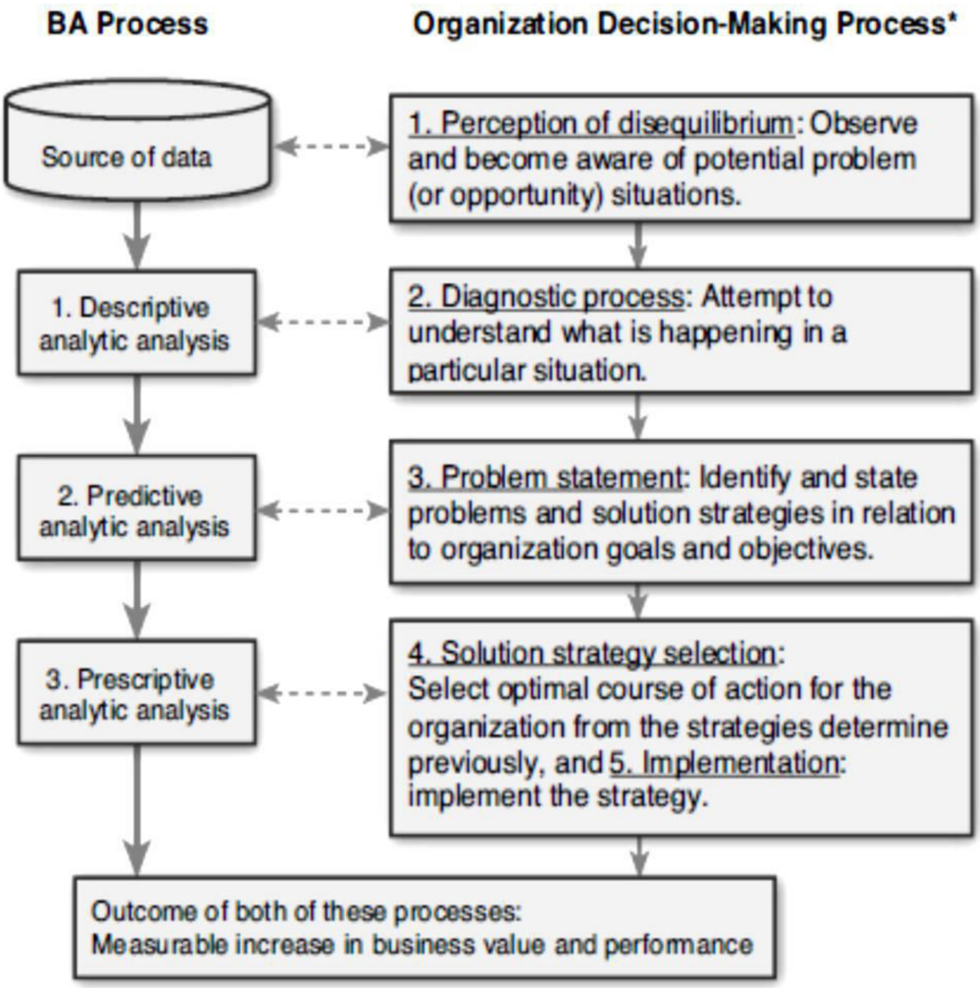
Until recently, it was believed that the development of technology was restricted only to human beings, but 21st century scientific studies indicate that other primates and certain dolphin communities have developed simple tools and passed their knowledge to other generations.

5.2 Understanding Business problems related to various businesses :

The BA process can solve problems and identify opportunities to improve business performance. In the process, organizations may also determine strategies to guide operations and help achieve competitive advantages. Typically, solving problems and identifying strategic opportunities to follow are organization decision-making tasks. The latter, identifying opportunities can be viewed as a problem of strategy choice requiring a solution.

A business analysis is the practice of identifying and clarifying problems or issues within a company and providing efficient solutions that satisfy the requirements of all stakeholders. ... Both information gathering techniques and communications with the various stakeholders are critical parts of the overall process.

5.3.1 Comparison of business analytics and organization decision-making processes:


BA Process
Organization Decision-Making Process*
Source of data
1. Descriptive analytic analysis
2. Predictive analytic analysis
3. Prescriptive analytic analysis
1. Perception of disequilibrium: Observe and become aware of potential problem (or opportunity) situations.
2. Diagnostic process: Attempt to understand what is happening in a particular situation.
3. Problem statement: Identify and state problems and solution strategies in relation to organization goals and objectives.
4. Solution strategy selection: Select optimal course of action for the organization from the strategies determine previously, and 5. Implementation: implement the strategy.
Outcome of both of these processes: Measurable increase in business value and performance

5.3.2.Requirements Gathering: Gather all the Data related to Business objective

There are many different approaches that can be used to gather information about a business. They include the following:
1.Review business plans, existing models and other documentation
2.Interview subject area experts
3.Conduct fact-finding meetings
4.Analyze application systems, forms, artifacts, reports, etc.

The business analyst should use one-on-one interviews early in the business analysis project to gage the strengths and weaknesses of potential project participants and to obtain basic information about the business. Large meetings are not a good use of time for data gathering.
Facilitated work sessions are a good mechanism for validating and refining "draft" requirements. They are also useful to prioritize final business requirements. Group dynamics can often generate even better ideas.
Primary or local data is collected by the business owner and can be collected by survey, focus group or observation. Third party static data is purchased in bulk without a specific intent in mind. While easy to get (if you have the cash) this data is not specific to your business and can be tough to sort through as you often get quite a bit more data than you need to meet your objective. Dynamic data is collected through a third party process in near real-time from an event for a specific purpose (read into that VERY expensive).
Three key questions you need to ask before making a decision about the best method for your firm.
··What is the timeline required to accomplish your business objective?
··What is your required return on investment?
··Is the data collection for a stand-alone event or for part of a broader data collection effort?

How to interpret Data to make it useful for Business:
Business intelligence (BI) is the set of techniques and tools for the transformation of raw data into meaningful and useful information for business analysis purposes. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. The goal of BI is to allow for the easy interpretation of these large volumes of data. Identifying new opportunities and implementing an effective strategy based on insights can provide businesses with a competitive market advantage and long-term stability.
BI technologies provide historical, current and predictive views of business operations. Common functions of business intelligence technologies are reporting, online analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytics and prescriptive analytics.
BI can be used to support a wide range of business decisions ranging from operational to strategic. Basic operating decisions include product positioning or pricing. Strategic business decisions include priorities, goals and directions at the broadest level. In all cases, BI is most effective when it combines data derived from the market in which a company operates (external data) with data from company sources internal to the business such as financial and operations data (internal data). When combined, external and internal data can provide a more complete picture which, in effect, creates an "intelligence" that cannot be derived by any singular set of data.

Business intelligence is made up of an increasing number of components including:

- Multidimensional aggregation and allocation
- Denormalization, tagging and standardization
- Realtime reporting with analytical alert
- A method of interfacing with unstructured data sources
- Group consolidation, budgeting and rolling forecasts
- Statistical inference and probabilistic simulation
- Key performance indicators optimization
- Version control and process management

- Open item management

Business intelligence can be applied to the following business purposes, in order to drive business value :

- Measurement – program that creates a hierarchy of performance metrics and benchmarking that informs business leaders about progress towards business goals (business process management).
- Analytics – program that builds quantitative processes for a business to arrive at optimal decisions and to perform business knowledge discovery. Frequently involves: data mining, process mining, statistical analysis, predictive analytics, predictive modeling, business process modeling, data lineage, complex event processing and prescriptive analytics.
- Reporting/enterprise reporting – program that builds infrastructure for strategic reporting to serve the strategic management of a business, not operational reporting. Frequently involves data visualization, executive information system and OLAP.
- Collaboration/collaboration platform – program that gets different areas (both inside and outside the business) to work together through data sharing and electronic data interchange.
- Knowledge management – program to make the company data-driven through strategies and practices to identify, create, represent, distribute, and enable adoption of insights and experiences that are true business knowledge. Knowledge management leads to learning management and regulatory compliance

In addition to the above, business intelligence can provide a pro-active approach, such as alert functionality that immediately notifies the end-user if certain conditions are met. For example, if some business metric exceeds a pre-defined threshold, the metric will be highlighted in standard reports, and the business analyst may be alerted via e-mail or another monitoring service. This end-to-end process requires data governance, which should be handled by the expert.

Data can be always gathered using surveys.
Your surveys should follow a few basic but important rules:
1. Keep it VERY simple. I recommend one page with 3-4 questions maximum. Customers are visiting to purchase or to have an experience, not to fill out surveys.
2. Choose only one objective for the survey. Don’t try to answer too many questions, ultimately you won’t get much useful data that way because your customer will get confused and frustrated.
3. Don’t give the respondent any wiggle room. Open ended questions are tough to manage. Specific choices that are broad enough to capture real responses gives you data that is much easier to use.
4. Always gather demographics. Why not? But rather than name and e-mail (leading to concerns with confidentiality and often less than truthful answers) gather gender, age and income; you might be surprised at who is actually buying what.

***** End of Unit-5 *****