

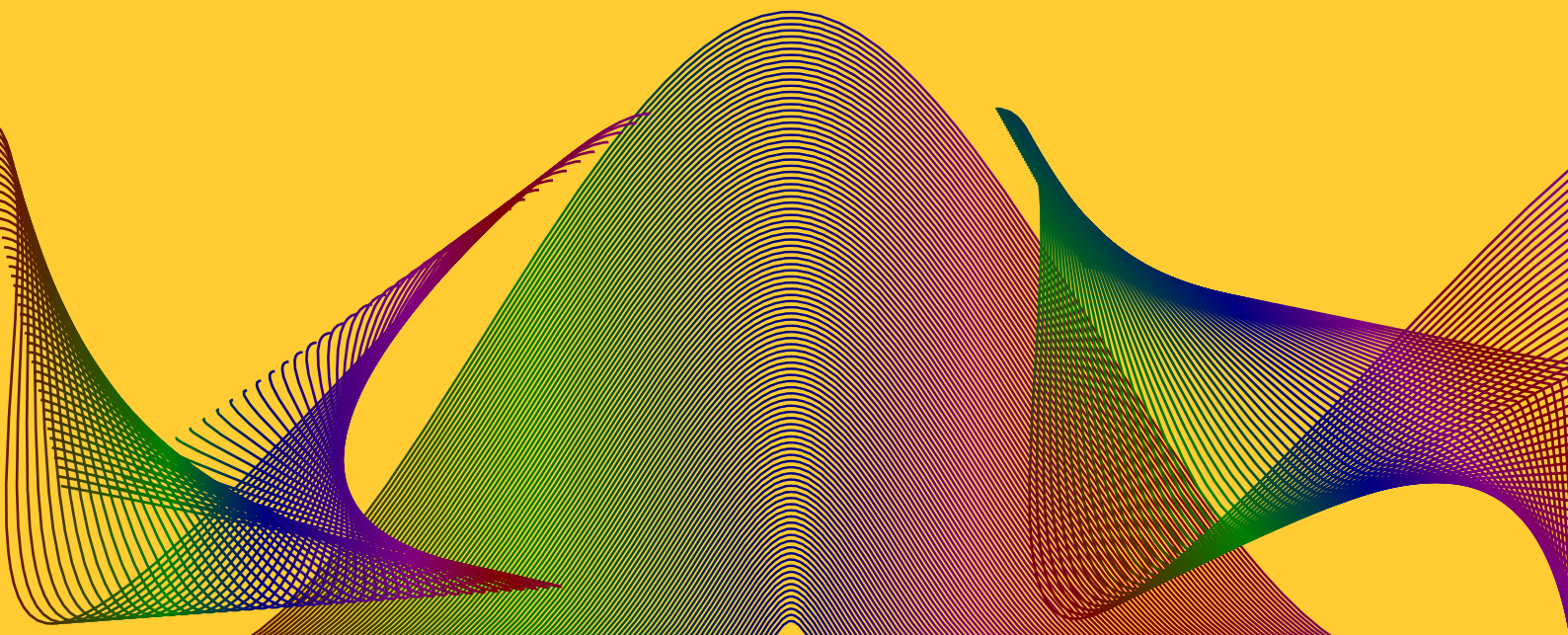


Hyderabad (Sind) National Collegiate Board's
Kishinchand Chellaram College
Churchgate, Mumbai - 20



Department of Statistics
under the aegis of STAR-DBT Scheme

Analyzing and Visualizing
Data with R Software
- A Practical Manual



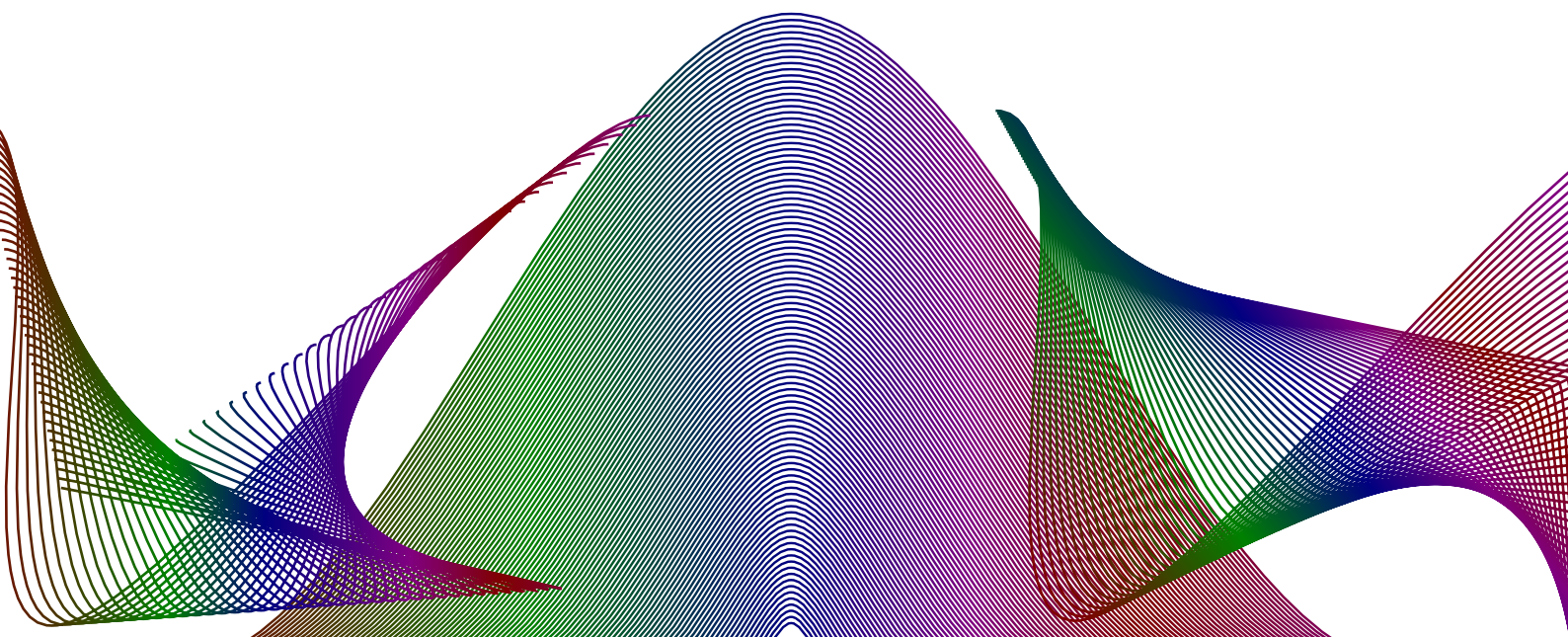


Hyderabad (Sind) National Collegiate Board's
Kishinchand Chellaram College
Churchgate, Mumbai - 20



Department of Statistics
under the aegis of STAR-DBT Scheme

Analyzing and Visualizing
Data with R Software
- A Practical Manual



Analyzing and Visualizing Big Data with R Software

Editor

Dr. Asha Angnamal Jindal

Department of Statistics



Kishinchand Chellaram College

Vidyasagar Principal K.M. Kundnani Chowk,
124, Dinshaw Wachha Road,
Churchgate, Mumbai 400020.

Tel: +91-22-2285 5726; +91-22-6698 1000;

Fax: +91-22-2202 9092;

Email: office@kccollege.edu.in

Website: <http://www.kccollege.edu.in/>



Shailja Prakashan

57-P, Kunj Vihar-II,
Yashoda Nagar, Kanpur-11

Ph.: 0512-2633004

Email: shailjaparakashan@gmail.com

ISBN 978-93-80788-71-5



9 789380 788715 >

ISBN 978-93-80788-71-5

Preface

Data science has its applications in all work areas like sentiment analysis, image analysis, sales forecasts, cost optimization etc. Big organizations like Google and Harvard have said that data science will be the best career option of the future!

Data Analytics field itself is still evolving and changing rapidly, with new strategies, tools and techniques coming online daily. These dynamics bring challenge to respond with innovative programs and curricular approaches that are connected deeply with Statistics analytics. Use of computers with Statistical packages has become essential for large datasets. These packages are very costly at initial stage with annual renewal cost.

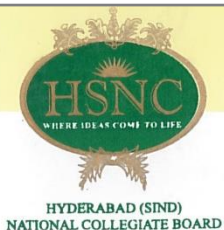
R is freely available software and it is easily downloaded from website www.r-project.org. It is not menu driven software. One should know appropriate command and functions for Statistical analysis in interactive mode of this software.

This practical manual covers all major topics required for learning R software to analyze data from F.Y.B.Sc. to T.Y.B.Sc. and some more. This manual contains illustrative examples and case studies which will be useful for academicians, researchers, industrialists and students for data analysis. As an activity under DBT-Star College Scheme this manual was an idea to also celebrate Diamond Jubilee of Department of Statistics.

This manual will serve the purpose of giving hands on training in undergraduate level under Star- DBT Scheme, Department of Biotechnology, Government of India as one of the reference book for data analysis.

I am thankful to all the contributors and my colleagues in Department of Statistics, K.C.College. I am thankful to Principal Dr. Hemlata Bagla for encouragement and guidance. I sincerely acknowledge Prof. (Dr.) S.A. Dubey for giving timely support in getting ISBN No. for this book. I am also thankful to Roshan Khilnani for designing book cover to forming all articles in a book form. I welcome suggestions and improvement in this manual from users of this souvenir/ manual.

Dr. Asha Angnamal Jindal
Editor



HYDERABAD (SIND) NATIONAL COLLEGIATE BOARD

President
Mr. Anil Harish
B.A., L.L.M.

Immediate Past President
Mr. Niranjan Hiranandani
B.Com., F.C.A.

Past President
Mr. Kishu H. Mansukhani
B.S.-M.E.

Secretary
Principal Dinesh Panjwani
B.A.(Hons.), M.Sc., M.Phil.

Rector
Prof. J. K. Bhambhani
M.Sc.



22.1.2018

I am very happy to learn that The National Statistics Data Championship 2018 has been organised by Department of Statistics at K. C. College. I understand that this Championship is organised through the support of star DBT grant received by the Science departments of the College. I am sure that this Championship will create a lot of interest amongst the students on the importance of statistics in the current world of Big Data and Data Analytics.

On behalf of HSNC Board, I congratulate K. C. College on the occasion of the Diamond Jubilee Celebration of Department of Statistics and for organising such an interesting and participative event so that the students are enthused and encouraged to delve into the world of statistics and data.

I am given to understand that this Championship will have participation from guides, mentors and students from various colleges across India.

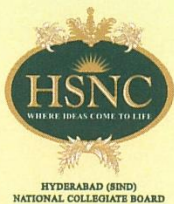
My best wishes to the Department of Statistics and Faculty and its students for intellectually stimulating event.

Anil Harish

Anil Harish

President

Hyderabad (Sind) National Collegiate Board



KISHINCHAND CHELLARAM COLLEGE

REACCREDITED 'A' GRADE BY NAAC (3rd Cycle)
BEST COLLEGE AWARD BY UNIVERSITY OF MUMBAI
AWARDEES 'STAR SCHEME' BY DBT & 'FIST PROGRAMME' BY DST



I/C PRINCIPAL

DR. HEMLATA K. BAGLA

M.Sc., Ph.D., FICCE



January 02, 2018

Message

With the motto of "Salvation through knowledge" and a vision to provide value-based and holistic education to the students and to equip them for global challenges, the College has worked tirelessly to emerge as a centre of academic excellence.

The Statistics Department of K.C. College is committed to advancing knowledge and learning tool teaching and research in statistics. It caters to the application of statistics in the faculty of Arts and Commerce as well as Computer System and its application to Commerce. During the last five years, the Department has published 30 research papers in peer reviewed national and international journals. The department has always been at the forefront of organizing and advocating application of statistics in various strolls of life by organizing workshops and classes to illuminate students with hands on approach.

The current venture of the Statistics department being the "National Big Data Analytics Championship 2018" based on the theme "Analyzing and Visualizing Big Data with R Software and MS-excel" held on January 02-03, 2018 & January 23-24, 2018 will surely provide practical insights to the participants about challenges faced when processing Big Data. Since, we live in the age where storing, processing and interpretation of data is of paramount importance and is growing exponentially. This manual on R will provide the essential basics to aid students in applying Statistics to Big Data.

The Department has performed in an exemplary manner by often producing university rank holders. I wish them all the best for this venture and future more to come...

Dr. Hemlata K. Bagla

I/c Principal

VIDYASAGAR PRINCIPAL K. M. KUNDNANI CHOWK
124, DINSHAW WACHHA ROAD, CHURCHGATE, MUMBAI - 400 020.

TEL: 91-22-6698 1000, 2285 5726 FAX: 91-22-2202 9092

EMAIL: office@kccollege.edu.in • WEB: www.kccollege.edu.in

INDEX

Title	Pg. No.
1 Introduction to R-Software Mr. Prashant Shah <i>Associate Professor and Head, Department of Statistics, K. J. Somaiya College of Science and Commerce, Vidyavihar, Mumbai.</i>	...001
2 Graphs and Diagram Mr. Prashant Shah <i>Associate Professor and Head, Department of Statistics, K. J. Somaiya College of Science and Commerce, Vidyavihar, Mumbai.</i>	...014
3 Measures of Central Tendency Mrs. Pratiksha M. Kadam <i>Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.</i>	...022
4 Measure of Dispersion Dr. Bhagat Gayval <i>Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.</i>	...031
5 Correlation, Regression and Curve Fitting Dr. Asha A. Jindal <i>Associate Professor and Head, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.</i>	...038
6 Time Series and Forecasting Techniques using R Mukesh Kumar Jain <i>CTO, VFS.GLOBAL, Mumbai, India</i>	...049
7 Probability and Probability Distributions Dr. Asha A. Jindal <i>Associate Professor and Head, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.</i>	...060
8 Sampling Distribution and Central Limit Theorem using R Dr. Rajendra Nana Chavhan <i>Assistant Professor Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.</i>	...083

9	Statistical Tests Using R	...094
	<i>Dr. Rajendra Nana Chavhan</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
10	Simple Random Sampling	...108
	<i>Mrs. Shailaja J. Rane</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
11	Stratified Random Sampling	...115
	<i>Dr. Asha A. Jindal</i> Associate Professor and Head, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
12	Analysis of Varince (ANOVA) using R	...118
	<i>Dr. Kalpana Dilip Phal</i> Associate Professor and Head, B.N.Bandodkar College of Science, Thane, Chendani, Thane (West) 400601.	
13	Designs of Experiment using R	...124
	<i>Dr. Kalpana Dilip Phal</i> Associate Professor and Head, B.N.Bandodkar College of Science, Thane, Chendani, Thane (West) - 400601	
14	Linear Programming Problem, Transportation Problem and Assignment problem using R-Software	...129
	<i>Dr. S. B. Muley</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
15	Theory of Estimation	...134
	<i>Mrs. Shailaja J. Rane</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
16	Financial Functions	...139
	<i>Mrs. Pratiksha M. Kadam</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	
17	Non-Parametric Test	...147
	<i>Dr. S. B. Muley</i> Assistant Professor, Department of Statistics, K. C. College, Churchgate, Mumbai – 400 020.	

- 18 Multiple Regression: A Case Study** ...160
Gourav Tiwari, Mohammed Saad Qadri, Misbahuddin Saba, Dr. Asha Jindal*
Researcher, Star DBT Scheme, Dept. of Statistics, K.C. College, Mumbai - 20, INDIA.
**Star - DBT Mentor & Associate Professor and Head, Dept. of Statistics, K.C. College, Mumbai, INDIA – 20.*
- 19 Logistics Regression: Human Resources Analytics** ...172
Why do our employees leave prematurely?
Pravesh .S. Tiwari¹, Divya .M. Poojari²
¹Data Analyst in Accenture; ²Statistical Programmer in Cognizant
- 20 Factor Analysis** ...181
Dr. Santosh P. Gite
Associate Professor and Head, Department of Statistics, University of Mumbai, Mumbai.
- 21 Sentiment Analysis** ...189
Jain Jimit, Karani Hardik, Sen Milankumar, Dr. Asha Jindal*
Researcher, Star DBT Scheme, Dept. of Statistics, K.C. College, Mumbai - 20, INDIA.
**Star - DBT Mentor & Associate Professor and Head, Dept. of Statistics, K.C. College, Mumbai - 20.*
- 22 Discriminant Analysis** ...195
Dr. Suresh Kumar Sharma
Professor, Department of Statistics & Coordinator, Centre for Systems Biology & Bioinformatics, Panjab University, Chandigarh-India
- 23 Cluster Analysis** ...199
Dr. Suresh Kumar Sharma
Professor, Department of Statistics & Coordinator, Centre for Systems Biology & Bioinformatics, Panjab University, Chandigarh-India.

Chapter 1

Introduction to R-Software

Mr. Prashant Shah, Associate Professor and Head, Department of Statistics,
K. J. Somaiya College of Science and Commerce, Vidyavihar, Mumbai.

1.1 R as a programming language

While R is perhaps best known as a statistical tool for analyzing data or for making graphs, it is also really useful as a simple programming language and compiler. In R, a program is just any group of commands that you wish to run as a set, to achieve some output.

1.1.1 Using Text Editors and ".R" Files in R

By using a text editor, we can write whole groups of commands and have the computer run them separately or all together. Further, text editors allow you to save your program for later use.

There are three different types of windows that are used by R: console, graphics, and text editor windows. The window where you enter line commands is the R Console. When you used the "plot" command, it opened a new window, which is the graphics window. Text editor windows are just simple text editors that are smart enough to interact with R.

On a PC, go to "File" and open "New script". To execute commands, either highlights the command(s) or put the cursor anywhere on that line and push the button in upper corner of the main R window for "Run line or selection."

Creating a new document (or script) opens a simple text editor in R. You can then enter multiple lines of commands that are not executed until you are ready. And, instead of executing commands one by one, you can execute them all at once or any set of them together. You can also save the file (usually as a "____.R" file) and rerun these commands at a later time.

1.2 What is Statistics?

The subject of statistics deals with

- Collection of data.
- Analysis of data.
- Presentation or organization of data.
- Interpretation of, results of, analysis of data.

1.3 What is Data?

A set of numerical or other measured values

For Eg. 1. Salaries of employees.

2. Export (Rs. in crore) of a company during 2010 to 2015.

3. Daily credit/debit transactions in bank.

4. Carbon dioxide content in the air, in different regions during different seasons.

5. Patient's disease history in hospitals.

To analyse voluminous data, a number of statistical software are available such as

- R-software
- SAS (**Statistical Analysis System**)
- SPSS (**Statistical Package for the Social Sciences**)
- Minitab

R is the most comprehensive statistical analysis package available. It incorporates all of the standard statistical tests, models, and analyses, as well as providing a comprehensive language for managing and manipulating data. R is free and open source software, allowing anyone to use and, importantly, to modify it.

1.4 R commands, case sensitivity

Technically R is an expression language with a very simple syntax. **It is case sensitive**, so A and a are different symbols and would refer to different variables.

Elementary commands consist of either expressions or assignments. If an expression is given as a command, it is evaluated, printed (unless specifically made invisible), and the value is lost.

An assignment also evaluates an expression and passes the value to a variable but the result is not automatically printed.

Commands are separated either by a semi-colon (;), or by a newline. Elementary commands can be grouped together into one compound expression by braces ({ and }).

Comments can be put almost anywhere, starting with a hashmark (#), everything to the end of the line is a comment.

If a command is not complete at the end of a line, R will give a different prompt, by default + on second and subsequent lines and continue to read input until the command is syntactically complete.

1.5 R-Commands to input data

a) Assignment Statement

- = or <-

b) Creating vectors

- c()
- scan()

c) Generating sequences

- :
- seq()
- seq(from = a, to = b, by = c)
- seq(length=d, from = a, by = c)

d) Replicating objects or elements

- rep()

1.6 Simple manipulations; numbers and vectors

1.6.1 Assignment:

An assignment means naming a value, so that it can be used later. Assignment has general form

Variable = expression or value (= is an assignment operator)

```
> x = 2 + 3    # x is assigned value 5
> x
[1] 5
> x + 2
[1] 7
> x = x * 3
> x
[1] 15
> x = 2 + 3; y = -4; z = x * y # Commands are separated by a semi-colon (';')
> x; y; z
[1] 5
[1] -4
[1] -20
> x = 2 + 3; y = -4; z = x * y; x; y; z # Commands are separated by a semi-
                                         colon (';')
```

1.6.2 Vectors

R operates on named data structures. The simplest such structure is the **numeric vector**, which is a single entity consisting of an ordered collection of numbers. To set up a vector

named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an assignment statement using the function `c()` which in this context can take an arbitrary number of vector arguments (`c` stands for “combine.”). The idea is that a list of numbers is stored under a given name, and the name is used to refer to the data. The numbers within the `c` command are separated by commas. A list is specified with the `c` command, and assignment is specified with the “<-” symbols. Notice that the assignment operator (“<-”), which consists of the two characters ‘<’ (“less than”) and ‘-’ (“minus”) occurring strictly side-by-side and it ‘points’ to the object receiving the value of the expression. A number occurring by itself in an expression is taken as a vector of length one.

If an expression is used as a complete command, the value is printed and lost. So now if we were to use the command

```
> 1/x
```

the reciprocals of the five values would be printed at the terminal (and the value of `x`, of course, unchanged).

The further assignment

```
> b <- c(x, 0, x)
```

would create a vector `b` with 11 entries consisting of two copies of `x` with a zero in the middle place.

To see what numbers is included in `x` type “`x`” and press the enter key:

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
> x
[1] 10.4  5.6  3.1  6.4  21.7
> typeof(x)
[1] "double"
```

1.6.3 Accessing vectors:

Individual elements of a vector can be accessed by using indices.

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
> x[3]                # third element of vector x is accessed.
[1] 3.1
> x[1]                # first element of vector x is accessed.
[1] 10.4
> x[2 : 4]            # elements from second to fourth of vector x are accessed.
[1] 5.6  3.1  6.4
> x[c(2,5)]           # elements having indices 2 and 5 are accessed.
[1] 5.6  21.7
> length(x)           # displays number of elements in vector x.
[1] 5
> x[3 : length(x)]    # elements having indices 3 to 5 of vector x
are accessed.
[1] 3.1  6.4  21.7
```

```

> x[4 : 2]      # elements from fourth to second reversely of vector x are
                accessed.
[1] 6.4   3.1   5.6
> x[0]
numeric(0)
> x[6]
[1] NA
> x[x > 6]      # elements of vector x having value > 6 are accessed.
[1] 10.4   6.4  21.7
> x[x < 6]      # elements of vector x having value < 6 are accessed.
[1] 5.6   3.1

```

Subset command can also be used with vectors.

```

> q = subset(x, x > 6)
> q
[1] 10.4   6.4  21.7
> p = subset(x, x < 6)
> p
[1] 5.6  3.1
> which(x < 6)  # displays index of elements of vector x whose value is < 6.
[1] 2 3
> x[-1]        # elements except first are accessed.
[1]  5.6  3.1  6.4  21.7
> x[c(-2,-5)]  # elements except second and fifth are accessed or x[-c(2,5)]
[1] 10.4   3.1   6.4
> x[-2 : -4]   # elements except second to fourth are accessed.
[1] 10.4  21.7
> x < 6
[1] FALSE  TRUE  TRUE FALSE FALSE

```

Notice that the first entry is referred to as the number 1 entry and the zero entry can be used to indicate how the computer will treat the data.

```

> 1/x
[1] 0.09615385 0.17857143 0.32258065 0.15625000 0.04608295
> x
[1] 10.4   5.6   3.1   6.4  21.7
> b <- c(x, 0, x)
> b
[1] 10.4   5.6   3.1   6.4  21.7  0.0  10.4   5.6   3.1   6.4  21.7

```

You can store strings using both single and double quotes.

```

> t <- c("somaiya", "mumbai", 'new delhi')
> t
[1] "somaiya"   "mumbai"    "new delhi"
> typeof(t)
[1] "character"

```

1.6.4 Alternative way to create data vectors

Vectors can be created and data can be entered alternatively by using scan function.

```

> x = scan()
1: 3 -5 7
4: 9 0 6.7

```

```

7: -2
8:
Read 7 items
> x
[1] 3.0 -5.0 7.0 9.0 0.0 6.7 -2.0
> y = scan()
1: 2 5 8 4 -2
6: 9 5
8:
Read 7 items
> y
[1] 2 5 8 4 -2 9 5

```

scan() function has many other arguments such as **what**, **nmax** etc

- **what**: This argument indicates types of data to be accepted, by default it is numeric. For character data type set what = "character"
- **nmax**: This argument indicates maximum number of elements to be accepted.

```

> t = scan(what = "character")
1: "somaiya" "vidyavihar"
3:
Read 2 items
> t
[1] "somaiya" "vidyavihar"
> x = scan(nmax = 4)
1: 5 -8 3 9 2 -11 6
Read 4 items
> x
[1] 5 -8 3 9

```

1.6.4 Vector arithmetic

Vectors can be used in arithmetic expressions, in which case the operations are performed element by element.

The elementary arithmetic operators are the usual +, -, *, / and ^ for raising to a power. In addition, several mathematical and statistical functions are also available in R for arithmetic operations. For eg.: log, log10, sort, min, max, range, length, exp, sin, cos, tan, sqrt, and so on, all have their usual meaning.

Vectors are mathematical objects. Standard arithmetic functions and operators apply to vectors on element wise basis.

While applying simple arithmetic functions and operators to vectors proper care should be taken. If the operands are of different lengths then shorter of the two is extended by repetition. However, if the length of the longer is not multiple of length of shorter then warning message is displayed.

```

> c(1,5,2,3) + c(1,3)
[1] 2 8 3 6

```

```
> c(1,5,2) + c(1,3)
[1] 2 8 3
```

Warning message:

```
In c(1, 5, 2) + c(1, 3) :
longer object length is not a multiple of shorter object length
```

1.7 Generating regular sequences

R has a number of facilities for generating commonly used sequences of numbers. For example 12:20 is the vector `c(12, 13, ..., 20)`. The colon operator has high priority within an expression, so, for example `2*12:20` is the vector `c(24, 26, ..., 40)`. Put `n <- 8` and compare the sequences `1:n-1` and `1:(n-1)`.

```
> 12:20
[1] 12 13 14 15 16 17 18 19 20
> p <- 12:20
> p
[1] 12 13 14 15 16 17 18 19 20
> q <- 3*12:20
> q
[1] 36 39 42 45 48 51 54 57 60
> n = 8
> t <- 5:(n-1)
> t
[1] 5 6 7
> w <- 5:n - 1
> w
[1] 4 5 6 7
```

The construction `20:12` may be used to generate a sequence backwards.

```
> 20:12
[1] 20 19 18 17 16 15 14 13 12
```

The function `seq()` is a more general facility for generating sequences. It has five arguments, only some of which may be specified in any one call. The first two arguments, if given, specify the beginning and end of the sequence, and if these are the only two arguments given the result is the same as the colon operator. That is **`seq(12,20)`** is the same vector as **`12:20`**.

Parameters to `seq()`, and to many other R functions, can also be given in **named form**, in which case the order in which they appear is irrelevant. The first two parameters may be named **`from=value`** and **`to=value`**; thus `seq(12,20)`, `seq(from=12, to=20)` and `seq(to=20, from=12)` are all the same as `12:20`. The next two parameters to `seq()` may be named **`by=value`** and **`length=value`**, which specify a step size and a length for the sequence respectively. If neither of these is given, the default `by=1` is assumed.

For example

```
> seq(-5, 5, by=.2) -> s3
> s3
```



```
[1] -5.0 -4.8 -4.6 -4.4 -4.2 -4.0 -3.8 -3.6 -3.4 -3.2 -3.0 -2.8 -2.6 -2.4 -
2.2
[16] -2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4 -0.2 0.0 0.2 0.4 0.6
0.8
[31] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6
3.8
[46] 4.0 4.2 4.4 4.6 4.8 5.0
```

Similarly following command generates a sequence of 18 elements

```
> s4 <- seq(length=18, from=-5, by=.2)
> s4
[1] -5.0 -4.8 -4.6 -4.4 -4.2 -4.0 -3.8 -3.6 -3.4 -3.2 -3.0 -2.8 -2.6 -2.4 -
2.2
[16] -2.0 -1.8 -1.6
```

`rep()` which can be used for replicating an object in various complicated ways. The simplest form is `s5 <- rep(x, times=5)` which will put five copies of `x` end-to-end in `s5`.

```
> x
[1] 305 16 122 68
> s5 <- rep(x, times=5)
> s5
[1] 305 16 122 68 305 16 122 68 305 16 122 68 305 16 122 68 305 16
122
[20] 68
```

Another useful version is `s6 <- rep(x, each=5)` which repeats each element of `x` five times before moving on to the next.

```
> s6 <- rep(x, each=5)
> s6
[1] 305 305 305 305 305 16 16 16 16 16 122 122 122 122 122 68 68 68
68
[20] 68
> s7 <- rep(1:4, c(2,1,2,1))
> s7
[1] 1 1 2 3 3 4
```

1.8 Matrix Operation

To form a matrix you can use following syntax.

`matrix(data =, nrow =, ncol =, byrow = "FALSE")`.

- `data` : Actual data may be written in any of the variable or values by using function `c()`.
- `nrow` : Number of rows of a matrix
- `ncol` : Number of columns of a matrix
- `byrow` : It specifies whether matrix values are filled row wise or column wise. `FALSE` is by default i.e. column wise. If you want row wise then use `TRUE`.

For example,

```
> a <- c(1,2,3,4,5,6,7,8,9,10,11,12)
> A <- matrix(data=a, nrow=3, ncol=4, byrow="TRUE")
> a
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> A
      [,1] [,2] [,3] [,4]
[1,] 1    2    3    4
[2,] 5    6    7    8
[3,] 9   10   11   12
```

Specific values in a vector or in a matrix are referenced using square brackets ([]). For example,

```
> x <- c(5,8,9,7,6)
> x
[1] 5 8 9 7 6
> x[2]
[1] 8
> A[2,4]
[1] 8
> A[3,]
[1] 9 10 11 12
> A[c(2,3),1]      #display 2nd and 3rd element of the first column of matrix A
[1] 5 9
> A[c(2,3),2]
[1] 6 10 #display 2nd and 3rd element of the second column of matrix A
```

Matrix operators are provided in the Table

Table 2: Matrix Operations

Operation or Function	Description
$A * B$	Element-wise multiplication
$A \%*\% B$	Matrix multiplication
$t(A)$	Transpose
$diag(x)$	Creates diagonal matrix with elements of x in the principal diagonal
$diag(A)$	Returns a vector containing the elements of the principal diagonal
$diag(k)$	If k is a scalar, this creates a k x k identity matrix.
$solve(A,b)$	Returns vector x in the equation $b = Ax$
$solve(A)$	Inverse of A where A is a square matrix.
$ginv(A)$	Moore-Penrose Generalized Inverse of A. it requires loading the MASS package.
$y \leftarrow qr(A)\$rank$	rank is the rank of A.
$cbind(A,B,...)$	Combine matrices(vectors) horizontally. Returns a matrix.
$rbind(A,B,...)$	Combine matrices(vectors) vertically. Returns a matrix.

1.9 Some commonly used Built-in functions

```
> x <- c(-6,9,0,-3,8,2,-5,4)
> x
[1] -6  9  0 -3  8  2 -5  4
> length(x)      #Displays the number of elements of vector x
[1] 8
> max(x)         #displays the maximum element of vector x
[1] 9
> min(x)         #displays the minimum element of vector x
[1] -6
> range(x)       #displays the range of the values of vector x
[1] -6  9
> sum(x)         # displays sum of the values of vector x
[1] 9
> cumsum(x)      # displays the cumulative sum of the values of vector x
[1] -6  3  3  0  8 10  5  9
> mean(x)        # displays the mean of the values of vector x
[1] 1.125
> median(x)      # displays the median of the values of vector x
[1] 1
> sort(x)       # Sort the values of vector x in the increasing order
[1] -6 -5 -3  0  2  4  8  9
> sort(x, decreasing = T)    # Sort the values of vector x in the decreasing
                             order
[1]  9  8  4  2  0 -3 -5 -6
> var(x)        # Sample variance with denominator (n-1)
[1] 32.125
> which(x == 4)  # displays index of the required element of vector x
[1] 8
> y <- c(3,4,-5)
> prod(y)       # displays product of the values of vector y
[1] -60
```

round() : Syntax for the function is round(object, digits)

This function rounds object upto digits decimals. For example,

```
> round(3.2156,3)
[1] 3.216
```

1.10 Data frames

Data frames can be created by using data.frame. A data frame may be regarded as a matrix. It may be displayed in matrix form, and its rows and columns extracted using matrix indexing conventions. It is a list of vectors of the same length. (If the vectors included in the data frame are not of the same length then vector having less elements is recycled a whole number of times)

```
> x <- c(-5,7,-3,8); y = 8:11; z = rep(-5,4); p = seq(1,12,3)
> q = c(1,5)
> r = 5:7
> x;y;z;p;q;r
[1] -5  7 -3  8
```

```
[1] 8 9 10 11
[1] -5 -5 -5 -5
[1] 1 4 7 10
[1] 1 5
[1] 5 6 7
> d1 = data.frame(x,y)
> d1
```

	x	y
1	-5	8
2	7	9
3	-3	10
4	8	11

First column indicates row numbers.

```
> d2 = data.frame(q,p)
> d2
```

	q	p
1	1	1
2	5	4
3	1	7
4	5	10

In this data frame d2 vector having fewer elements (i.e. vector q) is recycled a whole number of times (2 times, so that its length becomes as that of length of other vector p)

Different columns in data frame are vectors. Names can be given to these columns while creating data frames.

```
> d4 = data.frame("maths" = x, "stats" = y)
> d4
```

	maths	stats
1	-5	8
2	7	9
3	-3	10
4	8	11

Rows in data frames can be given names using **row.names** which is a vector of character strings indicating names of rows.

```
> d5 = data.frame("maths" = x, "stats" = y, row.names = c("Amit", "Vidya",
"Ganesh", "Tina"))
> d5
```

	maths	stats
Amit	-5	8
Vidya	7	9
Ganesh	-3	10
Tina	8	11

1.11 Accessing data from data frames

Data from data frame can be accessed using \$ notation

```
> d5 $ maths
[1] -5 7 -3 8
```



```
> d5 $ maths[3]
[1] -3
> d5[4,2]
[1] 11
```

1.12 Inbuilt data sets or Resident data sets

The data sets that come with R or one of the packages are known as Inbuilt data sets. To view all Inbuilt data sets names from package 'datasets' use following command.

```
> data()
```

For accessing existing data sets, command is as follows

```
> data(data set name)
> data(co2)
> co2          # displays data set co2
Note: Data frame can also be created using in-built data editor edit similar
to MS-Excel.
> stud <- edit(data.frame()) #this command displays in-built spread sheet.
> stud
```

	var1	var2
1	fybsc	45
2	fybsc	50
3	sybsc	55
4	msc	60

```
> names(stud) <- c("Standard", "Marks")
> stud
```

	Standard	Marks
1	fybsc	45
2	fybsc	50
3	sybsc	55
4	msc	60

1.13 Importing Data from Excel

The function read.table() is the easiest way to import data into R. The preferred raw data format is either a tab delimited or a comma-separate file (CSV).

Working directory can be checked using getwd().

Store the excel file in csv format in this working directory.

```
> d1 <- read.table("temp1.csv", header=TRUE, sep=",")
# This creates dataframe d1
> d1
```

	Roll.No	Name	Marks
1	21	fgdgf	45
2	22	wqeq	78
3	23	zxcvz	60
4	25	jkljl	47

```
> dm = as.matrix(exp)
> dm
```

	Item	Ramesh	Ganesh
[1,]	"Food"	"1600"	"1200"
[2,]	"Rent"	"1500"	"2000"
[3,]	"Electricity"	"1000"	"1500"
[4,]	"Misc."	"900"	"3000"

Chapter 2

Graphs and Diagram

Mr. Prashant Shah, Associate Professor and Head, Department of Statistics,
K. J. Somaiya College of Science and Commerce, Vidyavihar, Mumbai.

2.1 Introduction

Statistical data can be represented in the form of diagrams such as

- Simple bar diagram
- Multiple bar diagram
- Subdivided bar diagram
- Pie diagram or pie chart

2.2 Bar Diagrams

```
> barplot(height, beside = T, names.arg = NULL, col = NULL, border =
par("fg"), main = NULL, xlab = NULL, ylab = NULL, xlim = NULL, ylim =
NULL,...)
```

height: Either a vector or matrix of values describing the bars which make up the plot. If height is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector. If height is a matrix and beside is FALSE then each bar of the plot corresponds to a column of height, with the values in the column giving the heights of stacked sub-bars making up the bar. If height is a matrix and beside is TRUE, then the values in each column are juxtaposed rather than stacked.

names.arg: A vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the names attribute of height if this is a vector, or the column names if it is a matrix.

main: Overall title for the plot.

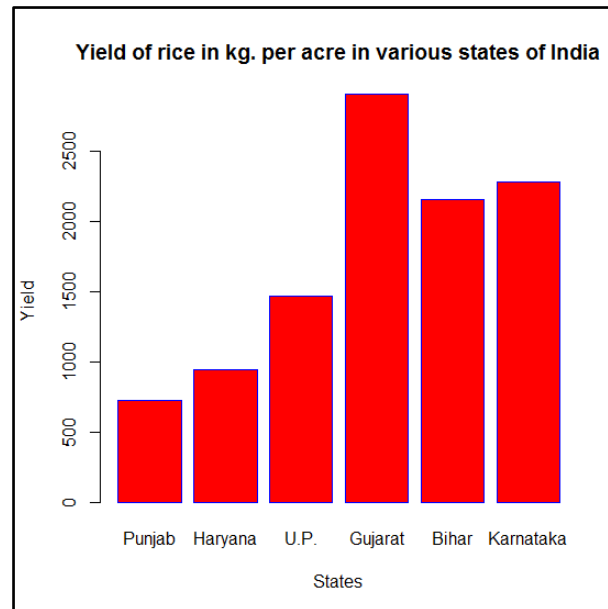
beside: A logical value. If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars (adjoining or contiguous bars)

Example: The following table gives the average approximate yield of rice in kg. per acre in various states of India in 2003-04. Represent it by **Simple Bar diagram**.

State :	Punjab	Haryana	U.P.	Gujarat	Bihar	Karnataka
Yield :	728	943	1469	2903	2153	2276

```
> x <- c("Punjab", "Haryana", "U.P.", "Gujarat", "Bihar", "Karnataka")
```

```
> y <- c(728, 943, 1469, 2903, 2153, 2276)
> x
[1] "Punjab"      "Haryana"     "U.P."        "Gujarat"     "Bihar"       "Karnataka"
> y
[1] 728  943 1469 2903 2153 2276
> barplot(y, names.arg = x, col = "red", border = "blue", main = "Yield of rice
in kg. per acre in various states of India", xlab = "States", ylab = "Yield")
```



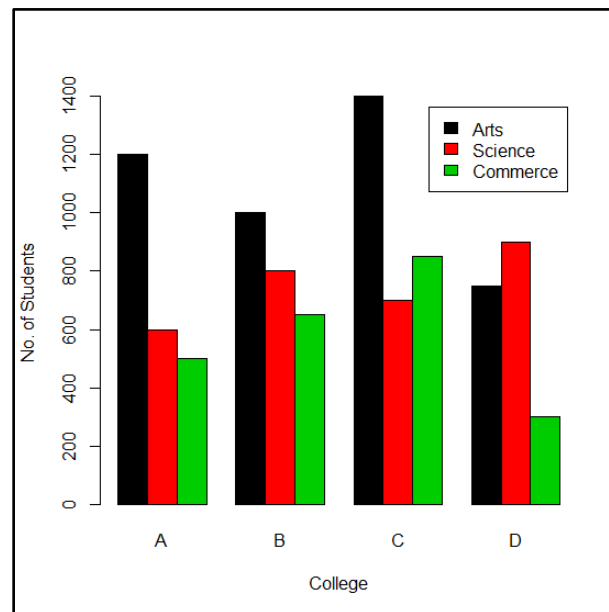
Example: Represent the following data on faculty-wise distribution of students, by **multiple bar diagram**.

College	Arts	Science	Commerce
A	1200	600	500
B	1000	800	650
C	1400	700	850
D	750	900	300

```
> clg <- c("A", "B", "C", "D")
> clgA <- c(1200, 600, 500)
> clgB <- c(1000, 800, 650)
> clgC <- c(1400, 700, 850)
> clgD <- c(750, 900, 300)
> d = data.frame(clgA, clgB, clgC, clgD)
> d
  clgA clgB clgC clgD
1 1200 1000 1400  750
2  600  800  700  900
3  500  650  850  300
> d1 = as.matrix(d)
> d1
     clgA clgB clgC clgD
[1,] 1200 1000 1400  750
```

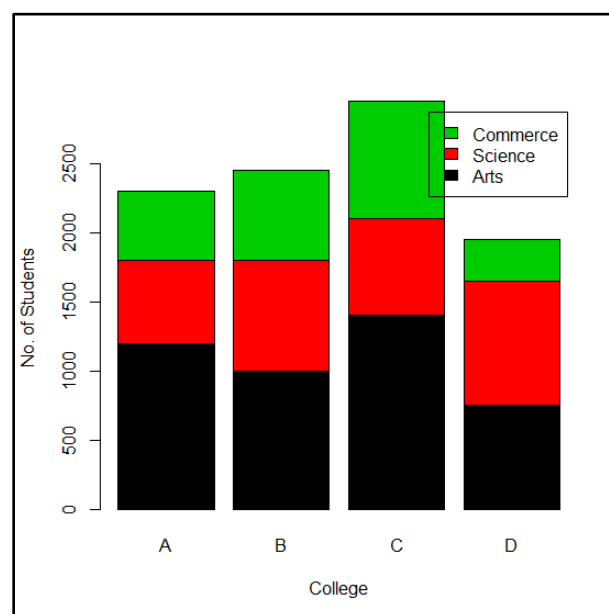


```
[2,]      600      800      700      900
[3,]      500      650      850      300
> barplot(d1, beside = T, names.arg = clg, col = 1:2:3, legend = c("Arts",
"Science", "Commerce"),xlab = "College", ylab = "No. of Students")
```

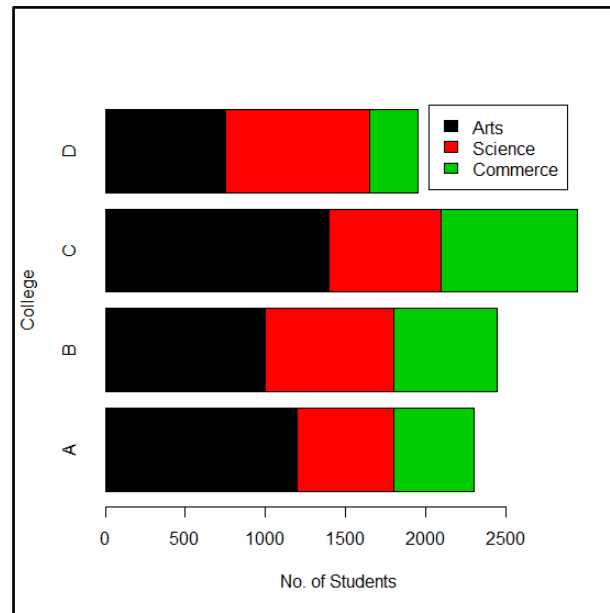


For the above example draw **subdivided bar diagram**.

```
> barplot(d1, beside = F, names.arg = clg, col = 1:2:3:4, legend = c("A",
"B", "C", "D"),xlab = "College", ylab = "No. of Students")
```



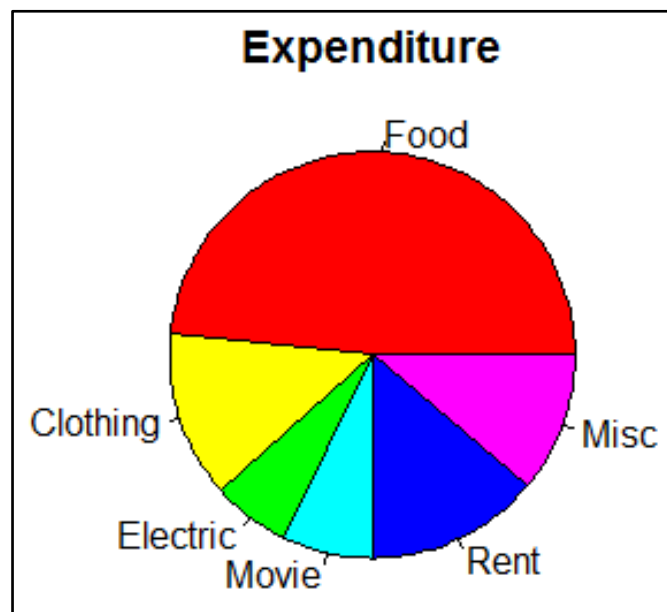
```
barplot(d1, beside = F, horiz = T, names.arg = clg, col = 1:2:3, legend =
c("Arts", "Science", "Commerce"),ylab = "College", xlab = "No. of Students")
```



Example: Represent the following data by a **pie diagram**:

Item :	Food	Clothing	Recreation	Indian	Rent	Miscellaneous
Expenditure (in Rs.)	87	24	11	13	25	20

```
> itm <- c("Food", "Clothing", "Electric", "Movie", "Rent", "Misc")
> exp ,- c(87, 24, 11, 13, 25, 20)
> pie(exp, main = "Expenditure", labels = itm, radius = 1,
col=rainbow(length(exp)))
```



2.3 Graphical Representation of data

Statistical data can be represented in the form of graphs such as

- Histogram
- Frequency polygon
- Ogive curve

R supports commands hist, plot, lines, points etc for drawing above graphs.

2.3.1 Histogram

```
> hist(x, breaks = classlimits, freq/probability = False/True, density =
NULL, col = NULL, border = NULL, main = paste("Histogram of", xname), xlim =
range(breaks), ylim = NULL, xlab = xname, ylab=yname, axes = TRUE . . . .)
```

x: A vector of values for which the histogram is desired.

breaks: A vector giving breakpoints (class limits) for histogram. This can be done using c() or seq(). For eg: **breaks=c(100, 300, 500, 700)** Compute a histogram for the raw data values and set the bins (bars) such that they run from 100 to 300, 300 to 500 and 500 to 700. However, the c() function can make your code very messy sometimes. That is why you can instead use **breaks=seq(x, y, z)**. The values of x, y and z are determined by yourself and represent, in order of appearance, the begin number of the x-axis, the end number of the x-axis and the interval in which these numbers appear.

```
> brk <- seq(148,178,5)
> hist(x, breaks = brk)
```

This command creates histogram with class limits 148 to 153, 153 to 158, 158 to 163, 163 to 168, 168 to 173, 173 to 178.

Note that you can also combine the two functions:

```
> hist(x, breaks=c(100, seq(200,700, 150)))
```

Make a histogram for the vector x, start at 100 on the x-axis, and from values 200 to 700, make the bins 150 wide

freq/probability: logical; if TRUE, the histogram graphic is a representation of frequencies; if FALSE, probability densities, are plotted (so that the histogram has a total area of one). Defaults to TRUE *if and only if* breaks are equidistant (and probability is not specified).

density: the density of shading lines, in lines per inch. The default value of NULL means that no shading lines are drawn.

col: a colour to be used to fill the bars. The default of NULL yields unfilled bars.

border: the color of the border around the bars. The default is to use the standard foreground color.

main: Overall title for the plot.

```
> brk <- seq(148,178,5)
> xnme = "Heights"
> hist(x, breaks = brk, freq = FALSE, main = paste("Histogram of" , xnme))

> x <- scan()
1: 170 151 154 160 158 154 171 156 160 157 148 165 158
14: 160 157 159 155 151 152 161 156 164 156 163 174 153 170 149 166 154
```

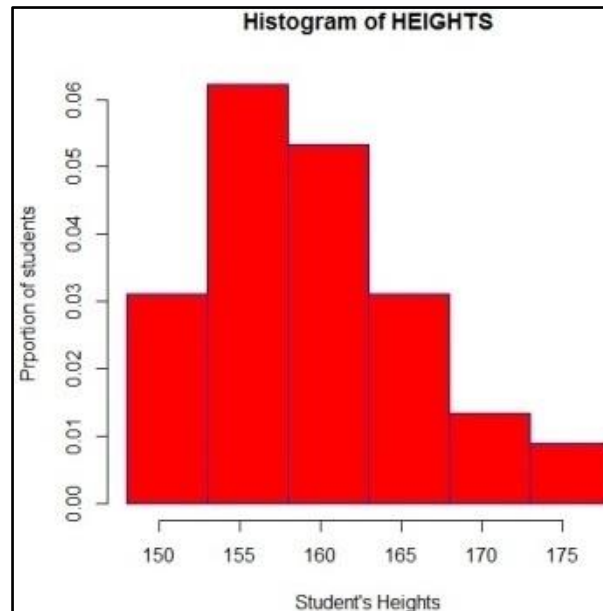
```
31: 166 160 160 161 154 163 164 160 148 162 167 165 158 158 176
```

```
46:
```

```
Read 45 items
```

```
> hist(x)
```

```
> hist(x, breaks = brk, freq = FALSE, col = "red", border = "blue", main =  
paste("Histogram of" , xnm), xlab = "Student's Heights", ylab="Prportion of  
students")
```



Histogram for ungrouped frequency data

x:	150	155	160	165	170	175
f:	6	11	14	9	3	2

```
> x <- seq(150,175,5)
> f <- c(6,11,14,9,3,2)
> y <- rep(x,f)
> hist(y)
> t = seq(147.5,177.5,5)
> hist(y, breaks = t)
```

Histogram for grouped frequency data

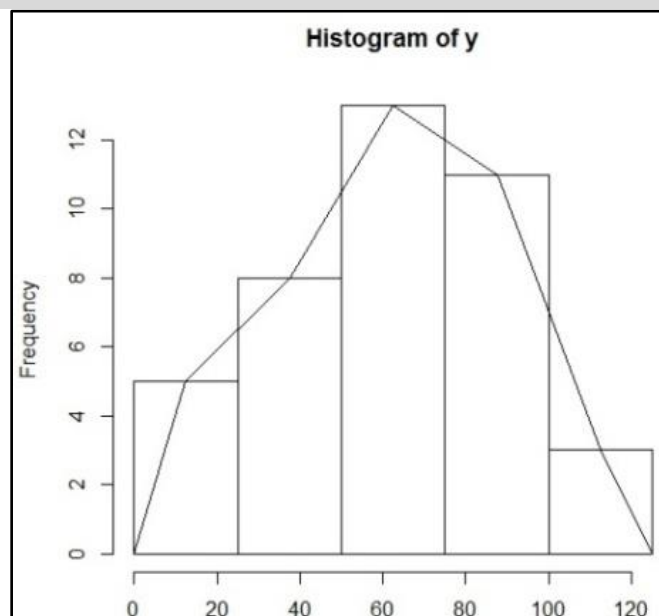
C.I.	0-25	25-50	50-75	75-100	100-125
f:	5	8	13	11	3

```
> midx <- seq(12.5,112.5,25)
> f <- c(5,8,13,11,3)
> cls_limit <- seq(0,125,25)
> y <- rep(midx,f)
> hist(y)
> hist(y, breaks=cls_limit)
```

2.3.2 Frequency polygon

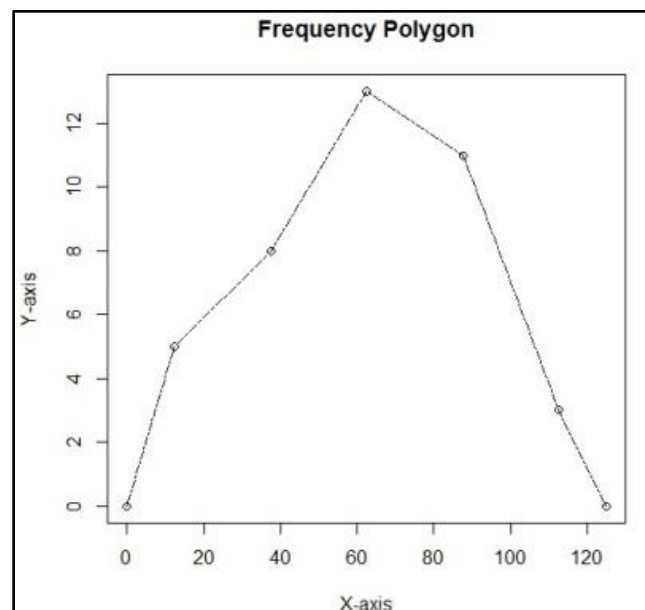
It is obtained by joining the points (x_i, f_i) where x_i is the midpoint of the i^{th} class interval and f_i is the corresponding frequency.

```
> lb <- seq(0,100,25)
> ub <- seq(25, 125, 25)
> midx <- (lb+ub)/2
> f <- c(5,8,13,11,3)
> x0 <- c(0, midx, 125)
> f0 <- c(0,f,0)
> y <- rep(midx,f)
> bks <- seq(0,125,25)
> hist(y,breaks=bks)
> lines(x0, f0)
```



OR

```
> plot(x0,f0, main = "Frequency Polygon", xlab ="X-axis", ylab = "Y-axis",
type = "o", lty =6, xlim = range(min(x0),max(x0)))
```



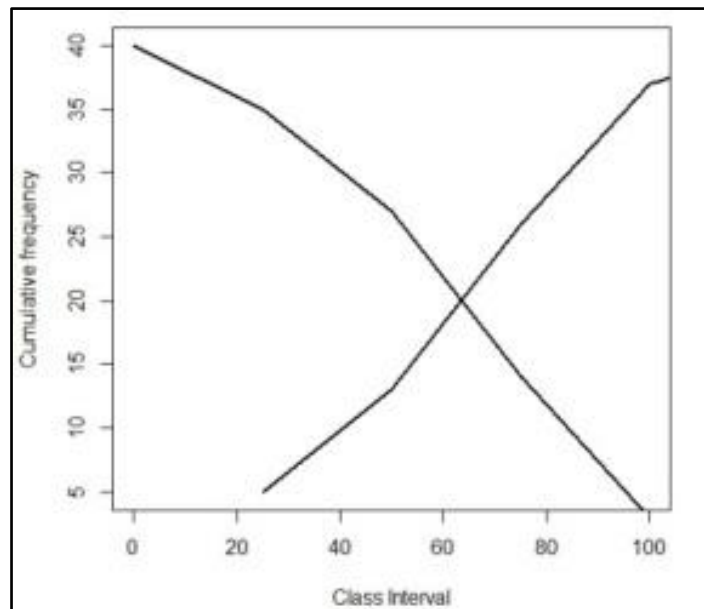
2.3.3 Ogives

C.I.	0-25	25-50	50-75	75-100	100-125
f:	5	8	13	11	3

```

> f <- c(5,8,13,11,3)
> f
[1] 5 8 13 11 3
> lc <- cumsum(f)
> lc
[1] 5 13 26 37 40
> uc <- 1:5
> uc
[1] 1 2 3 4 5
> for(i in 5:1)
+ {uc[i] <- sum(f[5:i])}
> uc
[1] 40 35 27 14 3
> lbx <- seq(0,100,25)
> lbx
[1] 0 25 50 75 100
> ubx <- seq(25,125,25)
> ubx
[1] 25 50 75 100 125
> plot(ubx,lc,type = "l",xlim = c(0,100),xlab = "Class Interval", ylab =
"Cumulative frequency",lwd =2)
> lines(lbx,uc,type = "l",xlim = c(0,100),xlab = "Class Interval", ylab =
"Cumulative frequency",lwd =2)

```



Chapter 3

Measures of Central Tendency

Mrs. Pratiksha M. Kadam, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

3.1 Introduction

According to Prof. Bowley, “Measures of central tendency (averages) are statistical constants which enable us to comprehend in a single effort the significance of the whole.” In this chapter we discuss the functions in R to calculate various measures of central tendency.

There are different types of averages.

1. Mathematical Averages:

- a. Arithmetic mean
- b. Geometric mean
- c. Harmonic mean

2. Positional Averages:

- a. Partition Values
 - Medians
 - Quartiles
 - Deciles
 - Percentiles
- b. Mode

3.2 Mathematical Averages

3.2.1 Arithmetic mean

For raw data:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Where n = the number of terms

x_i = i^{th} observation

For ungrouped frequency distribution:

$$\bar{x} = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}$$

Where n = total number of observations

x_i = i^{th} observation; f_i = frequency of i^{th} observation

For grouped frequency distribution:

$$\bar{x} = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}$$

Where x_i = mid-point of i^{th} class interval

f_i = frequency of i^{th} class

3.2.2 Geometric Mean

For raw data:

$$\bar{x} = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

Where n = the number of terms

x_i = i^{th} observation

For ungrouped frequency distribution:

$$\bar{x} = \left(\prod_{i=1}^n x_i^{f_i} \right)^{\frac{1}{N}}$$

Where $N = \sum_{i=1}^n f_i$

n = total number of observations)

x_i = i^{th} observation; f_i = frequency of i^{th} observation

For grouped frequency distribution:

$$\bar{x} = \left(\prod_{i=1}^n x_i^{f_i} \right)^{\frac{1}{N}}$$

Where $N = \sum_{i=1}^n f_i$

x_i = mid-point of i^{th} class interval

f_i = frequency of i^{th} class

3.2.3 Harmonic Mean

For raw data:

$$\bar{x} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

Where n = the number of terms

x_i = i^{th} observation

For ungrouped frequency distribution:

$$\bar{x} = \frac{N}{\sum_{i=1}^n \frac{f_i}{f_i x_i}}$$

Where $N = \sum_{i=1}^n f_i$

n = total number of observations)

x_i = i^{th} observation

f_i = frequency of i^{th} observation

For grouped frequency distribution:

$$\bar{x} = \frac{N}{\sum_{i=1}^n \frac{f_i}{f_i x_i}}$$

Where $N = \sum_{i=1}^n f_i$

x_i = mid-point of i^{th} class interval

f_i = frequency of i^{th} class

3.3 Positional Averages

3.3.1 Partition Values

a) Median

Median is the value that divides the data into two equal parts, when the data is arranged in numerical order. It is the middle value when data size N is odd. It is the mean of the middle two values, when data size N is even.

For ungrouped frequency distribution:

Find the cumulative frequencies for the data. The value of the variable corresponding to which a cumulative frequency is greater than $(N+1)/2$ for the first time. (Where f_i = frequency of i^{th} observation, $N = \sum_{i=1}^n f_i$)

For grouped frequency distribution:

First obtain the cumulative frequencies for the data. Then mark the class corresponding to which a cumulative frequency is greater than $N/2$ for the first time. Find the cumulative frequencies for the data. The value of the variable corresponding to which a cumulative frequency is greater than $(N+1)/2$ for the first time. (Where f_i = frequency of i^{th} observation, $N = \sum_{i=1}^n f_i$.) Then that class is median class. Then median is evaluated by the following formula:

$$\text{median} = l_1 + (l_2 - l_1) \left(\frac{\frac{N}{2} - cf}{f_m} \right)$$

Where $N = \sum_{i=1}^n f_i$

f_i = frequency of i^{th} class; l_1 = lower limit of the median class;

l_2 = upper limit of the median class; f_m = frequency of the median class.

cf = cumulative frequency of the class proceeding to the median class.

b) Quartiles

The data can be divided in to four equal parts by three points. These three points are known as quartiles. The quartiles are denoted by Q_i , $i = 1, 2, 3$. Q_i is the value corresponding to $(iN/4)^{\text{th}}$ observation after arranging the data in the increasing order.

For grouped frequency distribution:

First we obtain the cumulative frequencies for the data. Then mark the class corresponding to which a cumulative frequency is greater than $(iN)/4$ for the first time. (Where f_i = frequency of i^{th} observation, $N = \sum_{i=1}^n f_i$). Then that class is Q_i class. Then Q_i is evaluated by formula:

$i = 1, 2, 3$

$$Q_i = l_1 + (l_2 - l_1) \left(\frac{\frac{iN}{4} - cf}{f_q} \right)$$

Where l_1 = lower limit of the Q_i class

l_2 = upper limit of the Q_i class

cf = cumulative frequency of the class proceeding to the Q_i class.

f_q = frequency of the Q_i class.

c) Deciles

Deciles are nine points which divided the data in to ten equal parts. D_i is the value corresponding to $(iN/10)^{\text{th}}$ observation after arranging the data in the increasing order.

For grouped frequency distribution:

First obtain the cumulative frequencies for the data. Then mark the class corresponding to which a cumulative frequency is greater than $(iN)/10$ for the first time. (Where f_i = frequency of i^{th} observation, $N = \sum_{i=1}^n f_i$). Then that class is D_i class. Then D_i is evaluated by the following formula:

$$D_i = l_1 + (l_2 - l_1) \left(\frac{\frac{iN}{10} - cf}{f_d} \right)$$

$i = 1, 2, \dots, 10$.

Where l_1 = lower limit of the D_i class

l_2 = upper limit of the D_i class; f_d = frequency of the D_i class.

cf = cumulative frequency of the class proceeding to the D_i class.

d) Percentile

Percentiles are ninety-nine points which divided the data in to hundred equal parts. P_i is the value corresponding to $(iN)/100^{\text{th}}$ observation after arranging the data in the increasing order.

For grouped frequency distribution:

First obtain the cumulative frequencies for the data. Then mark the class corresponding to which a cumulative frequency is greater than $(iN)/100$ for the first time. (Where f_i = frequency of i^{th} observation, $N = \sum_{i=1}^n f_i$) Then that class is P_i class. Then P_i is evaluated by the following formula:

$$P_i = l_1 + (l_2 - l_1) \left(\frac{\frac{iN}{100} - cf}{f_p} \right)$$

Where $i=1, 2, \dots, 100$

l_1 = lower limit of the P_i class; l_2 = upper limit of the P_i class; f_p = frequency of the P_i class.

cf = cumulative frequency of the class proceeding to the P_i class;

3.3.2 Mode

The mode is the most frequent data value. Mode is the value of the variable which is predominant in the given data series. Thus in case of discrete frequency distribution, mode is the value corresponding to maximum frequency. Sometimes there may be no single mode if no one value appears more than any other. There may also be two modes (bimodal), three modes (trimodal), or more than three modes (multi-modal).

For grouped frequency distributions:

The modal class is the class with the largest frequency. After identifying modal class mode is evaluated by using interpolated formula. This formula is applicable when classes are of equal width.

$$Mode = l_1 + (l_2 - l_1) \left(\frac{d_1}{d_1 + d_2} \right)$$

Where l_1 = lower limit of the modal class

l_2 = upper limit of the modal class

$d_1 = f_m - f_0$ and $d_2 = f_m - f_1$

f_m = frequency of the modal class

f_0 = frequency of the class preceding to the modal class,

f_1 = frequency of the class succeeding to the modal class.

3.4 Calculations of Measures of Central Tendency using R

Note: In R code red coloured text denotes the code for the calculation and blue coloured text denotes the output of the code written before that statement.

For measures of central tendency, we need to install package “psych” from CRAN. Before we start executing these functions we must load package “psych”.

To install “psych” Package in R:

In R Gui, Click on Packages menu and select the option “Install package(s)”, Select 0-cloud [https] from the country options and click on OK. Then a list of functions will be displayed. From that list select function “psych” and click on Install.

To load “psych” package in R:

In R Gui, Click on Packages menu and select the option “Load package”. List of installed packages will be shown. From that list select “psych” and click on OK.

Examples solved using R

1. Given the following data about average rainfall in every month in the year of 2017.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec
Rainfall (in mm)	10	10	10	10	10	560	640	520	320	90	20	10

Calculate Arithmetic, Geometric, Harmonic mean, Median and Mode, First quartile, 56th percentile and 3rd decile for the above data.

R code:

```
> #ungrouped data
> rainfall = c(10, 10, 10, 10, 10, 560, 640, 520, 320, 90, 20, 10)
> mean(rainfall)
[1] 184.1667
> geometric.mean(rainfall)
[1] 46.69096
> harmonic.mean(rainfall)
[1] 17.92363
> median(rainfall)
[1] 15
# we define a function mode as follows:
> mode <- function(x) {
+   uniqx <- unique(x)
+   uniqx[which.max(tabulate(match(x, uniqx)))]
+ }
> mode(rainfall)
[1] 10
> quantile(rainfall, .25)
25%
10
> quantile(rainfall, .56)
56%
31.2
> quantile(rainfall, .3)
30%
10
```

2. The information about days and number of working hours for a week is given in the following table. Saturday and Sunday are holidays so working hours are not counted.

Day	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Working Hours	NA	8	6	5.5	7	4.5	NA

Calculate arithmetic, geometric, harmonic mean, median, mode, third quartile, 32nd percentile value and 8th decile of the above data.

R code:

```
> #ungrouped data with NA values
> x=c(NA, 8, 6, 5.5, 7, 4.5, NA)
> mean(x)
[1] NA
> # as NA is included mean is not calculated. We need to exclude NA values to
calculate the mean of the given data.
> mean(x, na.rm=TRUE) #na.rm represents remove NA values.
[1] 6.2
> geometric.mean(x, na.rm=TRUE)
[1] 6.081111
> harmonic.mean(x, na.rm=TRUE)
[1] 5.962573
> median(x, na.rm = TRUE)
[1] 6
> mode <- function(x) {
+   uniqx <- unique(x)
+   uniqx[which.max(tabulate(match(x, uniqx)))]
+ }
> y=na.omit(x)#to remove NA from the dataset.
> mode(y)
[1] 8
> x=c(NA, 8, 6, 5.5, 7, 4.5, NA)
> y=na.omit(x)
> quantile(y,.75)
75%
7
> quantile(y,.32)
32%
5.64
> quantile(y,.8)
80%
7.2
```

3. The table shows the scores obtained by a group of players in a test. Find the arithmetic, geometric, harmonic mean, median, mode and first quartile, 21st percentile and 6th decile of the scores.

Scores	0	1	2	3	4	5	6
Frequency	3	5	4	6	4	5	3

R code:

```

> x=c(0, 1, 2, 3, 4, 5, 6)
> f=c(3, 5, 4, 6, 4, 5, 3)
> n=sum(f)
> y=rep(x,f)
> local({pkg <- select.list(sort(.packages(all.available =
TRUE)),graphics=TRUE)
+ if(nchar(pkg)) library(pkg, character.only=TRUE)})
> mean(y)
[1] 3
> geometric.mean(y)
[1] 0
> harmonic.mean(y)
[1] 0
> median(y)
[1] 3
> mode <- function(x) {
+   uniqx <- unique(x)
+   uniqx[which.max(tabulate(match(x, uniqx)))]
+ }
> mode(y)
[1] 3
> quantile(y,.25)
25%
1.25
> quantile(y,.21)
21%
1
> quantile(y,.6)
60%

```

4. The following data represents the distribution of monthly electricity bills of the families in a society. Find Arithmetic, geometric, harmonic mean, median and mode, Q_1 , Q_3 , D_7 and P_{68} .

Bill in (Rs.)	0-200	200-400	400-600	600-800	800-1000	1000-1200	1200-1400
Frequency	1	3	11	14	9	4	2

R code:

```

> ub=c(200, 400, 600, 800, 1000, 1200, 1400)
> lb=c(0,200, 400, 600, 800, 1000, 1200)
> h=200
> x=(lb+ub)/2
> f=c(1, 3, 11, 14, 9, 4, 2)
> n=sum(f)
> am =sum(x*f)/n
> am
[1] 713.6364
> gm=10^(sum(f*log10(x))/n)
> gm
[1] 655.632
> hm=n/sum(f/x)

```

```

> hm
[1] 570.1341
> lcf=cumsum(f)
> medc=min(which(lcf>n/2))
> med=lb[medc]+(n/2-lcf[medc-1])*h/f[medc]
> med
[1] 700
> modc=which(f==max(f))
> mode=lb[modc]+h*((f[modc]-f[modc-1])/(2*f[modc]-f[modc-1]-f[modc+1]))
> mode
[1] 675
> q1c=min(which(lcf>n/4))
> q1=lb[q1c]+(n/4-lcf[q1c-1])*h/f[q1c]
> q1
[1] 527.2727
> q3c=min(which(lcf>3*n/4))
> q3=lb[q3c]+(3*n/4-lcf[q3c-1])*h/f[q3c]
> q3
[1] 888.8889
> d7c=min(which(lcf>7*n/10))
> d7=lb[d7c]+(7*n/10-lcf[d7c-1])*h/f[d7c]
> d7
[1] 840
> p68c=min(which(lcf>68*n/100))
> p68=lb[p68c]+(68*n/100-lcf[p68c-1])*h/f[p68c]
> p68
[1] 820.4444

```

3.5 References:

1. R for Beginners, Emmanuel Paradis
 2. Descriptive Statistics, Vipul Publications, Mrs. M. J. Golba.
-

Chapter 4

Measure of Dispersion

Dr. Bhagat Gayval, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

4.1 Range

It is difference between the smallest and largest values of the data. The range is the size of the smallest interval which contains all the data and provides an indication of Statistical dispersion. It is measured in the same units as the data. Since it only depends on two of the observations, it is most useful in representing the dispersion of small data sets.

Symbolically, Range=Max-Min

$$\text{Coefficient of Range} = \frac{\text{Max}-\text{Min}}{\text{Max}+\text{Min}}$$

4.2 Quartile Deviation

It is also measure of dispersion and it has cover 50% of data from all values. Quartile deviation (Q.D.) is given by formula:

$$\text{Q. D.} = \frac{1}{2} (Q_3 - Q_1)$$

$$\text{Coefficient of Q.D.} = \frac{Q_3 - Q_1}{Q_3 + Q_1}$$

Where Q_1 is the first quartile and Q_3 is the third quartile of the distribution.

4.3 Mean Deviation about 'a'

Mean deviation is useful for finding the dispersion since it's based upon all the observation and it is defined as the arithmetic mean of absolute deviations taken from any average or any value.

It is defined as follows:

$$\text{Mean Deviation about } a = \frac{1}{n} \sum_{i=1}^n |x_i - a|$$

Where 'a' can be mean or median or mode or any specified value.

In case of ungrouped/grouped frequency distribution

$$\text{Mean Deviation about } a = \frac{1}{N} \sum_{i=1}^n f_i |x_i - a|$$

Coefficient of mean deviation:

$$\text{Coefficient of mean deviation} = \frac{\text{Mean Deviation about } a}{a}$$

4.4 Variance

Variance is measures how far a data set is spread out and it is defined as the arithmetic mean of squares of deviations of the given values taken from arithmetic mean.

It is defined as

$$Var = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Where \bar{x} the mean, n is is the no. of observations of the data.

4.5 Standard Deviation

It is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the expected value of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values.

It is defined as

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\text{Coefficient of Variation (CV)} = \frac{\sigma}{\bar{x}} \times 100$$

4.6 Examples

4.6.1 Section A-Raw Data-R coding and Example

Example – Find the range, Quartile Deviation, Mean deviation about median, Variance, Standard Deviation and their coefficients for the following data-
25,29,30,17,19,30,18,28,31,33,26,28

Range and Coefficient of range (Crange)

```

> x<-c(25,29,30,17,19,30,18,28,31,33,26,28)
> r<-range(x)
> r
[1] 17 33
> diff(r)
[1] 16
> Crange =(max(x)-min(x))/(max(x)+min(x))
> Crange
[1] 0.32
#Quartile Deviation (QD) & Coefficient of QD
> QD=(quantile(x,0.75)-quantile(x,0.25))/2
> QD
> 3.25
> CoeffQD=(quantile(x,0.75)-
quantile(x,0.25))/(quantile(x,0.75)+quantile(x,0.25))
> CoeffQD
0.1214953
# Mean Deviation from median
# Library ('psych')
# mad function calculates Mean Deviation from median
> mad(x)
[1] 3.7065
> cmd=(mad(x))/(median(x))          #calculated coefficient of mean deviation
about median
> cmd
[1] 0.132375
# Variance & CV
> variance<-var(x)    #sample variance
> variance
[1] 28.87879
> CV=(sd(x)*100)/mean(x)
> CV
[1] 20.53719
# Standard Deviation(SD) & Standard Error (SE)
> SD<-sd(x) #sample standard deviation
> psd=(SD*sqrt(length(x)-1))/ sqrt(length(x))
> psd      #population standard deviation
[1] 5.145116
> cv=(psd/mean(x))*100
> cv
[1] 19.66287

```

4.6.2 Section B -ungrouped data set

Example – Find the range, Quartile Deviation, Mean deviation, Variance, Standard Deviation and their coefficients for the following data-

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frequency	5	14	21	23	60	80	86	125	112	93	56	43	32	24	22	16

Range and Coefficient of range

```

> grp=seq(0,15,by=1)
> f=c(5,14,21,23,60,80,86,125,112,93,56,43,32,24,22,16)
> data=rep(grp,f)
> r<-range(data)
> diff(r)
[1] 15
> coeffrange=(max(data)-min(data))/(max(data)+min(data))
> coeffrange
[1] 1
#Quartile Deviation (QD) & Coefficient of QD
QD=(quantile(data,0.75)-quantile(data,0.25))/2
> QD
1.625
> CoeffQD=(quantile(data,0.75)-
quantile(data,0.25))/(quantile(data,0.75)+quantile(data,0.25))
> CoeffQD
0.220339

```

Mean Deviation from median

Library ('psych')

mad function calculates Mean Deviation from median

```

> mad(data)
[1] 2.9652
> cmd=(mad(data))/(median(data)) #calculated coefficient of mean deviation
about median
> cmd
[1] 0.4236

```

Variance & Coefficient of Variance (CV)

```

> variance<-var(data)
> variance
[1] 9.548566

```

Standard Deviation(SD) & Standard Error (SE)

```

> SD<-sd(data) #sample standard deviation
> psd=(SD*sqrt(length(data)-1))/ sqrt(length(data))
> psd #population standard deviation
[1] 3.088172
> cv=(psd/mean(data))*100
> cv
[1] 40.66811

```

4.6.3 Section C -Grouped data set

Example – Find the range, Quartile Deviation, Mean absolute deviation, Variance, Standard Deviation for the following data-

Age	20-30	30-40	40-50	50-60	60-70
No. of person	25	42	28	15	10

```

> grp=seq(0,15,by=1)

```

```

> f=c(5,14,21,23,60,80,86,125,112,93,56,43,32,24,22,16)
> data=rep(grp,f)
> cmd=(mad(data))/(median(data)) #calculated coefficient of mean deviation
about median
> cmd
[1] 0.4236
> SD<-sd(data) #sample standard deviation
> psd=(SD*sqrt(length(data)-1))/ sqrt(length(data))
> psd #population standard deviation
[1] 3.088172
> cv=(psd/mean(data))*100
> cv
[1] 40.66811
> lb = seq(20,60,10)
> lb
[1] 20 30 40 50 60
> ub = seq(30,70,10)
> ub
[1] 30 40 50 60 70
> midx = (lb+ub)/2
> midx
[1] 25 35 45 55 65
> f = c(25,42,28,15,10)
> y = rep(midx,f)
> range = ub[length(ub)] - lb[1] #calculates range
> range
[1] 50
> cf = cumsum(f) #calculates cumulative frequency of greter than type
> cf
[1] 25 67 95 110 120
> q1_mincf = min(which(cf >= sum(f)/4))
> q1_mincf
[1] 2
> q1_l1 = lb[q1_mincf]; q1_l2 = ub[q1_mincf]
> q1_l1;q1_l2
[1] 30
[1] 40
> h = (q1_l2-q1_l1)
> h
[1] 10
> first_quart = q1_l1 + (h*(sum(f)/4-cf[q1_mincf-1])/f[q1_mincf])
> first_quart
[1] 31.19048
> x_bar = sum(f*midx)/sum(f)
> x_bar
[1] 40.25
> dev_mean = f * (midx - x_bar)^2
> dev_mean
[1] 5814.062 1157.625 631.750 3263.438 6125.625
> variance = sum(dev_mean)/sum(f)
> variance
[1] 141.6042

```

4.7 Skewness and Kurtosis

4.7.1 Skewness

Lack of symmetry in distribution is called as Skewness. We know that the Skewness can be positive or negative or zero. If the relation of mean>median>mode then it will be positive and curved as right tail. If the relation of mean<median<mode then it will get negative and curve as left tail. If the values of mean=median=mode then there is no Skewness.

Mathematically measures of Skewness have studied as follows:

(A) Absolute Skewness measures:

- I) Karl Person's measure of Skewness=Mean-Mode=3(Mean-Median)
- II) Bowley's measure of Skewness=(Q₃-Q₂)-(Q₂-Q₁)

(B) Relative or coefficient of Skewness measures:

- I) Karl Person's coefficient of Skewness

$$SK_P = \frac{\text{Mean} - \text{Mode}}{S.D.} = \frac{3(\text{Mean} - \text{Median})}{S.D.}$$

If SK_P>0 the curve is positively skewed, if SK_P=0 then the curve is symmetric and if SK_P<0 then the curve is said to be negatively skewed curve.

- II) Bowley's coefficient of Skewness

$$SK_B = \frac{(Q_3 + Q_1 - 2Q_2)}{(Q_3 - Q_1)}$$

If SK_B>0 the curve is positively skewed, if SK_B=0 then the curve is symmetric and if SK_B<0 then the curve is said to be negatively skewed curve.

- III) Measures based on moments

$$\beta_1 = \frac{\mu_3^2}{\mu_2^3}$$

Relative measure of Skewness

$$\gamma_1 = \pm\sqrt{\beta_1}$$

If $\gamma_1 > 0$ then the curve is positively skewed, if $\gamma_1 = 0$ then the curve is symmetric and if $\gamma_1 < 0$ then the curve is negatively skewed curve.

4.7.2 Kurtosis:

Kurtosis enables us to have an idea about the flatness or peakedness of the frequency curve. Kurtosis is measuredly compared with normal distribution. Mainly Kurtosis will be defined by three types such as Leptokurtic, Mesokurtic and Platykurtic distribution.

Mesokurtic distribution is as likely as normal distribution. In Leptokurtic distribution, the Kurtosis greater than Mesokurtic distribution and in Platykurtic distribution the Kurtosis is less than Mesokurtic distribution.

It is defined as follows:

$$\beta_2 = \frac{\mu_4}{\mu_2^2} , \quad \gamma_2 = \beta_2 - 3$$

Where Platykurtic curve is defined as $\beta_2 < 3$ or $\gamma_2 < 0$,

Leptokurtic curve is defined as $\beta_2 > 3$ or $\gamma_2 > 0$,

And Mesokurtic curve is defined as $\beta_2 = 3$ or $\gamma_2 = 0$.

4.7.3 Examples

Raw Data-R coding and Example

Example – Find the Skewness and Kurtosis and for the following data-
25,29,30,17,19,30,18,28,31,33,26,28

```
> # Karl Person's coefficient of Skewness
> x<-c(25,29,30,17,19,30,18,28,31,33,26,28)
> psd=(SD*sqrt(length(x)-1))/ sqrt(length(x))
> skp=(3*(mean(x)-median(x)))/ psd
> skp
[1] -1.859036
> #Bowley's coefficient of Skewness
> a=quantile(x,0.75); b=quantile(x,0.25); c=2*quantile(x,0.5)
> num=a+b-c; denom=a-b
> skb=num/denom; skb
-0.3846154
> # Measure based on Moments
> library(moments)
> skw=skewness(x); skw
[1] -0.6760079
> cs=sqrt(abs(skw))
> coefficient=-(cs)
> coefficient
[1] -0.822197
> # Kurtosis -based on Moments
> library(moments)
> kur=kurtosis(x)
> kur
[1] 2.068371
> coefK=kur-3
> coefK
[1] -0.9316292
```

Chapter 5

Correlation, Regression and Curve Fitting

Dr. Asha A. Jindal, Associate Professor and Head, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

5.1 Introduction

Correlation analysis is used if interest to see the existence of relationship however if nature of the relationship between each of the independent variables and dependent variable is known then regression or mathematical equation/model can be developed.

The objective of regression analysis is to investigate the relationship between interval variables. This analysis is employed to forecast/predict the worth of one variable on the idea of other variables. One will simply appreciate as most corporations and government establishments uses this statistical method to predict variables like product demand, interest rates, inflation rates costs of raw materials, labor price and so on.

This method involves developing a mathematical equation that describes the relationship between the variable to be forecast (Dependent variable) and variables that the statistical professional believes are associated with the variable (i.e independent variables/explanatory variables/regressors) and are denoted by X_1, X_2, \dots, X_k (where K is number of independent variables).

5.2 Linear regression model with one explanatory variable (Two variable regression model)

$$Y_i = \beta_0 + \beta_1 X_i + e_i$$

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

$$\hat{e}_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i$$

For given n pairs of observations on Y and X , we would like to determine the Sample Regression Function (SRF) in such a manner that sum of residuals $\sum e_i$ is as small as possible. But $\sum e_i$ gives equal weights to the residuals of smaller as well as greater magnitude which result into $\sum e_i = 0$. To avoid this method of least square is used to obtain the estimates of β_0 and β_1 . In this method $\sum e_i^2$ is minimized as squaring e_i gives more weight to the residuals of greater magnitude than that of smaller magnitude. The estimators obtain by this method also known as ordinary least square (OLS) estimators.

Correlation coefficient can be computed using the functions **cor()** or **cor.test()**:

- **cor()** computes the **correlation coefficient**
- **cor.test()** test for association/correlation between paired samples. It returns both the **correlation coefficient** and the **significance level**(or p-value) of the correlation .

5.3 Examples

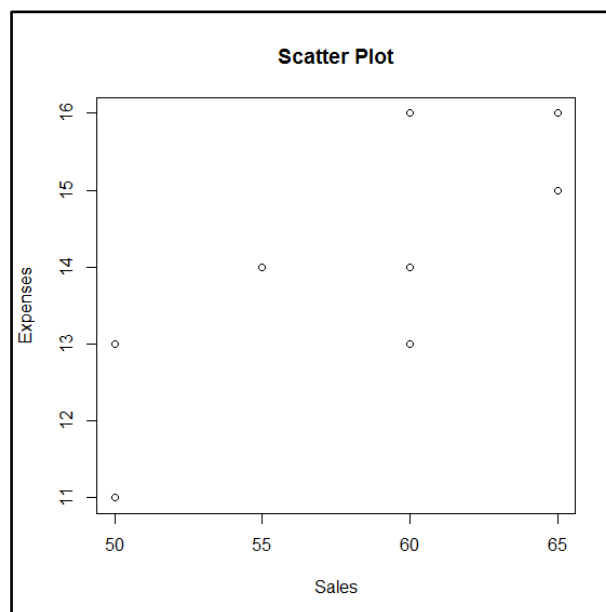
1. Plot the Scatter diagram from the following pairs of values. Find the correlation coefficient and covariance between the sales (Rs. Lakhs) and expenses (Rs. Lakhs) from the data given below :

Sales	50	50	55	60	65	65	65	60	60	50
Expenses	11	13	14	16	16	15	15	14	13	13

Also, calculate the regression equation of Sales on expenses.

Solution:

```
> x=c(50,50,55,60,65,65,65,60,60,50)
> y=c(11,13,14,16,16,15,15,14,13,13)
> plot(x,y,main="Scatter Plot",xlab="Sales",ylab="Expenses")
```



#Correlation and Covariance

#use-Specifies the handling of missing data. Options are all.obs (assumes no missing data - missing data will produce an error), complete.obs (listwise deletion) and pairwise.complete.obs (pairwise deletion)

```
> data=data.frame(x,y)
> cor(data, use="all.obs", method="pearson")
      x      y
x 1.000000 0.7865665
y 0.7865665 1.0000000
```



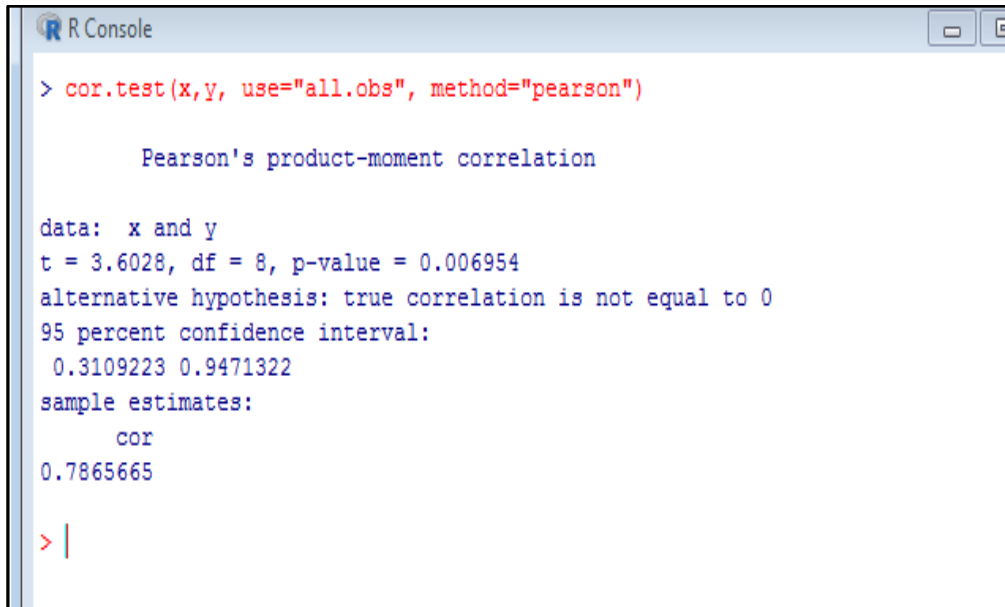
```
>cor(data, use="all.obs", method="spearman")
```

```
      x      y
x 1.0000000 0.7975307
y 0.7975307 1.0000000
```

```
>cov(data, use="complete.obs")
```

```
      x      y
x 40.000000 7.777778
y 7.777778 2.444444
```

To examine significance of correlation coefficient:



```
R Console

> cor.test(x,y, use="all.obs", method="pearson")

Pearson's product-moment correlation

data: x and y
t = 3.6028, df = 8, p-value = 0.006954
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3109223 0.9471322
sample estimates:
cor
0.7865665

> |
```

Interpretation : The **p-value** of the test is 0.006954, which is less than the significance level $\alpha = 0.05$. We can conclude that sales and expenses are significantly correlated with a correlation coefficient of 0.7865 and p-value of 0.006954.

Simple Linear Regression

```
> fit <- lm(y ~ x, data=data)
```

```
> summary(fit) # show results
```

```
Call:
```

```
lm(formula = y ~ x, data = data)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.44444 -0.38194  0.09722  0.57639  1.61111
```

```
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.72222     3.14699   0.865  0.41221
x            0.19444     0.05397   3.603  0.00695 **
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.024 on 8 degrees of freedom
```

```
Multiple R-squared:  0.6187,    Adjusted R-squared:  0.571
```

```
F-statistic: 12.98 on 1 and 8 DF,  p-value: 0.006954
```

Interpretation:**The fitted model is $y=2.7222+0.19444x$** **Since adjusted $R^2=0.57 \Rightarrow 57\%$ of variation in sales is explained by independent variable expenses.**i) $H_0 : \beta_1=0$ $H_1 : \beta_1 \neq 0$ F- statistic=12.98 and p-value=0.006954 < 0.01 , reject H_0 and conclude that $\hat{\beta}_1$ is significant at 1% l.o.s.

ii) Tests for regression coefficients

 $H_0 : \beta_0 = 0$ $H_1 : \beta_0 \neq 0$ Test statistics $t = \frac{\hat{\beta}_0}{S.E(\hat{\beta}_0)} = 0.865$, p value = 0.41212 > 0.01 \therefore Do not reject H_0 .i.e. β_0 is non significant. $H_0 : \beta_1 = 0$ $H_1 : \beta_1 \neq 0$ Test statistics $t = \frac{\hat{\beta}_1}{S.E(\hat{\beta}_1)} = 3.603$ p value = 0.00695 < 0.01 \therefore Reject H_0 . β_1 is significant**# 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

```

2. Fit a regression equation to the following data

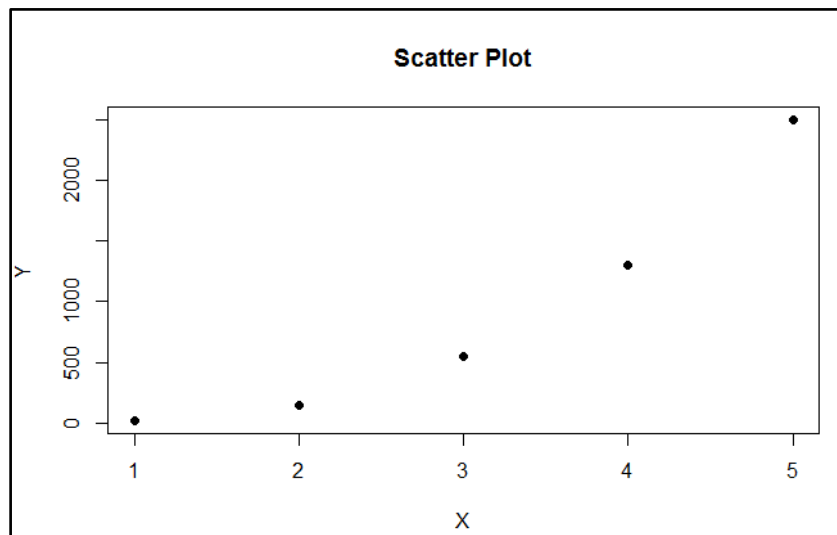
x	1	2	3	4	5
y	20	150	550	1300	2500

Simple Regression

```

> x=1:5
> y=c(20,150,550,1300,2500)
> plot(x,y,main="Scatter Plot",xlab="X",ylab="Y", pch = 20, cex = 1.5)

```

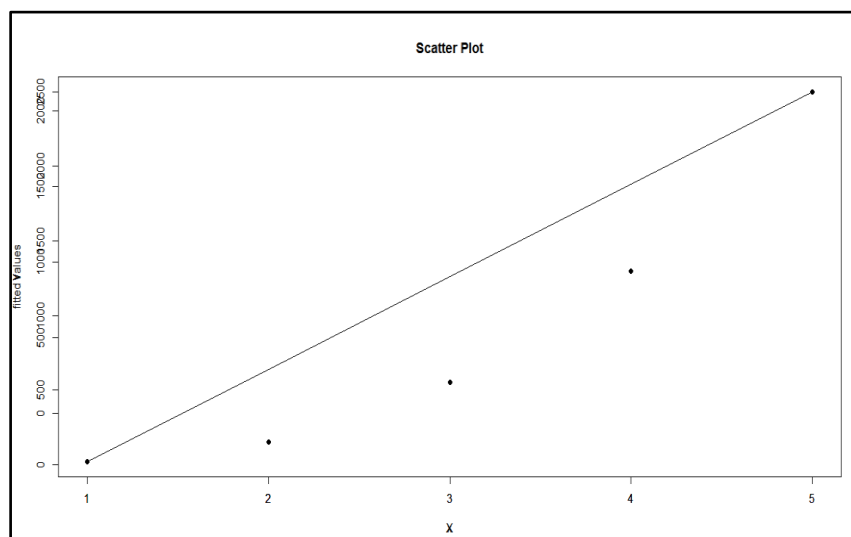


```
> r1=lm(y~x)
> r1

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
        -929         611

> par(new=TRUE)
> plot(x,r1$fitted,type="l")
```



5.4 Curve Fitting

The observed value of the two variables need not always show linear relationship between the two variables. If the scatter diagram indicates curvilinear relationship between two variables then one of the following equations may fit the given data:

- i. Quadratic Curve
- ii. Power Curve
- iii. Exponential Curve
- iv. Logarithmic Curve

Let y be the dependent variable and x be the independent variable. Assuming that paired observations on the variables x and y are available, fitting a curve is to obtain the value of constants involved in the equation by method of least squares. Using estimates of constants, equation can be obtained and hence, estimate of y from given value of x .

Fitting Quadratic Second Degree polynomial Curve:

The quadratic equation is $y = a + bx + cx^2$

5.5 Examples

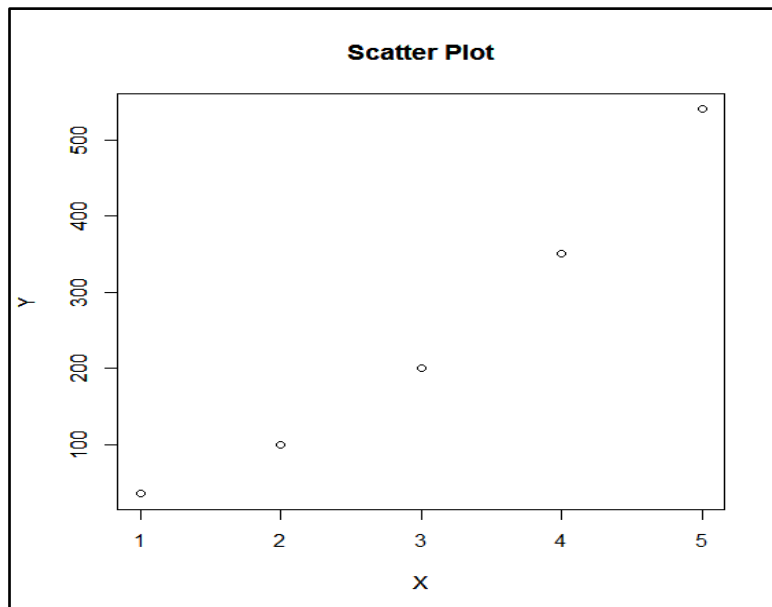
3. Fit a quadratic curve to the following data and estimate y when $x=5$.

x	1	2	3	4	5
y	35	100	200	350	540

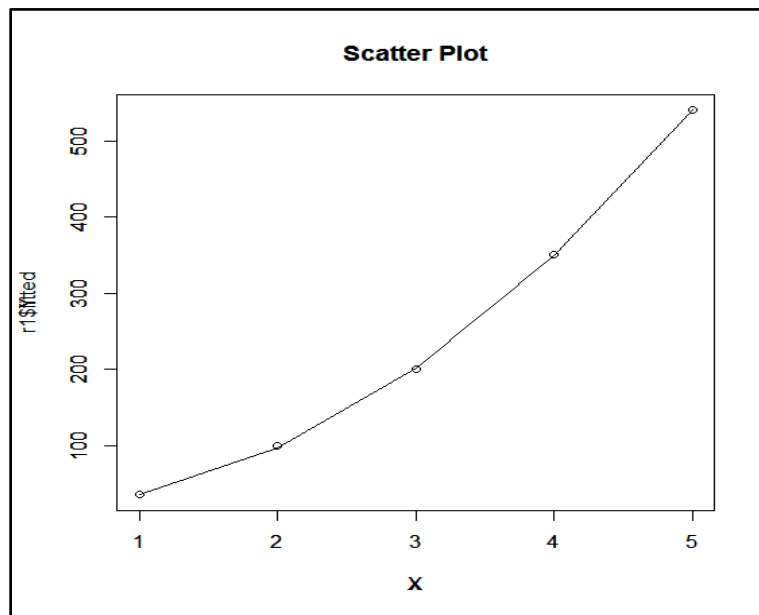
```
> x=1:5
> y=c(35,100,200,350,540)
> plot(x,y,main="Scatter Plot",xlab="X",ylab="Y")
> r1=lm(y~poly(x,2,row=TRUE))
> r1

Call:
lm(formula = y ~ poly(x, 2, raw = TRUE))

Coefficients:
      (Intercept)  poly(x, 2, raw = TRUE)1
           17.000                -2.571
poly(x, 2, raw = TRUE)2
           21.429
```



```
> par(new=TRUE)
> plot(x,r1$fitted,type="l")
```



The quadratic equation is $y=17-2.571x+21.429x^2$

When $x=5$, estimate of $y=539.87$.

Notes:

```
# Codes for Third Order polynomial
x=1:5
y=c(20,150,550,1300,2500)
plot(x,y)
r1=lm(y~poly(x,3,raw=TRUE))
r1
par(new=TRUE)
plot(x,r1$fitted,type="l")
```

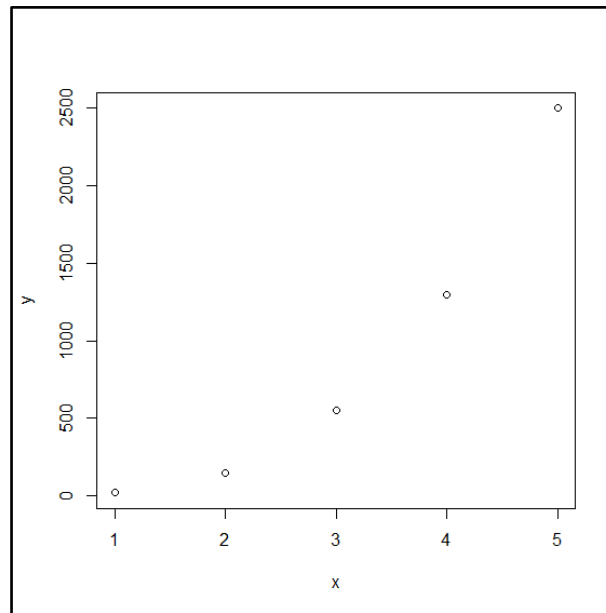
Fitting Power Curve

The power curves of the form $y=ab^x$

4. Fit a power curve to the following data and estimate y when $x=6$.

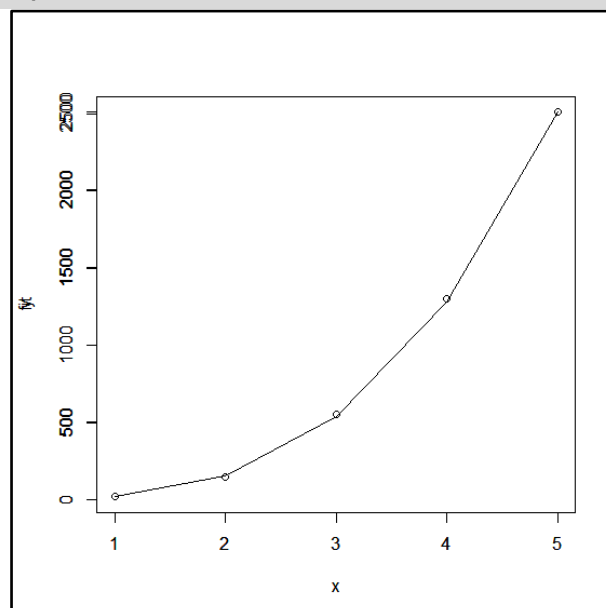
x	1	2	3	4	5
y	20	150	550	1300	2500

```
> x=1:5
> y=c(20,150,550,1300,2500)
> plot(x,y)
```



```
> r1=lm(log(y)~log(x))
> r1
Call:
lm(formula = log(y) ~ log(x))

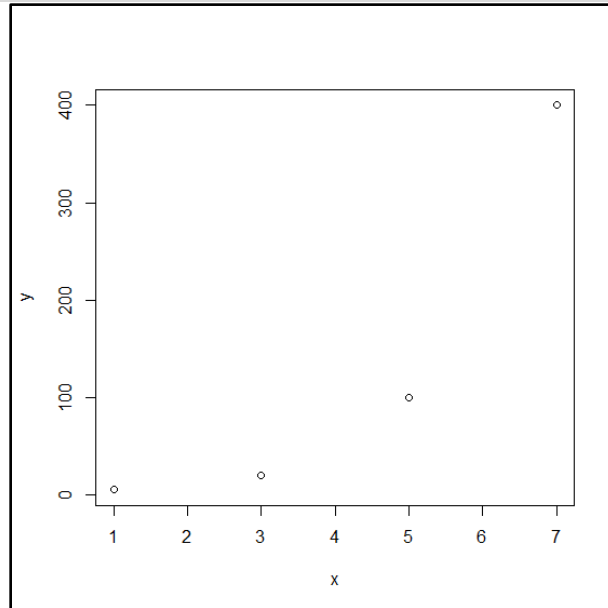
Coefficients:
(Intercept)      log(x)
      2.974         3.016
> a=exp(r1$coeff[1])
> b=r1$coeff[2]
> print(a)
(Intercept)
  19.57021
> print(b)
log(x)
 3.01627
> par(new=TRUE)
> plot(x,fit,type="l")
```



Fitting Exponential Curve**The exponential curve is of the form $y=ab^x$** **5. Fit a curve of the type $y=ab^x$ to the following data.**

x	1	3	5	7
y	5	20	100	400

```
> x=c(1,3,5,7)
> y=c(5,20,100,400)
> plot(x,y)
```

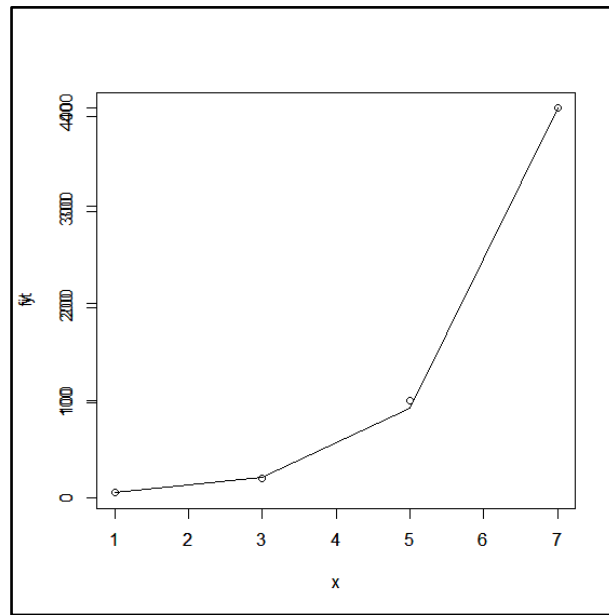


```
> r1=lm(log(y)~x)
> r1

Call:
lm(formula = log(y) ~ x)

Coefficients:
(Intercept)          x
    0.8493         0.7378

> a=exp(r1$coeff[1])
> b=exp(r1$coeff[2])
> print(a)
(Intercept)
  2.338121
> print(b)
x
2.091279
> fit=a*b^x
> par(new=TRUE)
> plot(x,fit,type="l")
```



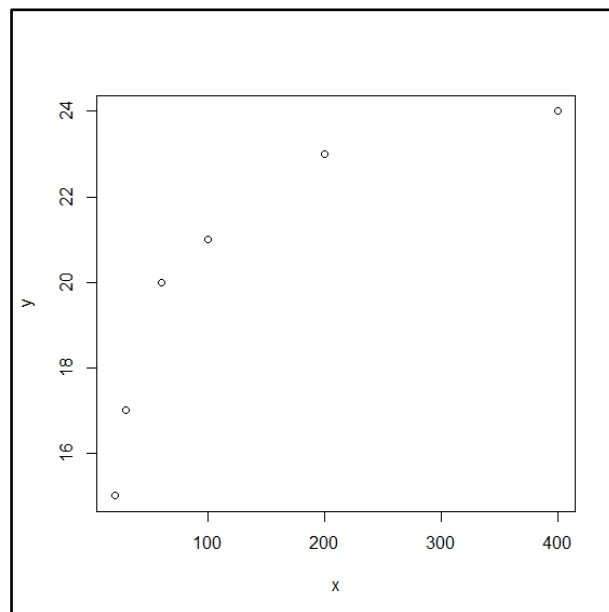
Fitting Logarithm Curve

The logarithmic curve is of the form $y = a + b \log x$

6. Fit a quadratic curve to the following data and estimate y when $x=5$.

x	20	30	60	100	200	400
y	15	17	20	21	23	24

```
> x=c(20,30,60,100,200,400)
> y=c(15,17,20,21,23,24)
> plot(x,y)
```



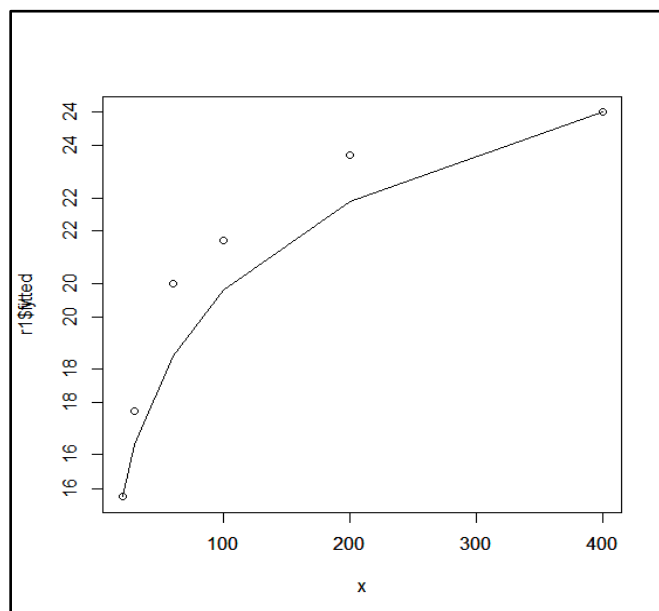
```
> r1=lm(y~log(x))
> r1
```



```
Call:
lm(formula = y ~ log(x))

Coefficients:
(Intercept)      log(x)
      6.841      2.992

> par(new=TRUE)
> plot(x,r1$fitted,type="l")
```



Chapter 6

Time Series and Forecasting Techniques using R

Mukesh Kumar Jain, CTO, VFS.GLOBAL, Mumbai, India

6.1 Background

In the bid to stay ahead of their competitors, companies are now collecting more and more data and perform deep analytics extensively and systematically to gain insights into user behaviour and provide unprecedented personalization to its users with the strategic objective of creating a distinctive competency. Today, analytics is an integral part of any business and organizations are investing in talent and upskilling them. Infact, now is a good time to get into the world of Analytics - to learn basics and start implementing, reap the benefits and ride this wave.

6.2 Analytics Concepts

Simply put, Analytics is scientific process of deriving insights from data in order to make business decisions. It involves the extensive use of data, statistical & quantitative analysis, explanatory and predictive models, as well as fact-based management to drive decision making & organisational action. With the increase in the availability of data, Analytics has now become a crucial differentiator that determines both the top line and the bottom line of any organisation.

It is commonly observed that, Big Data is synonymously used for Analytics. In fact, some people will call any form or data or reports as Analytics as it is getting popular and quickly find its way into mainstream businesses. While the concept of 'Analytics' (also known as Data Analysis in the past) has been prevalent for more than 50 years in multiple forms which even included capturing numerical data in tables, performing manual analysis to derive industry insights and market trends, 'Big Data' as a concept is relatively new which has gained traction in the past 10 years globally.

But recent developments indicate that the boundaries that define both these concepts are now blurring, with increased access to high end computing that also offers cost effective storage along with cheaper bandwidth which has now provides the opportunity to be able to make real-time analytics on large volumes of data.

6.3 Time Series

Time series data is a sequence of observations collected from a process with equally spaced periods of time.

Examples:

- Dow Jones Industrial Averages
- Daily data on sales
- Monthly inventory
- Monthly interest rates, costs
- Forecasting power consumption
- Daily closing prices of stock indices, and so on

Essential of Good time series:

- Data must be for a sufficient period
- Equal time gap
- Constant or normal period.

6.4 Importance of Time Series

There are many benefits of time series which can be written by us for business purposes

Helpful for study of past behaviour

Time series are very helpful in study of past behaviour of business. On this basis, we can invest our money in that type of business. It is duty of businessman to make time series of past sale or profit and see what is the trend of sale or profit in that type of business.

Helpful in forecasting

Forecasting is science of estimation. Today is the day of competition so if you have to win from competition then you must learn this science, this science can be utilized if we make time series and on the basis we can read the history and then we can decide what happen in future. Suppose if we can make the time series of past strategy of our competitor then on this basis we can estimate future strategy of our competitor and on this base we can change our strategy for defeating our competitor.

Helpful in evaluating the achievements

Time series is an equipment in your hand on this basis you can evaluate your business achievements if you did good, your performance shows your good face in the time series by up-word trend of your performance. If your business performance is very bad then you can make new policies to stable your business.

Helpful in comparison

If we can calculate our two or more branches time series then we can compare the performance of our branches. On their performance we can give them rewards.

6.5 Components of Time Series

6.5.1 Secular Trend(T)

Gradual long term movement (up or down). Easiest to detect
e.g.. Population growth In India

6.5.2 Cyclical Patterns(C)

Periodic in nature. An up/down repetitive movement. Repeats itself over a long period of time
e.g. Quarterly demand, things that are specific in early of the year, end of year, etc.

6.5.3 Seasonal Pattern(S)

Results from events that are periodic and recurrent in nature.
e.g. Sales in festive seasons

6.5.4 Irregular Component(I)

Erratic movements that are not predictable because they do not follow a pattern. Disturbances or residual variation that remain after all the other behaviours have been accounted for. e.g. Earthquake

6.6 Forecasting using Time Series Techniques

Creating Time Series model step by step is important to build the model and be able to compare multiple models and find the right one that accurately represent the data. Once you have the model in place, you can leverage that to forecast and continue to enhance the model.

6.6.1 Accuracy Measurements

The time series models are approximation of historic data and they are bound to have errors / deviation from actuals. While using the time series techniques, you would come up with multiple models and you would need to use techniques to measure accuracy of the model. There are multiple models to measure error and accuracy. We will use Mean Absolute Percentage (MAPE) and Root Mean Square Error (RMSE) accuracy measurement models. To calculate the accuracy of the model, the actual values are compared with the forecasted

values and overall error is error is calculated. The simplest way to remember both the accuracy models is to read it backwards and perform calculations.

a) Mean Absolute Percentage Error (MAPE)

Here are the steps to calculate MAPE, take MAPE in reverse order

1. "E" Find out **error** between forecasted and actual values
2. "P" Find out **percentage** error on actual values (error / actual value)
3. "A" Take **absolute** value of the error percentage
4. "M" Take **mean** of all the percentage error

Using this technique, we can find out the error for each of the model and compare to find the model that has the lowest MAPE value and use that model for forecasting. This model is around percentage of errors.

b) Root Mean Square Error (RMSE)

Here are the steps to calculate RMSE, take RMSE in reverse order

1. "E" Find out **error** between forecasted and actual values
2. "S" Take **square** of error
3. "M" Take **mean** of all the squared errors
4. "R" Take square **root** of all the percentage error

Using this technique, we can find out the error for each of the model and compare to find the model that has the lowest RMSE value and use that model for forecasting. This model is similar to standard deviation.

Here are the key models for time series forecasting

6.6.2 Simple Moving Average (SMA)

This is standard moving average, one can choose the time period (K) – and accordingly calculate the moving average. This is called "Simple Moving Average". The time period can be Month or Weeks or days, etc. This moving average becomes the forecast for the next period. e.g. if you choose 3 months as the period, the average of 3 months Jan-Mar will be calculated and used as forecast for the 4th month (Apr). Similarly average of Feb-Apr is calculated and used that for forecasting the values for 5th month (May), and so on.

The formula will be as follows: $F_{t+1} = (Y_t + Y_{t-1} + Y_{t-2} + Y_{t-3} + \dots + Y_{t-K+1}) / K$

Where F is the forecasted value for time period "t+1"

Y is actual value for all the time period "t", "t-1", "t-2" and so on upto "t-k+1"

And K is the time period for which we will calculate simple moving average

In simple moving average equal weightage is given to all the last "k" values

Determining the value of "k":

Short term simple moving average responds quickly to changes in data given in underlying data while Long term simple moving average are comparably slow to react.

Tips:

- If your data is more dependent on the recent values, use lower “k”.
- If your data is dependent on long term trend, have higher value of “k”.
- If you are dealing with data that repeats itself after certain interval (e.g. hourly, weekly cycle, monthly cycle, quarterly, and yearly festive season), appropriately choose the value of “K”

Challenges:

- Recent data should have more impact on forecasted sale. But Simple Moving Average assigns equal weight to recent & historical data
- Simple Moving Average does not address trending and seasonality factors in the forecasted output.

6.6.3 Weighted Moving Average (WMA)

The Weighted Moving Average is improvised version of simple moving average. This time series model is based on applying weightage on each of the data point while calculating moving average. One need to choose the time period (K) – and provide weightage for each of the data points.

Similar to simple moving average, the time period can be Month or Weeks or days, etc. This weighted moving average becomes the forecast for the next period. e.g. if you choose 3 months as the period, apply the weight for each period and then take average of the time period (e.g. 3 months Jan-Mar) will be calculated and used as forecast for the 4th month (Apr). Similarly weighted average of Feb-Apr is calculated and used that for forecasting the values for 5th month (May), and so on.

The formula will be as follows:

$$F_{t+1} = (W_t * Y_t + W_{t-1} * Y_{t-1} + W_{t-2} * Y_{t-2} + W_{t-3} * Y_{t-3} + \dots + W_{t-k+1} * Y_{t-k+1})$$

Where F is the forecasted value for time period “t+1”

W_t-Weight assigned to the values in time period “t”, “t-1”, “t-2” and so on upto “t-k+1”

Y_t- Actual value for all the time period “t”, “t-1”, “t-2” and so on upto “t-k+1”

And K is the time period for which we will calculate weighted moving average

These particular weights signify the relative importance of each term on the average. The sum of all the associated weights should be equal to 1.

Determining the value of “k”:

With weighted moving average, one can provide appropriate weights to each of the period. For e.g. one might want to provide more weightage to the most recent month (last month) compared to few months before that OR if you know the data is exhibiting a particular pattern which repeats itself every quarter, you might apply high weightage for data that is 3 months old compared to data of last month (e.g. Jan is start of quarter and Mar is end of quarter, for predicting values for the month of April, it would be highly likely to exhibit patten of Jan more than Mar (if it follows quarterly pattern). In these cases one can provide

high weightage to the 1st data point, followed by lower for the 2nd data point and then the 3rd one.

Weightage Examples:

- For value of K=3, and data that exhibits quarterly pattern, one might use the following weights:

Case	Weight1	Weight2	Weight3	Weight4	Total
1. WMA3	0.6	0.3	0.1	0.0	1.0
2. WMA3	0.2	0.3	0.5	0.0	1.0
3. WMA4	0.1	0.1	0.1	0.7	1.0
4. WMA4	0.25	0.25	0.25	0.25	1.0

- WMA3 = Weighted Moving Average of K=3
- WMA4 = Weighted Moving Average of K=4
- Case 1. WMA3 – where the first month is provided higher weightage (ideal for data that exhibits quarterly behaviour)
- Case 2. WMA3 – where the first month is provided lower weightage and higher for next month and highest for the 3rd month (ideal for data that would exhibit pattern based on the last month. If your sale is high in the last month, and you believe the data is moving / trending in certain direction – one would need to put more weightage)
- Case 3. WMA4 – where the first 3 months is provided with very lower weightage and 4th month / latest month is given highest weightage. Typically used when you want to give some weightage for earlier month and majority of dependency to the last month.
- Case 4. WMA4 – is similar to Simple Moving Average of K=4 – by distributing equal weightage.

Challenges:

- Determining the ideal weights for each of the time period could be a challenge
- (we would see how to find this automatically in R)
- Does not take into consideration learning from the last data point of estimated value vs actual value.

6.6.4 Simple Exponential Smoothing (SES)

Exponential smoothing is an adjustment technique which takes previous period's forecast, and adjusts it up or down based on what actually occurred in that period. It accomplishes this by calculating a weighted average of the two values. The formula takes the form:

$$F_{t+1} = \alpha * D_t + (1 - \alpha) * F_t$$

F_t - Forecasted Sales in t period

D_t - Actual Sales in t period

α – Data Smoothing Factor

α should lie in between 0 and 1 so that a part of difference between previous actual sale and forecasted sale is used in updating. If α is close to 1 then it has less smoothing effect and give greater weight to the recent changes in data. If α is close to 0 then it has greater smoothing effect and less responsive to recent changes in data.

Case	Exponent (0 ≤ Alpha ≤ 1)
1. SES (0.2)	0.2
2. SES (0.8)	0.8

Challenges:

- Determining the ideal value of alpha could be a challenge
- (we would see how to find this automatically in R)
- Does not take into consideration any trend in the data

6.6.5 Double Exponential Smoothing

Double exponential smoothing performs better forecasting if data has trending factor. It basically uses two parameters – α (data smoothing factor) and β (trend smoothing factor). The formula for double exponential smoothing is given by:

$$\text{when } t=1 \quad S_1=X_0, b_1=X_1-X_0$$

$$\text{when } t > 1 \quad S_t=\alpha X_t+(1-\alpha)(S_{t-1}+b_{t-1}),$$

$$b_t=\beta(S_t-S_{t-1})+(1-\beta)b_{t-1}$$

$$X_{t+1}=S_t+b_t$$

α - Data Smoothing Factor

β - Trend Smoothing Factor

S_t - Smoothing component of forecasted value

b_t - Trending component of forecasted value

X_{t+1} - Forecasted value

α (Data Smoothing) should lie in between 0 and 1

β (Trend Smoothing) should lie in between 0 and 1

Challenges:

- Determining the ideal value of alpha and beta could be a challenge
- (we would see how to find this automatically in R)
- Does not take into consideration any seasonality in the data

6.6.6 Triple Exponential Smoothing (TES)

In Triple Exponential Smoothing, it factors in multiple aspects to factor in seasonality in the data.

Triple exponential smoothing is given by the formulas:

$$S_t=\alpha X_t/C_{t-L}+(1-\alpha)(S_{t-1}+b_{t-1}),$$

$$b_t=\beta(S_t-S_{t-1})+(1-\beta)b_{t-1}$$

$$C_t = Y(X_t / S_t) + (1 - Y) * C_{t-L}$$

$$F_{t+1} = (S_t + b_t) * C_{t-L}$$

S_t - Data Smoothing Component of Forecasted Value

b_t - Trend Smoothing Component of Forecasted Value

C_t - Season Smoothing Component of Forecasted Value

α - Data Smoothing Factor ($0 < \alpha < 1$)

β - Trend Smoothing Factor ($0 < \beta < 1$)

Y - Season Smoothing Factor ($0 < Y < 1$)

Challenges:

- Determining the ideal value of alpha, beta & gamma could be a challenge
- (we would see how to find this automatically in R)

6.7 R Program for Time Series

Here is a small snapshot of sales data (Filename: "MonthlySalesData.csv")

sales
185041
183819
265239
238523
166799
210087
165960
256837
254980
191314
180391
173324

Here is the R code

```
#Code for Time Series
#Set the working directory
install.packages("forecast")
install.packages("TTR")
install.packages("Metrics")
install.packages("tseries")
library(forecast)
library(TTR)
library(Metrics)
library(tseries)
#Reading the data
sales <- read.csv("MonthlySalesData.csv")
#look at the data
View(sales)
head(sales)
tail(sales,10)
```

```

summary(sales)
#Create time series from the input data, [,1] is for first column and all
rows. freq = 12, is for 12 months. For quarters it will be freq = 4
salests <- ts(sales[,1],start=1999,freq=12)
#Let's view, what is the output
Salests
#Plot the sales data on Time Series
plot(salests)
# Simple Moving Average
sma2 <- SMA(salests,n=2)
sma3 <- SMA(salests,n=3)
sma4 <- SMA(salests,n=4)
sma24 <- SMA(salests,n=24)
write.csv(sma2,"sma2.csv")
write.csv(sma3,"sma3.csv")
write.csv(sma4,"sma4.csv")
write.csv(sma24,"sma24.csv")
rmse(sales[3:nrow(sales),] , sma2[2:(nrow(sales)-1)])
rmse(sales[4:nrow(sales),] , sma3[3:(nrow(sales)-1)])
rmse(sales[5:nrow(sales),] , sma4[4:(nrow(sales)-1)])
rmse(sales[25:nrow(sales),] , sma4[24:(nrow(sales)-1)])

# Weighted Moving Average
wma2 <- WMA(salests,n=2,c(0.3,0.7))
wma3 <- WMA(salests,n=3,c(0.2,0.3,0.5))
wma4 <- WMA(salests,n=4,c(0.1,0.2,0.3,0.4))
rmse(sales[3:nrow(sales),] , wma2[2:(nrow(sales)-1)])
rmse(sales[4:nrow(sales),] , wma3[3:(nrow(sales)-1)])
rmse(sales[5:nrow(sales),] , wma4[4:(nrow(sales)-1)])
#STL means Seasonal Trend Decomposition using Loess ("LO"cal regr"ESS"ion)
#Divides into Seasonal, Trend and Remainder.
#s.window controls how rapidly the seasonal component can change
seasonaldecom <- stl(salests, s.window="periodic")
head(seasonaldecom,24)

#Plot salests time series into three components - seasonal, trend and
remainder
plot(seasonaldecom)
monthplot(salests)

#Single Exponential Smoothing
SES<-HoltWinters(salests, alpha=0.2, beta=FALSE, gamma=FALSE)

#Predict SES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred1<-predict(SES,n.ahead=12,prediction.interval=TRUE)
salests.pred1

#Plot the base graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))

lines(SES$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred1[,1],col="blue") #Fit the future predicted values

```

```

lines(salests.pred1[,2],col="red") #Fit the upper interval
lines(salests.pred1[,3],col="red") #Fit the lower interval

#Double Exponential Smoothing to add TREND component
DES<-HoltWinters(salests, alpha=0.2, beta=0.2, gamma=FALSE)

#Predict DES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred2<-predict(DES,n.ahead=12,prediction.interval=TRUE)
salests.pred2

#Plot the graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))
lines(DES$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred2[,1],col="blue") #Fit the future predicted values
lines(salests.pred2[,2],col="red") #Fit the upper interval
lines(salests.pred2[,3],col="red") #Fit the lower interval

#Triple Exponential Smoothing to add seasonality
TES<-HoltWinters(salests, alpha = 0.2, beta=0.2, gamma=0.2)

#Predict TES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred3<-predict(TES,n.ahead=12,prediction.interval=TRUE)
salests.pred3

#Plot the graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))
lines(TES$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred3[,1],col="blue") #Fit the future predicted values
lines(salests.pred3[,2],col="red") #Fit the upper interval
lines(salests.pred3[,3],col="red") #Fit the lower interval

#Find the coefficients programatically
#Single Exponential Seasoning, coefficient tells you the level
SES_Auto<-HoltWinters(salests, beta=FALSE,gamma=FALSE)
SES_Auto$alpha

#Predict SES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred4<-predict(SES_Auto,n.ahead=12,prediction.interval=TRUE)
salests.pred4

#Plot the graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))

lines(SES_Auto$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred4[,1],col="blue") #Fit the future predicted values
lines(salests.pred4[,2],col="red") #Fit the upper interval
lines(salests.pred4[,3],col="red") #Fit the lower interval

#Put Trend component, Double exponential smoothing

```

```
DES_Auto<-HoltWinters(salests, gamma=FALSE)
DES_Auto$alpha
DES_Auto$beta

#Predict DES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred5<-predict(DES_Auto,n.ahead=12,prediction.interval=TRUE)
salests.pred5

#Plot the graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))
lines(DES_Auto$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred5[,1],col="blue") #Fit the future predicted values
lines(salests.pred5[,2],col="red") #Fit the upper interval
lines(salests.pred5[,3],col="red") #Fit the lower interval

#Automatic Triple smoothening
TES_Auto<-HoltWinters(salests)
TES_Auto$alpha
TES_Auto$beta
TES_Auto$gamma

#Predict TES, Prediction interval gives me upper and lower bound of the
confidence interval
salests.pred6<-predict(TES_Auto,n.ahead=12,prediction.interval=TRUE)
salests.pred6

#Plot the graph
plot.ts(salests, xlim = c(1999,2018),ylim=c(150000,500000))
lines(TES_Auto$fitted[,1],col="green") #Fit the historical fitted values
lines(salests.pred6[,1],col="blue") #Fit the future predicted values
lines(salests.pred6[,2],col="red") #Fit the upper interval
lines(salests.pred6[,3],col="red") #Fit the lower interval

#Creating File for final predited values
Final_Predictions<-predict(TES_Auto,36)
Final_Predictions
write.csv(Final_Predictions,"PredictedValues.csv")
```

Chapter 7

Probability and Probability Distributions

Dr. Asha A. Jindal, Associate Professor and Head, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

7.1 Probability

In real life, experiments are classified into two categories.

- Deterministics experiments
- Probabilistics experiments

In probability theory we are concerned with random experiments. The set of all possible outcomes of a random experiment is called as a sample space.

In computing probabilities of different events using R software we use function choose (n,r) which gives the value of number of combination of n objects taken r at a time(order is not important) whereas function factorial (n)/factorial (n-r) gives the value of number of n objects taken r at a time(order is important).

If a random experiment results in 'n' equally likely, mutually exclusive and exhaustive cases and if 'm' of them are favourable to the event A then the probability of event A is the ratio of m to n.

$$P(A) = \frac{m}{n} = \frac{\text{Total No.of cases favourable to event A}}{\text{Total no.of cases}}$$

1) calculate a) $^{10}C_3$ b) 8C_4 c) 9P_3 d) 5P_2 .

Solution:

```
> a1=choose (10,3)
> a1
[1] 120
> a2=choose (8,4)
> a2
[1] 70
> a3=factorial (9)/factorial (9-3)
> a3
[1] 504
> a4=factorial (5)/factorial (5-2)
> a4
[1] 20
```

2) In a group of 6 boys and 4 girls, four children are to be selected. In how many different ways can they be selected such that at least one boy should be there?

Solution:

```
> q= (choose (6,1) *choose (4,3) +choose (6,2) *choose (4,2) +choose (6,3)
*choose (4,1) +choose(6,4) *choose (4,0))
> q
[1] 209
```

3) From a group of 7 men and 6 women, five persons are to be selected to form a committee so that at least 3 men are there in the committee. In how many ways can it be done?

Solution:

```
> r =(choose (7,3) *choose (6,2) +choose (7,4) *choose (6,1) +choose (7,5)
*choose (6,0))
> r
[1] 756
```

4) In how many different ways can the letters of the word 'CORPORATION' be arranged so that the vowels always come together?

Solution:

```
> s= (factorial (7)/factorial (2) *factorial (5)/factorial (3))
> s
[1] 50400
```

5) How many 3-letter words with or without meaning, can be formed out of the letters of the word, 'LOGARITHMS', if repetition of letters is not allowed?

Solution:

```
> t=factorial (10)/factorial (10-3)
> t
[1] 720
```

6) In how many different ways can the letters of the word, 'LEADING', be arranged such that the vowels should always come together?

Solution:

```
> u=factorial (5) *factorial (3)
> u
[1] 720
```

7) How many arrangements can be made out of the letters of the word, 'ENGINEERING'?

Solution:

```
> v=factorial (11)/ (factorial (2) ^factorial (3) *factorial (3) *factorial
(2))
> v
[1] 277200
```

8) How many 6-digit telephone numbers can be formed if each number starts with 35 and no digit appears more than once?

Solution:

```
> w=factorial (8)/factorial (8-4)
> w
[1]1680
```

9) A box contains 4 red,3 white and 2 blue balls. Three balls are drawn at random. Find out the number of ways of selecting the balls of different colours?

Solution:

```
> X= (choose (4,1) *choose (3,1) *choose (2,1))
> x
[1] 24
```

10) What is the probability of drawing two Ace cards from well shuffled pack of 52 playing cards?

Solution:

```
>y= (choose (4,2)/choose (52,2))
>y
[1] 0.004524887
```

11) A box contains 5 red and 7 blue marbles.A sample of 4 is drawn at random what is probability of selecting at least two blue marbles?

Solution:

```
>z=(choose (5,2) *choose (7,2) +choose (5,1) *choose (7,3) +choose (5,0)
*choose (7,4))/ (choose(12,4))
> z
[1] 0.8484848
```

7.2 Probability Distributions

Binomial Distribution

R supports following functions related to binomial distribution with specified parameters.

<code>dbinom(x,n,p)</code>	It gives individual binomial probability at $X=x$.
<code>pbinom(x,n,p)</code>	It gives cumulative binomial probability function. $P(X \leq x)$.
<code>qbinom(x,n,p)</code>	It gives quantile function.
<code>rbinom(m,n,p)</code>	It generates a random sample of size m from binomial distribution.

Similar functions starting with letter d, p, q and r are used in connection with different distributions.

Following are some commonly used distributions with their R names.

Distributions	R name	Additional Arguments
Binomial	binom	Size, probability
Poisson	Pois	Parameter lambda
Hypergeometric	hyper	M, N-M, n
Geometric	geom	probability
Negative Binomial	nbinom	Size, probability
Uniform	unif	min, max
Exponential	exp	rate
Normal	norm	mean, sd
Log--normal	lnorm	meanlog, sdlog
Cauchy	cauchy	location, scale
Gamma	gamma	shape, scale
Beta	beta	shape1, shape2, ncp
Student's t	t	df, ncp
F	f	df1, df2, ncp
Chi-square	chisq	df, ncp
Logistic	logis	location, scale
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m. n

1) If $X \sim \text{Bino}(10, 0.6)$. Find a) $P(X=0)$ b) $P(X=2)$ c) $P(X \leq 3)$ d) $P(X > 5)$

Solution:

Given : $X \sim \text{Bin}(n=10, p=0.6)$

```
> a1=dbinom (0,10,0.6)
> a1
[1] 0.0001048576
b) P(X=2)
> b1=dbinom (2,10,0.6)
> b1
[1] 0.01061683
c) P(X<=3)
> c1=pbinom (3,10,0.6)
> c1
[1] 0.05476188
d) P(X>5)
> d1=1-pbinom (5,10,0.6)
> d1
[1] 0.6331033
```


2) If $X \sim P(3.2)$. Find a) $P(X=0)$ b) $P(X=3)$ c) $P(X=5)$ d) $P(X \leq 1)$ e) $P(X > 3)$ f) $P(X \geq 5)$.

Solution:

```
> X~P(m=3.2)
> a1=dpois (0,3.2)
> a1
[1] 0.0407622
> b1=dpois (3,3.2)
> b1
[1] 0.222616
> c1=dpois (5,3.2)
> c1
[1] 0.1139794
> d1=ppois (10,3.2)
> d1
[1] 0.9995028
> e1=1-ppois (3,3.2)
> e1
[1] 0.3974803
> f1=1-ppois (5,3.2)
> f1
[1] 0.1054081
```

3) If $X \sim \text{HyperGeo}(N=25, M=5, n=3)$.

Find a) $P(X=0)$ b) $P(X=2)$ c) $P(X=5)$ d) $P(X \leq 1)$ e) $P(X > 3)$ f) $P(X \geq 2)$.

Solution:

Given : $X \sim \text{HyperGeo}(N = 25, M = 5, n = 3)$

```
> a1=dhyper (0,5,20,3)
> a1
[1] 0.4956522
> b1=dhyper (2,5,20,3)
> b1
[1] 0.08695652
> c1=dhyper (5,5,20,3)
> c1
[1] 0
> d1=phyper (1,5,20,3)
> d1
[1] 0.9086957
> e1=1-phyper (3,5,20,3)
> e1
[1] 0
> f1=1-phyper (2,5,20,3)
> f1
[1] 0.004347826
```

4) Plot probability mass function (pmf) and distribution function for the following random variables a) $X \sim P(2.6)$ b) $X \sim \text{Bino}(8, 0.65)$ c) $X \sim \text{HyperGeo}(N=50, M=10, n=7)$

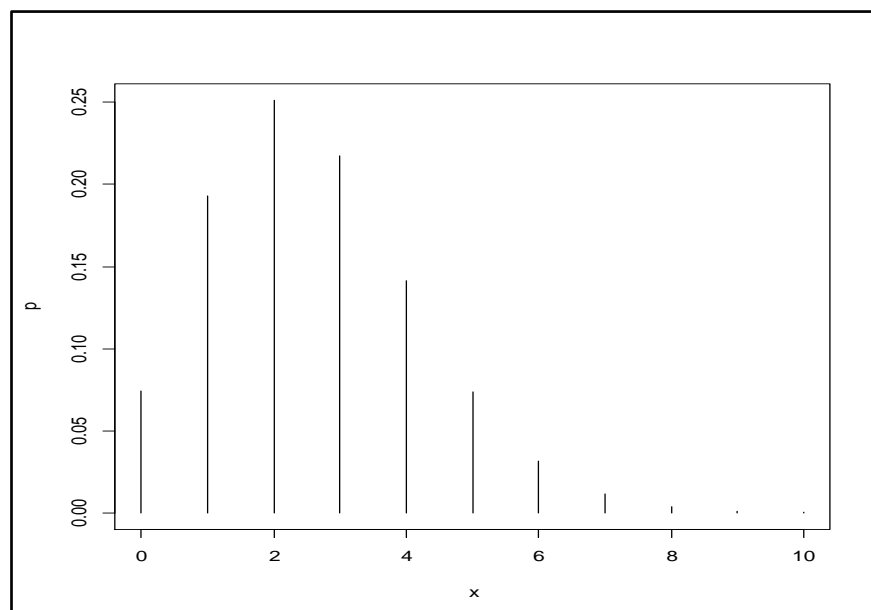
Solution:

a) $X \sim P(2.6)$

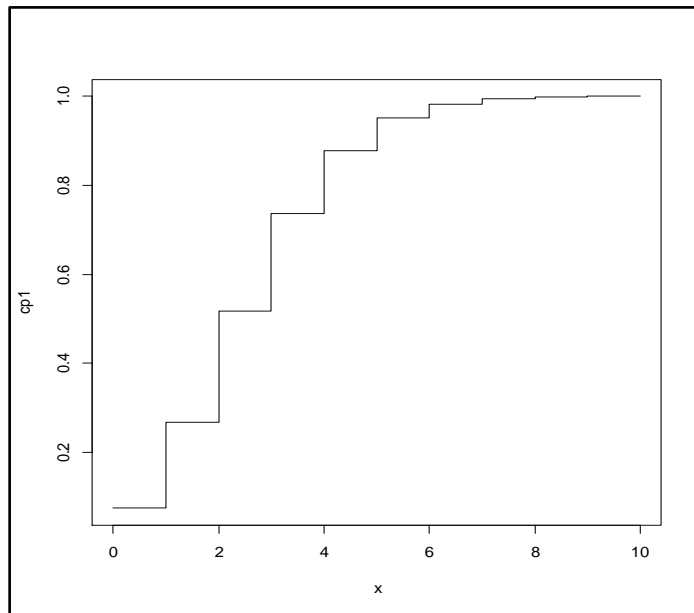
```
> m=2.6
> x=0:10
> p=dpois(x, m)
> d=data.frame(x, p)
> d
```

	x	p
1.	0	0.0742735782
2.	1	0.1931113034
3.	2	0.2510446944
4.	3	0.2175720684
5.	4	0.1414218445
6.	5	0.0735393591
7.	6	0.0318670556
8.	7	0.0118363349
9.	8	0.0038468089
10.	9	0.0011113003
11.	10	0.0002889381

```
> plot(x, p, "h")
```



```
> cp=ppois(x, m)
> cp1=round(cp,4)
> d1=data.frame(x, cp1)
> plot(x, cp1, "s")
```

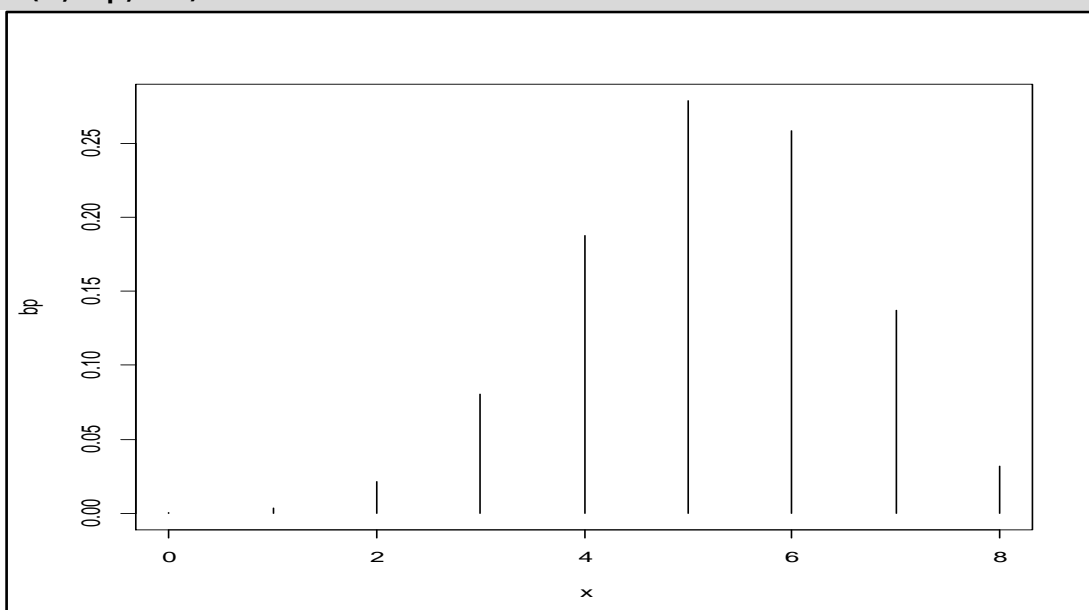


b) $X \sim \text{Bino}(8, 0.65)$

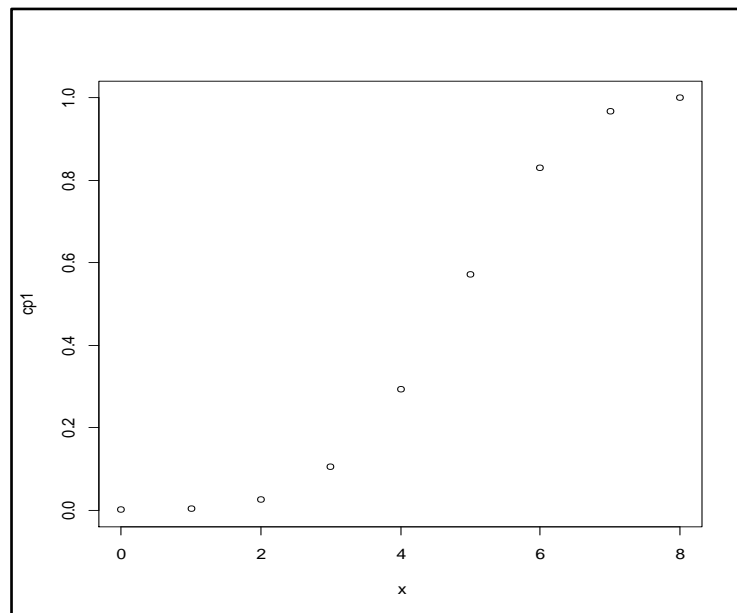
```
> n=8; p=0.65
> x=0: n
> bp=dbinom(x, n, p)
> d=data.frame("x-values"=x,"probabilities"=bp)
> d
```

	x-values	probabilities
1	0	0.0002251875
2	1	0.0033456434
3	2	0.0217466823
4	3	0.0807733916
5	4	0.1875096590
6	5	0.2785857791
7	6	0.2586867948
8	7	0.1372623809
9	8	0.0318644813

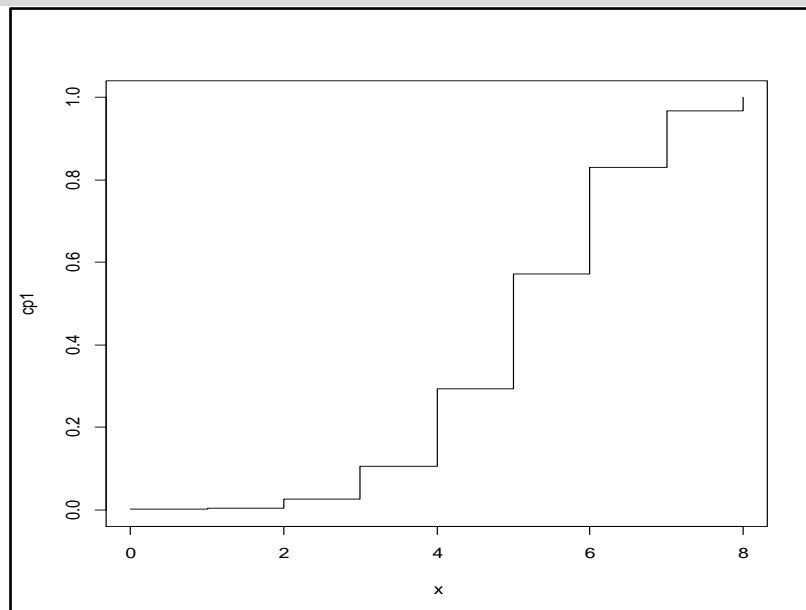
```
> plot (x, bp,"h")
```



```
> cp=pbinom (x, n, p)
> cp1=round (cp,4)
> d1=data.frame(x, cp1)
> plot (x, cp1)
```



```
> plot (x, cp1,"s")
```



c) $X \sim \text{HyperGeo}(N=50, M=10, n=7)$

```
> N=50; M=10; n=7
> x=0: n
> hp=dhyper (x, M, N-M, n)
> d=data.frame(x, hp)
> d
```

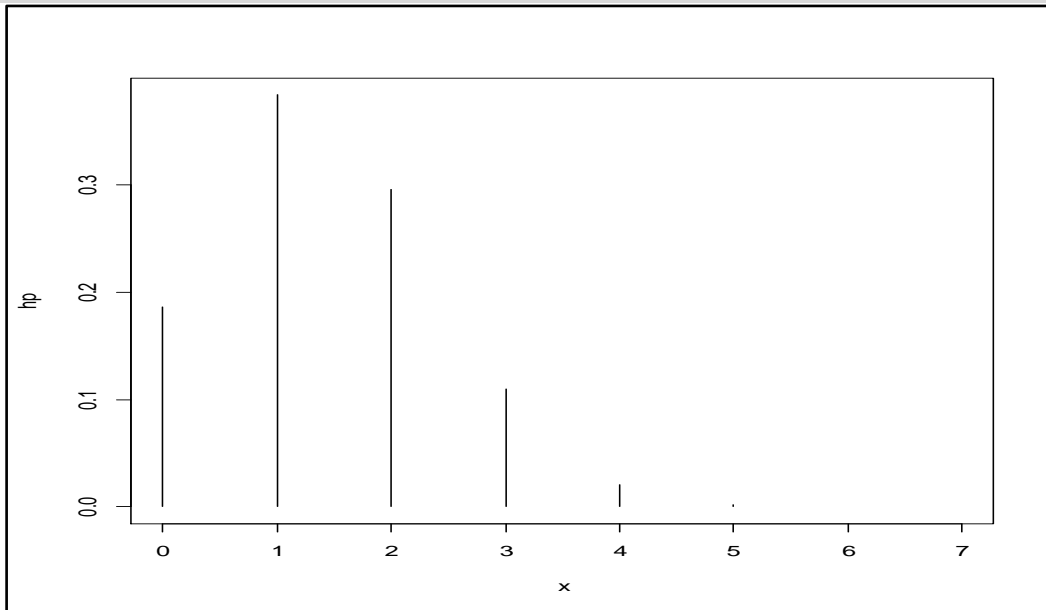
	x	hp
1	0	1.866514e-01
2	1	3.842822e-01
3	2	2.964463e-01
4	3	1.097949e-01

```

5      4  2.077201e-02
6      5  1.967875e-03
7      6  8.409722e-05
8      7  1.201389e-06

```

```
>plot (x, hp,"h")
```



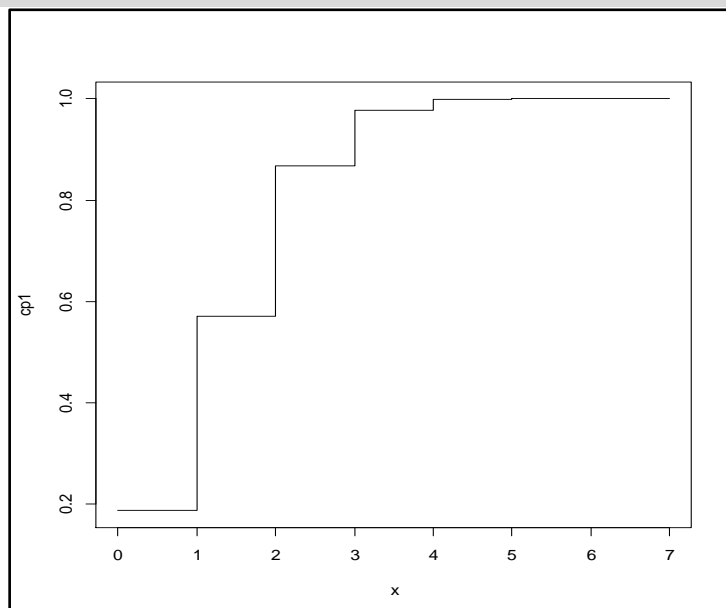
```

> cp=phyper (x, M, N-M, n)
> cp1=round(cp,4)
> di=data.frame(x, cp1)
> di

```


	x	cp1
1	0	0.1867
2	1	0.5709
3	2	0.8674
4	3	0.9772
5	4	0.9979
6	5	0.9999
7	6	1.0000
8	7	1.0000

```
>plot (x, cp1,"s")
```



Heads:	0	1	2	3	4	5	6
Frequency:	7	64	140	210	132	75	12

	x	f	expected.frequency
1	0	7	9
2	1	64	56
3	2	140	145
4	3	210	200
5	4	132	154
6	5	75	64
7	6	12	11



A scatter plot showing the relationship between f (x-axis) and $ef1$ (y-axis). The x-axis ranges from 0 to 200 with major ticks every 50 units. The y-axis ranges from 0 to 200 with major ticks every 50 units. There are 10 data points plotted as 'x' marks. A solid black line represents the linear regression fit, starting at the origin (0,0) and extending to approximately (210, 205). The data points are approximately at (5, 10), (10, 15), (65, 55), (75, 65), (130, 155), (140, 145), and (205, 205).

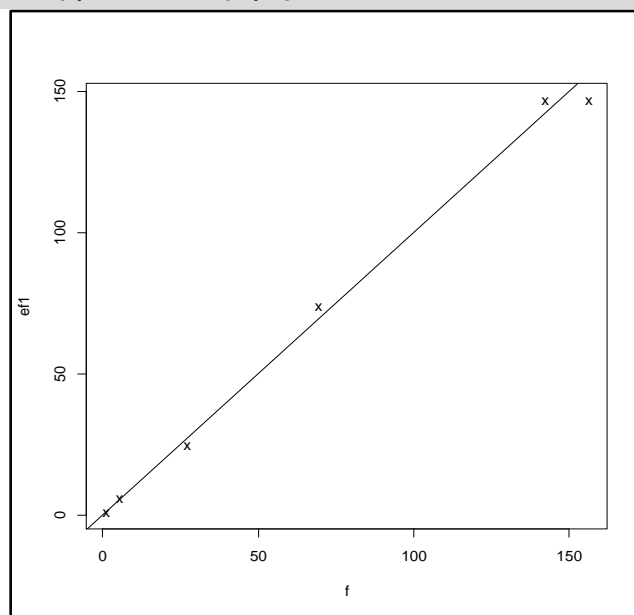
f	ef1
5	10
10	15
65	55
75	65
130	155
140	145
205	205

69

```

> m=sum(x*f)/sum(f)
> px=dpois (x, m)
> px=round(px,4)
> ef=sum(f)*px
> ef1=round(ef,0)
> d=data.frame(x, f,"expected frequency"=ef)
> d
  x    f expected.frequency
1  0  142             147.16
2  1  156             147.16
3  2   69             73.56
4  3   27             24.52
5  4    5              6.12
6  5    1              1.24
> plot (f, ef1, pch="x"); abline (0,1)

```



7) Plot the pmf of a) $X \sim \text{Bino}(30, 0.05)$ b) $X \sim P(1.5)$ and comment on graph

Solution:

Given: $X \sim \text{Bino}(30, 0.05)$

```

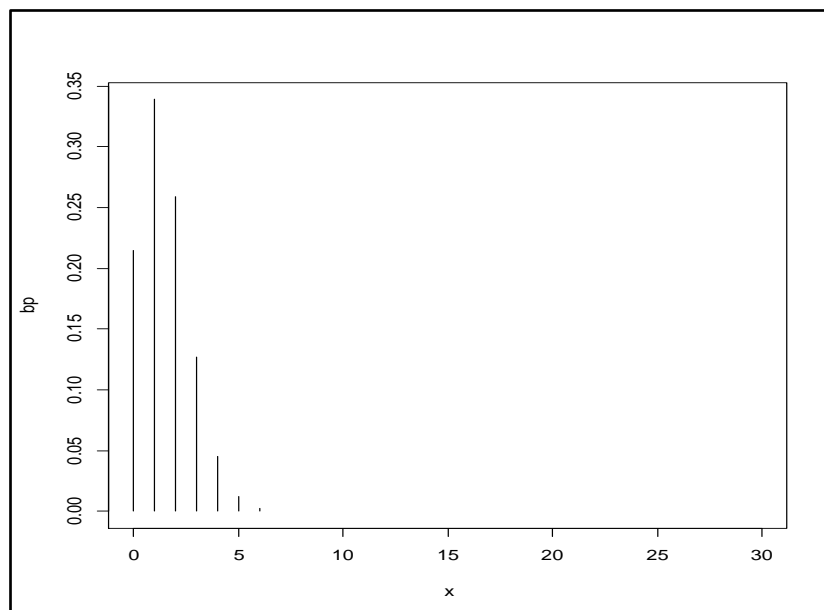
> n=30; p=0.05
> x=0: n
> bp=dbinom (x, n, p)
> d=data.frame("x-values"=x,"probabilities"=bp)
> d
  x.values probabilities
1         0  2.146388e-01
2         1  3.389033e-01
3         2  2.586367e-01
4         3  1.270496e-01
5         4  4.513605e-02
6         5  1.235302e-02
7         6  2.708997e-03
8         7  4.888415e-04
9         8  7.396944e-05

```

```

10      9      9.516536e-06
11     10     1.051828e-06
12     11     1.006534e-07
13     12     8.387780e-09
14     13     6.112552e-10
15     14     3.906518e-11
16     15     2.193133e-12
17     16     1.082138e-13
18     17     4.690382e-15
19     18     1.782894e-16
20     19     5.926516e-18
21     20     1.715570e-19
22     21     4.299675e-21
23     22     9.257674e-23
24     23     1.694769e-24
25     24     2.601619e-26
26     25     3.286255e-28
27     26     3.326169e-30
28     27     2.593504e-32
29     28     1.462502e-34
30     29     5.308539e-37
31     30     9.313226e-40
> plot (x, bp,"h")

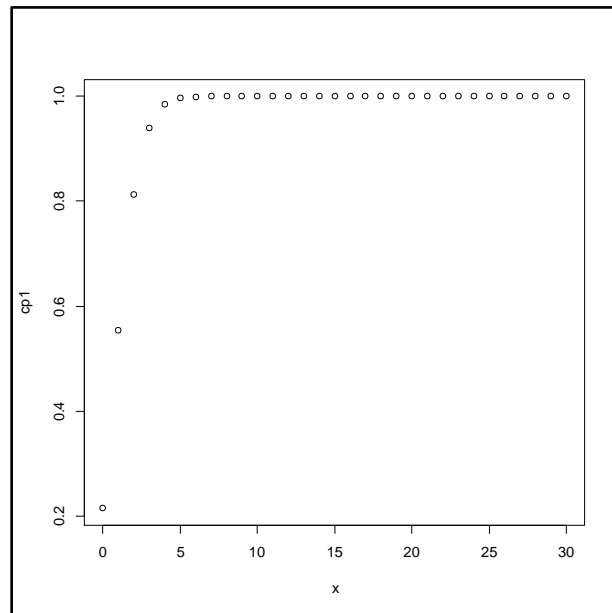
```



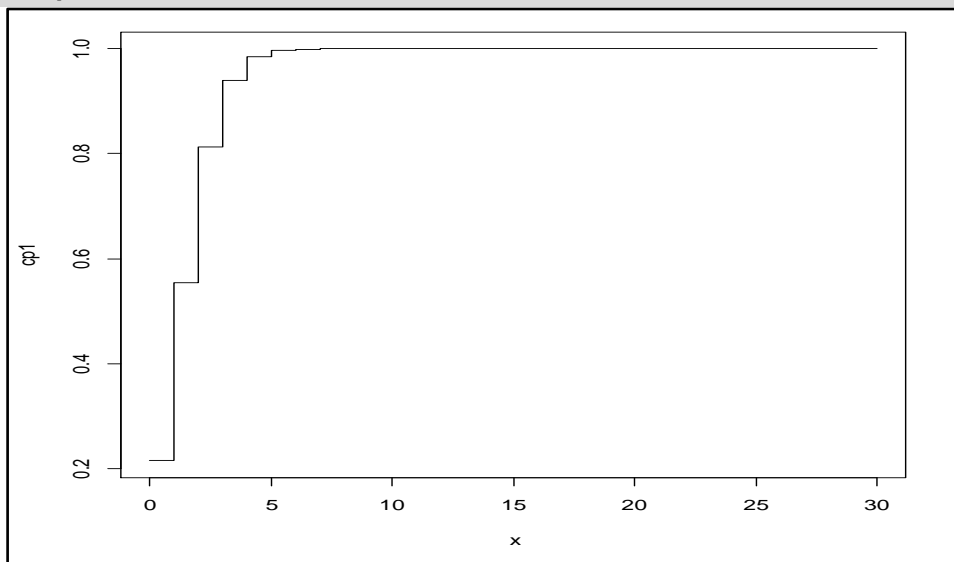
```

> cp=pbinom (x, n, p)
> cp1=round (cp,4)
> d1=data.frame(x, cp1)
> plot (x, cp1)

```

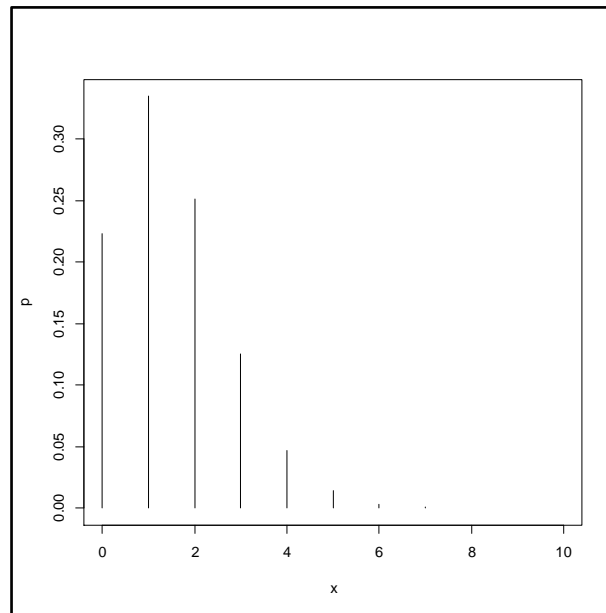
```
> plot (x, cp1,"s")
```



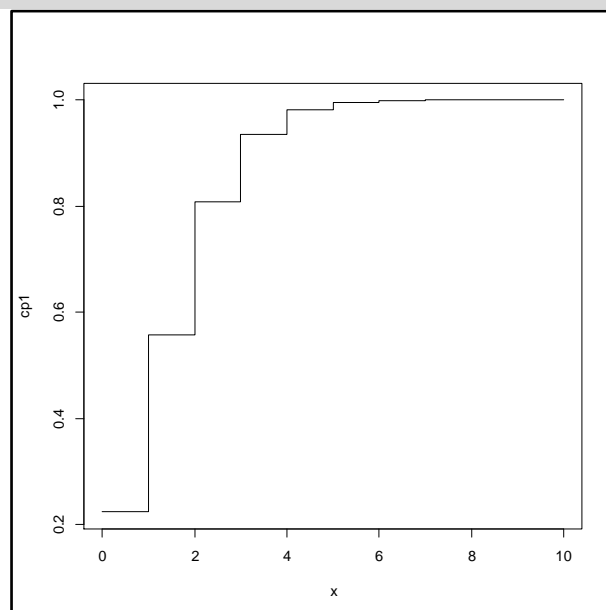
```
> # b]X~P (1.5) #
> m=1.5
> x=0:10
> p=dpois (x, m)
> d=data.frame(x, p)
> d
```

	x	p
1	0	2.231302e-01
2	1	3.346952e-01
3	2	2.510214e-01
4	3	1.255107e-01
5	4	4.706652e-02
6	5	1.411996e-02
7	6	3.529989e-03
8	7	7.564262e-04
9	8	1.418299e-04
10	9	2.363832e-05
11	10	3.545748e-06

```
> plot (x, p,"h")
```



```
> cp=ppois (x, m)
> cp1=round(cp,4)
> d1=data.frame(x, cp1)
> plot (x, cp1,"s")
```



8) $X \sim \text{Negative Bin}(r=2, P= 0.05)$ then compute

- $P(X=0), P(X=1), P(X \leq 1), P(X \geq 2)$
- Evaluate Nbinomial probabilities and plot the graph of p.m.f and c.d.f.

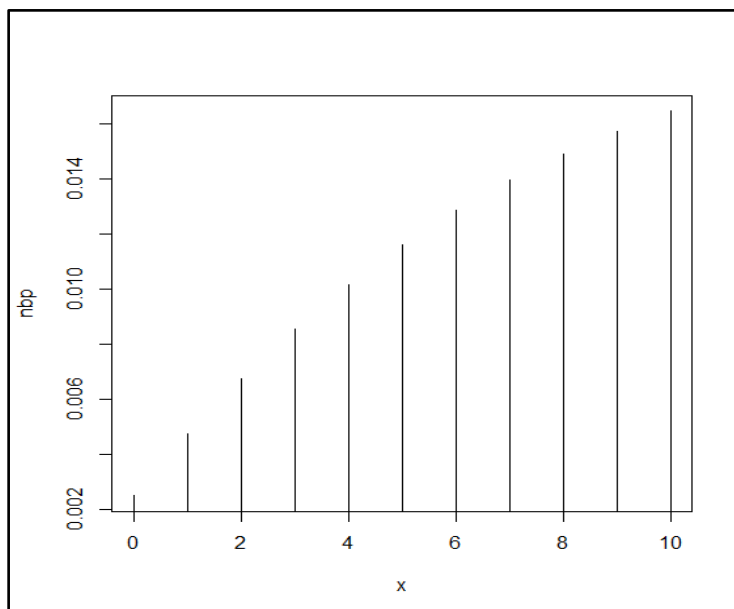
Solution:i.

```
> dnbinom(0,2,0.05)
[1] 0.0025
> dnbinom(1,2,0.05)
[1] 0.00475
> pnbinom(1,2,0.05)
[1] 0.00725
> 1-pnbinom(1,2,0.05)
[1] 0.99275
ii)
```

```

> p=0.05;r=2
> x=0:10
> nbp=dnbinom(x,r,p)
> d=data.frame("X-Value"=x,"Probability"=nbp)
> d
      X.X.Value      Probability
1            0      0.00250000
2            1      0.00475000
3            2      0.00676875
4            3      0.00857375
5            4      0.01018133
6            5      0.01160671
7            6      0.01286411
8            7      0.01396675
9            8      0.01492696
10           9      0.01575624
11          10      0.01646527
> plot(x,nbp,"h")

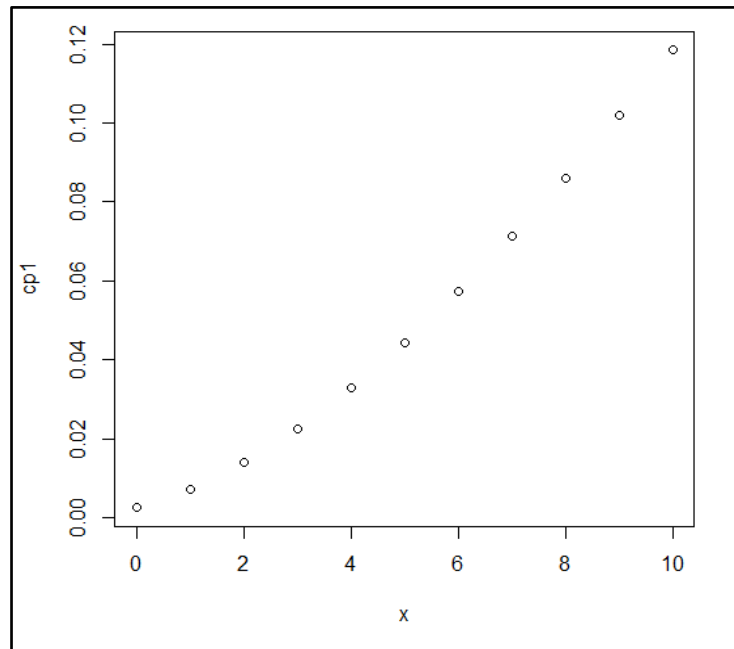
```



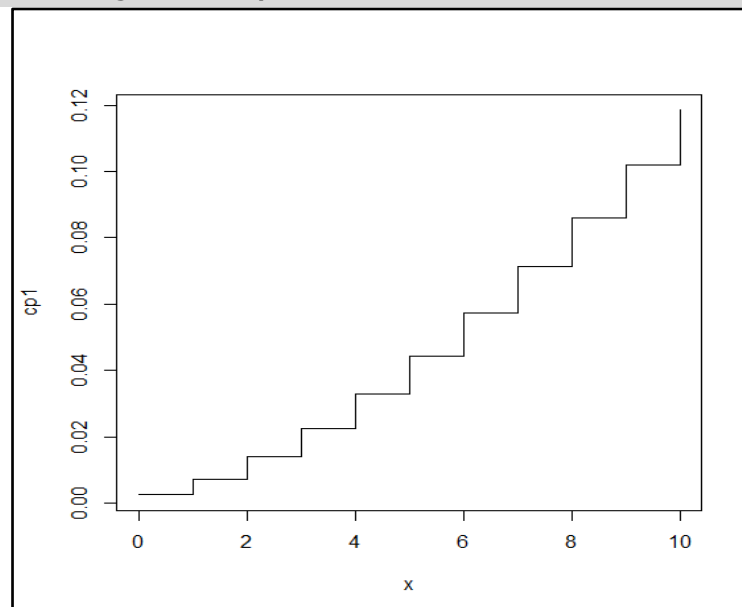
```

> cp1=round(cp,4)#round function round off cp values upto 4 decimal
> d1= data.frame(x,cp1)
> d1
      x      cp1
1     0  0.0025
2     1  0.0073
3     2  0.0140
4     3  0.0226
5     4  0.0328
6     5  0.0444
7     6  0.0572
8     7  0.0712
9     8  0.0861
10    9  0.1019
11   10  0.1184
> plot(x,cp1) #Just points are plotted

```



```
> plot(x,cp1,"s") #It gives step function
```



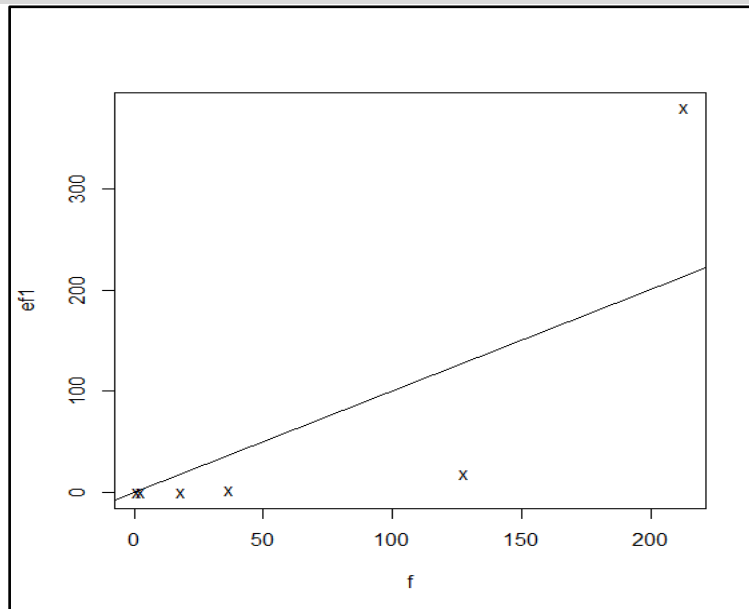
9) Fit the Negative Binomial Distribution to following data:

X:	0	1	2	3	4	5
f:	213	128	37	18	4	5

Solution:

```
> x=0:5; f=c(213,128,37,18,3,1)
> m=sum(f*x)/sum(f)
> var=(sum(f*x*x)/sum(f))-m*m
> p=m/var;q=1-p;r=m*p/q
> px=dnbinom(x,r,p)
> px1=round(px,5)
> ef=sum(f)*px1
> ef1=round(ef,0)
> d=data.frame(x,f,"exp.freq."=ef1)
```

```
> d
      x      f  exp.freq
1     0    213      379
2     1    128       19
3     2     37        2
4     3     18         0
5     4      3         0
6     5      1         0
> plot(f,efl,pch="x");abline(0,1) #pch gives the point markers
```



10) Let $X \sim N(50, 40)$. Find $P(X \leq 60)$, $P(X \geq 100)$, $P(10 \leq X \leq 20)$ and $P(X \leq k) = 0.293$.

Solution:

```
> mu=50; sd=sqrt(40)
> p1=pnorm(60,mu,sd)
> p1
[1] 0.9430769
> p2=1-pnorm(100,mu,sd)
> p2
[1] 1.332268e-15
> p3=pnorm(20,mu,sd)-pnorm(10,mu,sd)
> p3
[1] 1.050591e-06
> p4=qnorm(0.293,mu,sd)
> p4
[1] 46.55538
```

11) Fit a normal distribution to the following data of height (in cms) of 200 Indian adult males

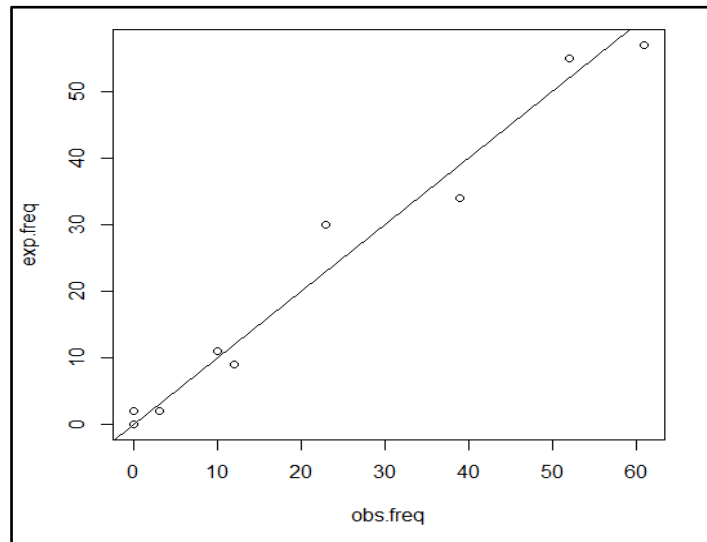
Height in cms	144-150	150-156	156-162	162-168	168-174	174-180	180-186
No of Adults	3	12	23	52	61	39	10

Solution:

```

> l1=seq(144,180,6)
> u1=seq(150,186,6)
> f=c(3,12,23,52,61,39,10)
> x=(l1+u1)/2
> n=sum(f)
> k=length(f)
> m=sum(f*x)/n;v=sum(f*(x-m)^2)/n;sd=sqrt(v)
> l1=c(-9999,l1,186)
> cp=pnorm(l1,m,sd)
> p=diff(cp)
> p=c(p,1-cp[k+2])
> u1=c(144,u1,9999);f=c(0,f,0)
> ef=round(n*p,0)
> d=data.frame("Lower Limit"=l1,"Upper Limit"=u1,"Obs.freq"=f,"prob"=p,"cum
prob"=cp,"expfreq"=ef)
> d
  Lower.Limit Upper.Limit Obs.freq prob      cum.prob expfreq
1      -9999         144         0 0.0009277682 0.0000000000      0
2         144         150          3 0.0085408285 0.0009277682      2
3         150         156         12 0.0474590553 0.0094685967      9
4         156         162         23 0.1504843558 0.0569276520     30
5         162         168         52 0.2727415211 0.2074120077     55
6         168         174         61 0.2828190953 0.4801535289     57
7         174         180         39 0.1677990586 0.7629726242     34
8         180         186         10 0.0569156032 0.9307716828     11
9         186        9999          0 0.0123127140 0.9876872860      2
> plot(f,ef,xlab="obs.freq",ylab="exp.freq","p")
> abline(0,1)

```



12) Find a) $P(X \leq 0.8)$ b) $P(X > 0.5)$

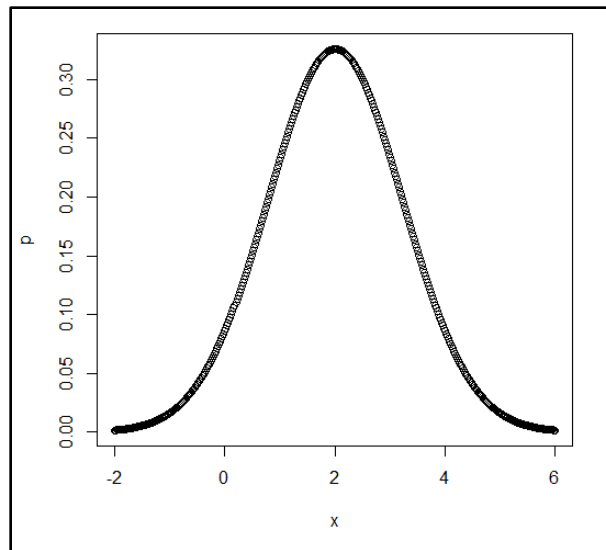
If, i. $X \sim \text{Normal}(2, 1.5)$ ii. $X \sim \text{Normal}(0, 1)$ iii. $X \sim \text{Exp}(1.5)$ iv. $X \sim \text{beta}(2, 1.5)$

v. $X \sim \text{Gamma}(2, 1.5)$ vi. $X \sim \text{ChiSq}(10)$ vii. $X \sim t(8)$ viii. $X \sim F(10, 10)$

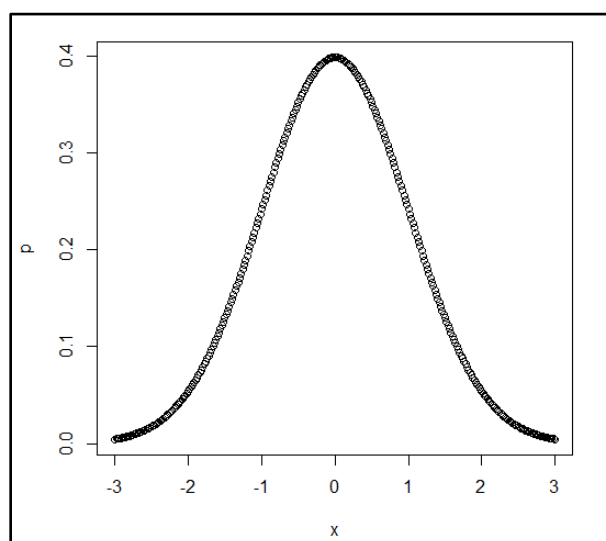
ix. $X \sim U(0, 5)$

Solution:

```
> a=pnorm(0.8,2,sqrt(1.5),lower.tail=1)
> a
[1] 0.1635934
> b=pnorm(0.5,2,sqrt(1.5),lower.tail=0)
> b
[1] 0.8896643
> x=seq(-2,6,by=0.02)
> p=dnorm(x,2,sqrt(1.5))
> plot(x,p)
```



```
> a=pnorm(0.8,0,sqrt(1),lower.tail=1)
> a
[1] 0.7881446
> b=pnorm(0.5,0,sqrt(1),lower.tail=0)
> b
[1] 0.3085375
> x=seq(-3,3,by=0.02)
> p=dnorm(x,0,sqrt(1))
> plot(x,p)
```

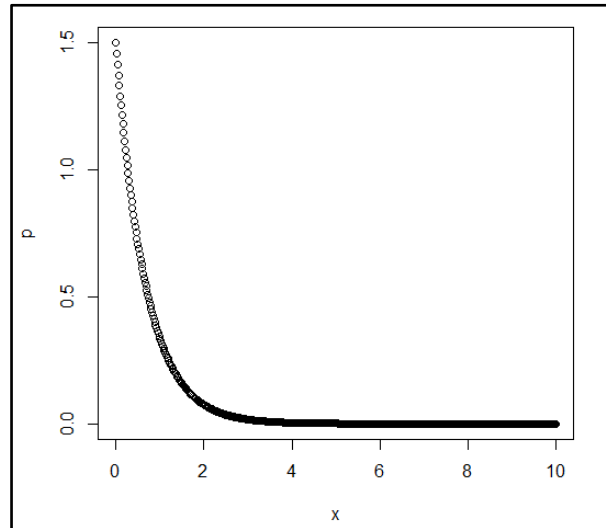


```
> a=pexp(0.8,1.5,lower.tail=1)
```

```

> a
[1] 0.6988058
> b=pexp(0.5,1.5,lower.tail=0)
> b
[1] 0.4723666
> x=seq(0,10,by=0.02)
> p=dexp(x,1.5)
> plot(x,p)

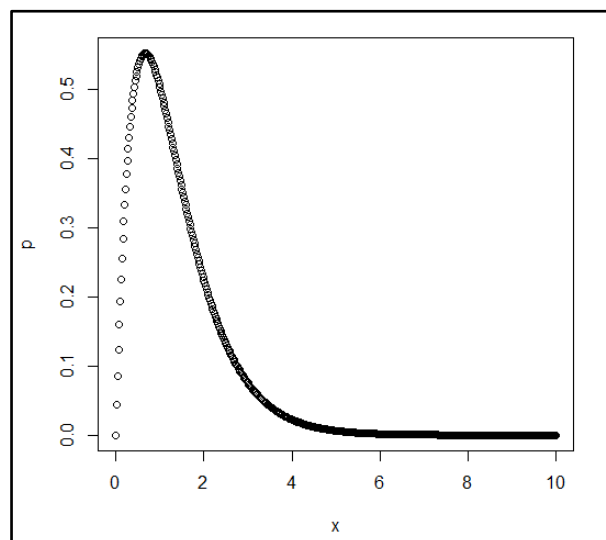
```



```

> a=pgamma(0.8,2,1.5)
> a
[1] 0.3373727
> b=pgamma(0.5,2,1.5,lower.tail=0)
> b
[1] 0.8266415
> x=seq(0,10,by=0.02)
> p=dgamma(x,2,1.5)
> plot(x,p)

```



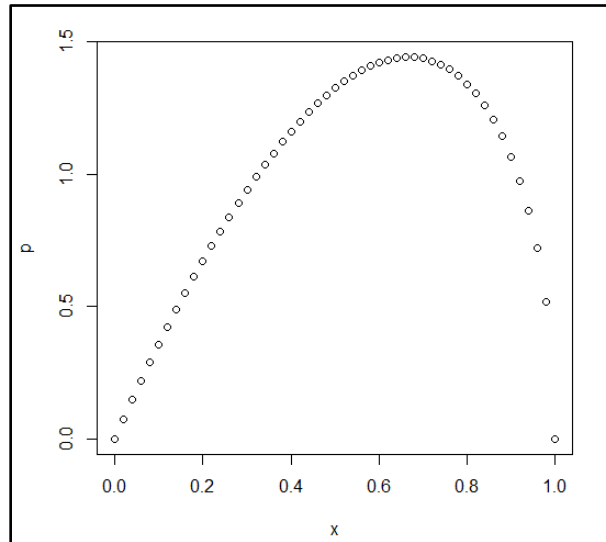
```

> a=pbeta(0.8,2,1.5)
> a
[1] 0.803226

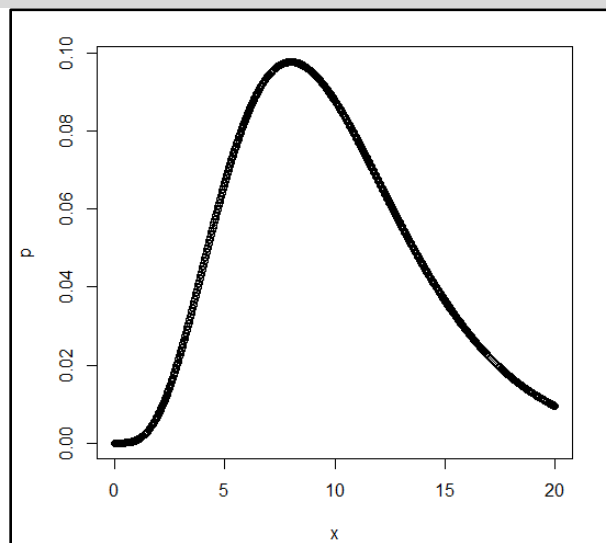
```



```
> b=pbeta(0.5,2,1.5,lower.tail=0)
> b
[1] 0.6187184
> x=seq(0,1,by=0.02)
> p=dbeta(x,2,1.5)
> plot(x,p)
```

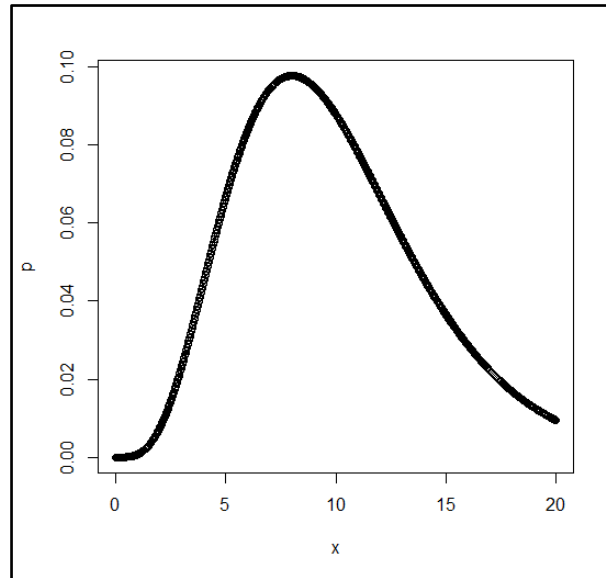


```
> a=pchisq(0.8,10)
> a
[1] 6.124333e-05
> b=pchisq(0.5,10,lower.tail=0)
> b
[1] 0.9999934
> x=seq(0,20,by=0.02)
> p=dchisq(x,10)
> plot(x,p)
```

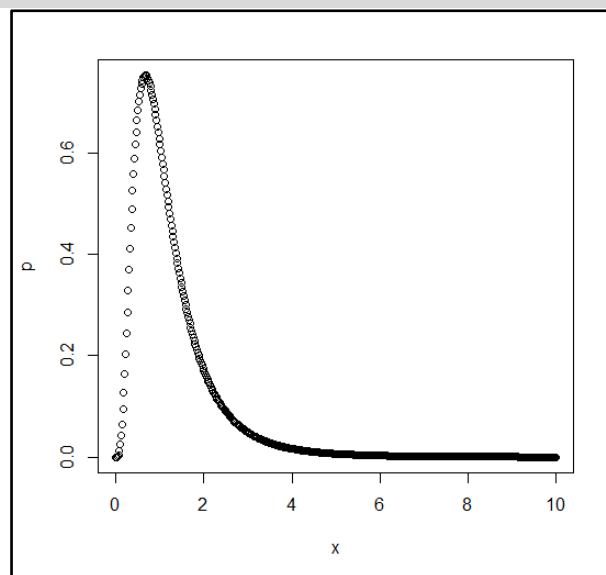


```
> a=pt(0.8,8)
> a
```

```
[1] 0.7765933
> b=pt(0.5,8,lower.tail=0)
> b
[1] 0.315268
> x=seq(-10,10,by=0.02)
> p=dt(x,8)
```

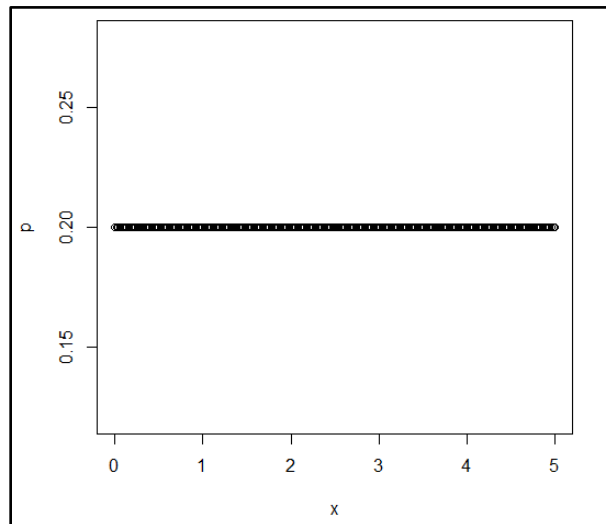


```
> a=pf(0.8,10,10)
> a
[1] 0.3655069
> b=pf(0.5,10,10,lower.tail=0)
> b
[1] 0.8551542
> x=seq(0,10,by=0.02)
> p=df(x,10,10)
> plot(x,p)
```



```
> a=punif(0.8,0,5)
> a
[1] 0.16
```

```
> b=punif(0.5,0,5,lower.tail=0)
> b
[1] 0.9
> x=seq(0,5,by=0.02)
> p=dunif(x,0,5)
> plot(x,p)
```



Chapter 8

Sampling Distribution and Central Limit Theorem using R

Dr. Rajendra Nana Chavhan, Assistant Professor Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

8.1 Introduction

In this chapter, I have demonstrated the sampling distribution of some well-known statistics as sample mean, sample variance and sample median. I used Poisson, Normal and Exponential distributions. I have also demonstrated the central limit theorem using sampling distributions.

8.2 Sampling Distribution

The sampling distribution of statistic is the distribution of statistic, considered as a random variable, when derived from random sample of size n . It may be considered as distribution of the statistic for all possible random samples from the same population of a given size. I have demonstrated sampling distribution of

1. Sample mean of discrete random variable with probability function
2. Sample mean of $X \sim \text{Exp}(1.2)$
3. Sample variance of $X \sim N(5, 9)$
4. Sample median where $X \sim \text{Poisson}(3.1)$

One can extend the study of sampling distributions with other sample statistic and distributions. This sampling distributions can be used for determining empirical probabilities.

Procedure for studying the sampling distribution

I used the simulation technique for studying the sampling distribution of different statistic using well known discrete as well as continuous probability distributions. I used sample size $n = 5, 10, 25$ and 50 , and 1000 repetitions. I used following steps

- Step 1. Drawing of random sample from considered population.
- Step 2. Calculation of sample statistic for different sample size ($n = 5, 15, 25$ and 50)

- Step 3. Comparison of population value with expected value of sample statistic for different sample size ($n = 5, 15, 25$ and 50) i.e. comparison of mean.
- Step 4. Comparison of variation of sample statistic for different sample size ($n = 5, 15, 25$ and 50) by studying variance.
- Step 5. Drawing of histogram for overall comparison.

8.2.1 Sampling distribution of sample mean of discrete random variable with probability function

Consider the following probability distribution

X	:	0	1	2	3
$P(X = x)$:	0.1	0.4	0.3	0.2

Here $E(X) = 1.6$ and $Var(X) = 0.84$, we study the sampling distribution of sample mean. We now that $E(\bar{X}) = 1.6$ and $Var(\bar{X}) = \frac{0.84}{n}$. I have written R-Program 1 for studying the sampling distribution of sample mean for above discrete probability distribution.

R-Program 1: R code for studying Sampling distribution of sample mean of discrete random variable

```
set.seed(1)      #for producing the same sequence of random variable every time
n=50;            #sample size
rep=1000;        #repetitions
xv=c(0,1,2,3)    #X values
prob=c(0.1,0.4,0.3,0.2) #Probability Values
#random sample from Discrete Distribution
x1=sample(xv,n*rep,replace = TRUE,prob=prob);
x=matrix(x1,rep,n)      #arrangement of random numbers in matrix
s.mean5=rowMeans(x[,1:5]) #sample mean n=5
s.mean10=rowMeans(x[,1:10]) #sample mean n=10
s.mean25=rowMeans(x[,1:25]) #sample mean n=25
s.mean50=rowMeans(x[,1:50]) #sample mean n=50
s.mean=data.frame(s.mean5,s.mean10,s.mean25,s.mean50) #bind all means
apply(s.mean,2,mean);apply(s.mean,2,var) #Calculation of mean and variance
par(mfrow=c(2,2));
hist(s.mean5,xlab = "(a)",main="n=5");
hist(s.mean10,xlab = "(b)",main="n=10");
hist(s.mean25,xlab = "(c)",main="n=25");
hist(s.mean50,xlab = "(d)",main="n=50")
```

We put the numerical output of R-Program 1, i.e. five point summary, mean and variance of sample mean of sizes $n = 5, 15, 25$ and 50 in the Table 1.

Table 1: Descriptive statistics of sample mean of discrete distribution

Sample size(n)	Minimum	Q1	Q2	Mean	Q3	Maximum	Variance
5	0.40	1.20	1.60	1.572	1.80	2.80	0.1719
10	0.80	1.40	1.60	1.587	1.80	2.50	0.0898

25	1.12	1.44	1.60	1.587	1.72	2.20	0.0358
50	1.14	1.52	1.60	1.600	1.68	2.04	0.0173

One can see that as sample increases mean of sample mean approaches to population mean and variances approaches to $\frac{0.84}{n}$. We can also see the shape of the sample means for considered sample sizes from Figure 1.

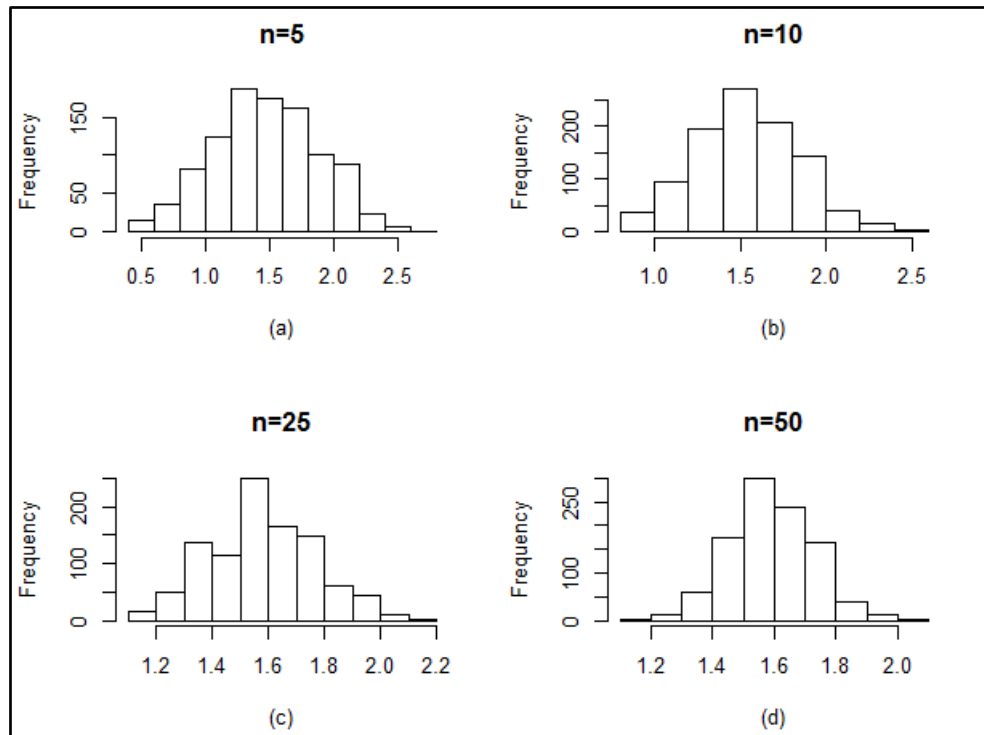


Figure 1: Sampling distribution of sample mean of discrete probability distribution for sample size (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$.

One can observe the overall shape, changing pattern of shape, variation, outliers, Skewness, outliers etc. of sample mean from Figure 1. We can conclude that mean of sample mean is concentrating towards the population mean $E(X) = 1.6$ whereas variation decreases.

8.2.2 Sampling distribution of sample mean where $X \sim N(10, 4)$

Here I studied the sampling distribution of sample mean where parent population is normal with mean 10 and variance 4. I have written the following R-Program 2 for studying Sampling distribution of sample mean where $X \sim N(10, 4)$.

R-Program 2: R code for studying Sampling distribution of sample mean of $X \sim N(10, 4)$

```
set.seed(25)      #for producing the same sequence of random variable everytime
n=50;              #sample size
rep=1000;          #repetitions
x1=rnorm(rep*n,10,2); #random sample from Population N(10,4)
x=matrix(x1,rep,n) #arrangement of random numbers in matrix
```

```

s.mean5=rowMeans(x[,1:5])      #sample mean n=5
s.mean10=rowMeans(x[,1:10])    #sample mean n=10
s.mean25=rowMeans(x[,1:25])    #sample mean n=25
s.mean50=rowMeans(x[,1:50])    #sample mean n=50
s.mean=data.frame(s.mean5,s.mean10,s.mean25,s.mean50) #bind all means
summary(s.mean) #gives six point summary(min,Q1,Q2,mean,Q3 and max)
apply(s.mean,2,var) #Calculation of Variance
par(mfrow=c(2,2));
hist(s.mean5,xlab = "(a)",main="n=5");
hist(s.mean10,xlab = "(b)",main="n=10");
hist(s.mean25,xlab = "(c)",main="n=25");
hist(s.mean50,xlab = "(d)",main="n=50")

```

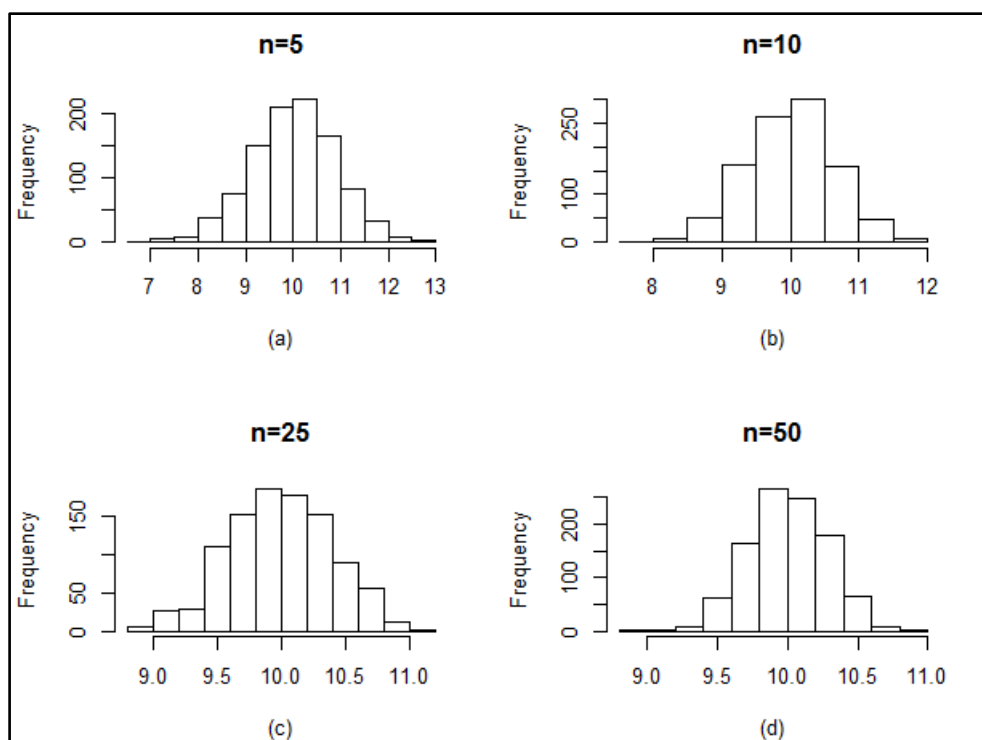


Figure 2: Sampling distribution of sample mean of $X \sim N(10, 4)$ for sample of sizes (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$.

Figure 2 shows the histogram for sample mean of sizes (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$ where parent population is $N(10, 4)$. One can observe the frequency distribution, overall shape of sample mean of normal distribution having mean 10 and variance 4. As sample size increases, sample mean gets closer to population mean with decrement in variances and spread. This can be confirmed from descriptive statistics given in Table 2. Numerical output of R-Program 2, i.e. five point summary, mean and variance of sample mean of sizes $n = 5, 15, 25$ and 50 is given in the Table 2.

Table 2: Descriptive statistics of sample mean of $N(10, 4)$

Sample size(n)	Minimum	Q1	Q2	Mean	Q3	Maximum	Variance
5	6.630	9.388	10.030	10.005	10.598	12.763	0.817
10	7.625	9.549	10.016	9.994	10.432	11.813	0.407

25	8.827	9.716	9.993	9.989	10.278	11.050	0.156
50	8.961	9.812	10.000	10.002	10.204	10.971	0.076

8.2.3 Sampling distribution of sample variance where $X \sim \text{Exp}(1.2)$

Here I studied the sampling distribution of sample variance where sample is drawn from exponential distribution with parameter 1.2. I have written the following R-Program 3 for studying sampling distribution of sample variances where $X \sim \text{Exp}(1.2)$

R-Program 3: R code for studying Sampling distribution of sample variance of $X \sim \text{Exp}(1.2)$

```
set.seed(25) #for producing the same sequence of random variable every time
n=50; #sample size
rep=1000; #repetitions
x1=rexp(rep*n,1.2);#random sample from Population Exponential with mean=1/1.2
x=matrix(x1,rep,n); #arrangement of random numbers in matrix
s.var5=apply(x[,1:5],1,var); #sample variance n=5
s.var10=apply(x[,1:10],1,var); #sample variance n=10
s.var25=apply(x[,1:25],1,var); #sample variance n=25
s.var50=apply(x[,1:50],1,var); #sample variance n=50
s.var=data.frame(s.var5,s.var10,s.var25,s.var50) #bind all variances
summary(s.var) #gives six point summary(min,Q1,Q2,mean,Q3 and max)
apply(s.var,2,var) #Calculation of Variance
par(mfrow=c(2,2));
hist(s.var5,xlab = "(a)",main="n=5");
hist(s.var10,xlab = "(b)",main="n=10");
hist(s.var25,xlab = "(c)",main="n=25");
hist(s.var50,xlab = "(d)",main="n=50")
```

Numerical output of R-Program 3, i.e. five point summary, mean and variance of sample variance of $\text{Exp}(1.2)$ of sizes $n = 5, 15, 25$ and 50 is given in the Table 3.

Table 3: Descriptive statistics of sample variance of $\text{Exp}(1.2)$

Sample size(n)	Minimum	Q1	Q2	Mean	Q3	Maximum	Variance
5	0.004	0.182	0.398	0.692	0.829	14.281	0.914
10	0.029	0.301	0.500	0.678	0.852	7.894	0.405
25	0.128	0.416	0.613	0.697	0.844	3.696	0.171
50	0.194	0.494	0.649	0.695	0.831	2.267	0.080

Figure 3 shows the histogram of sample variance of sizes (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$ where parent population is exponential with parameter 1.2. From Figure 3, one can see that distribution of sample variance is positively skewed.

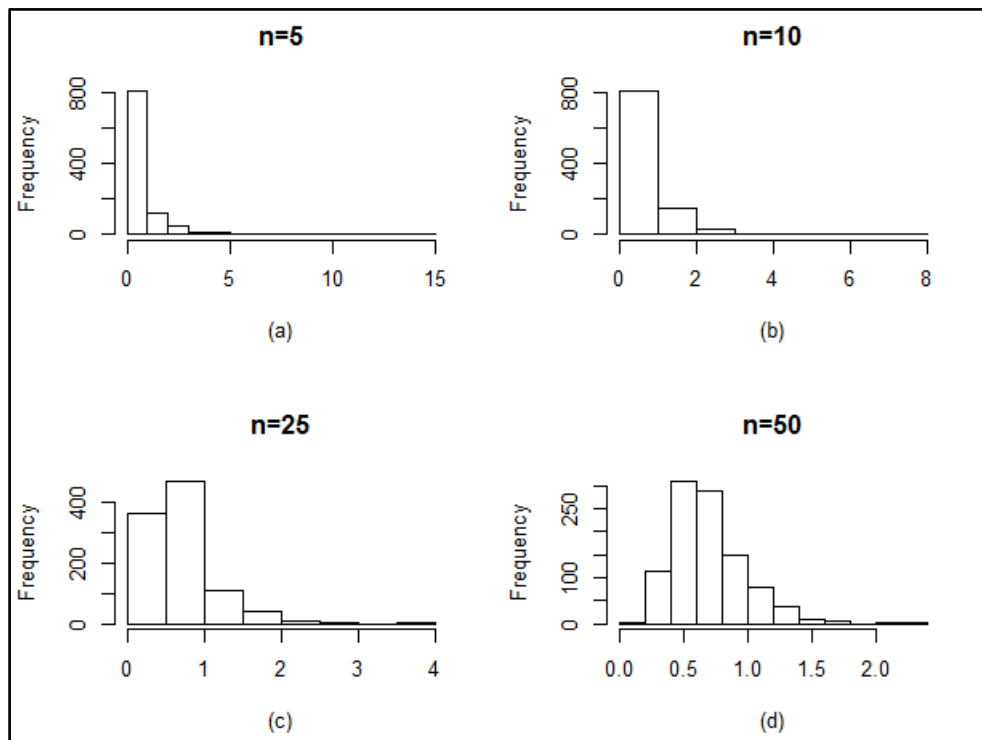


Figure 3: Sampling distribution of sample variance of $X \sim \text{Exp}(1.2)$ for sample of sizes (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$.

8.2.4 Sampling distribution of sample median where $X \sim \text{Pois}(3.1)$:

Here I studied the sampling distribution of sample median where sample is drawn from Poisson distribution with mean 3.1. I have written the following R-Program 4 for studying sampling distribution of sample median where $X \sim \text{Pois}(3.1)$.

R-Program 4: R code for studying sampling distribution of sample median of $X \sim \text{Pois}(3.1)$

```
set.seed(25) #for producing the same sequence of random variable every time
n=50; #sample size
rep=1000; #repetitions
x1=rpois(rep*n,3.1); #random sample from Population Poisson with mean=3.1
x=matrix(x1,rep,n); #arrangement of random numbers in matrix
s.med5=apply(x[,1:5],1,median); #sample median n=5
s.med10=apply(x[,1:10],1,median); #sample median n=10
s.med25=apply(x[,1:25],1,median); #sample median n=25
s.med50=apply(x[,1:50],1,median); #sample median n=50
s.med=data.frame(s.med5,s.med10,s.med25,s.med50) #bind all Medians
summary(s.med) #gives six point summary(min,Q1,Q2,mean,Q3 and max)
apply(s.med,2,var) #Calculation of Variance
par(mfrow=c(2,2));
hist(s.med5,xlab = "(a)",main="n=5");
hist(s.med10,xlab = "(b)",main="n=10");
hist(s.med25,xlab = "(c)",main="n=25");
hist(s.med50,xlab = "(d)",main="n=50")
```

Table 3 contains the descriptive statistics of sample median for different sample sizes obtained from numerical output of R-Program 4.

Table 3: Descriptive statistics of sample median of $Poiss(3.1)$

Sample size(n)	Minimum	Q1	Q2	Mean	Q3	Maximum	Variance
5	1	2	3	2.976	4	6	0.9624
10	1	2.5	3	2.944	3.5	5.5	0.4783
25	2	3	3	2.924	3	5	0.2605
50	2	3	3	2.943	3	4	0.1082

Figure 4 shows histogram of sample median of Poisson with mean 3.1 which shows frequency distribution of sample median.

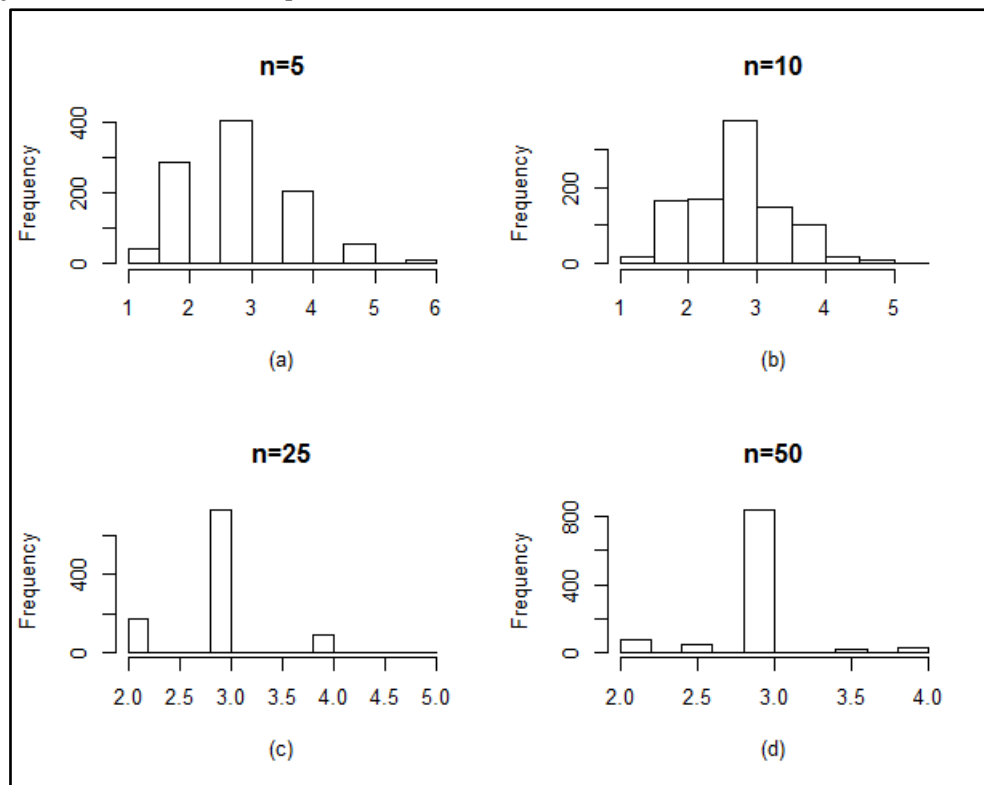


Figure 4: Sampling distribution of sample median of $X \sim Poiss(3.1)$ for sample of sizes (a) $n = 5$ (b) $n = 10$ (c) $n = 25$ and (d) $n = 50$.

8.3 Central Limit Theorem (CLT)

If X_1, X_2, \dots, X_n is a random sample of size n (large) from any probability distribution (either discrete or continuous) with finite mean μ and variance σ^2 then sample mean $\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$ will tends to normal distribution with mean μ and variance $\frac{\sigma^2}{n}$. Here I demonstrated the CLT for the following probability distributions

1. Negative Binomial Distribution
2. Continuous Uniform Distribution

I used $n = 10, 50, 100$ and 250 for demonstration. Shapiro test is used to test normality. I have also plot histogram along with normal curve to asses the normality.

8.3.1 Negative Binomial Distribution

Consider X_1, X_2, \dots, X_n is random sample from negative binomial with $k = 5$ and $p = 0.7$. Here X represents the number of failure before k successes. I have written the following R-Program 5 for studying sampling distribution of sample mean and to demonstrate the CLT where $X \sim NB(5, 0.7)$.

R-Program 5: R code for demonstration of CLT of $X \sim NB(5, 0.7)$

```
set.seed(5)      #for producing the same sequence of random variable every time
n=250;           #sample size
rep=1000;        #repetitions
x1=rnbinom(rep*n,5,0.7); #random sample from Negative Binomial k=5, p=0.7
x=matrix(x1,rep,n); #arrangement of random numbers in matrix
s.mean10=apply(x[,1:10],1,mean); #sample mean n=10
s.mean50=apply(x[,1:50],1,mean); #sample mean n=50
s.mean100=apply(x[,1:100],1,mean); #sample mean n=100
s.mean250=apply(x[,1:250],1,mean); #sample mean n=250
nt10=shapiro.test(s.mean10); #Normality test of sample mean n=10
nt50=shapiro.test(s.mean50); #Normality test of sample mean n=50
nt100=shapiro.test(s.mean100); #Normality test of sample mean n=100
nt250=shapiro.test(s.mean250); #Normality test of sample mean n=250
#P-value of the normality test
print(c(nt10$p.value,nt50$p.value,nt100$p.value,nt250$p.value))
#Function from plotting Histogram with Normal curve
hist_curve<-function(x){
  N=length(x);H=hist(x,breaks=50,xlab="",main="");dx=(H$breaks[2]-
H$breaks[1]);
  x0=H$breaks;x1=c(x0[1]-dx/2,x0+dx/2);
  lines(x1,N*dnorm(x1,mean(x),sd(x))*dx,col="blue")
}
par(mfrow=c(2,2));
hist_curve(s.mean10);title(main="n=10",xlab="(a)");
hist_curve(s.mean50);title(main="n=50",xlab="(b)");
hist_curve(s.mean100);title(main="n=100",xlab="(c)");
hist_curve(s.mean250);title(main="n=250",xlab="(d)");
```

Table 5 shows the P-value of Shapiro test of normality.

Table 5: P-value for Shapiro test of normality

Sample size(n)	10	50	100	250
P-value	0.0000	0.1241	0.3139	0.7999

CLT hold for $n = 50, 100, 250$ which can be confirmed from P-value given in Table 5. In Figure 5, I used to draw histogram with normal curve. One can see the normal curve fits well for (b) $n=50$, (c) $n=100$ and (d) $n=250$. As sample size increases normal curve fits well.

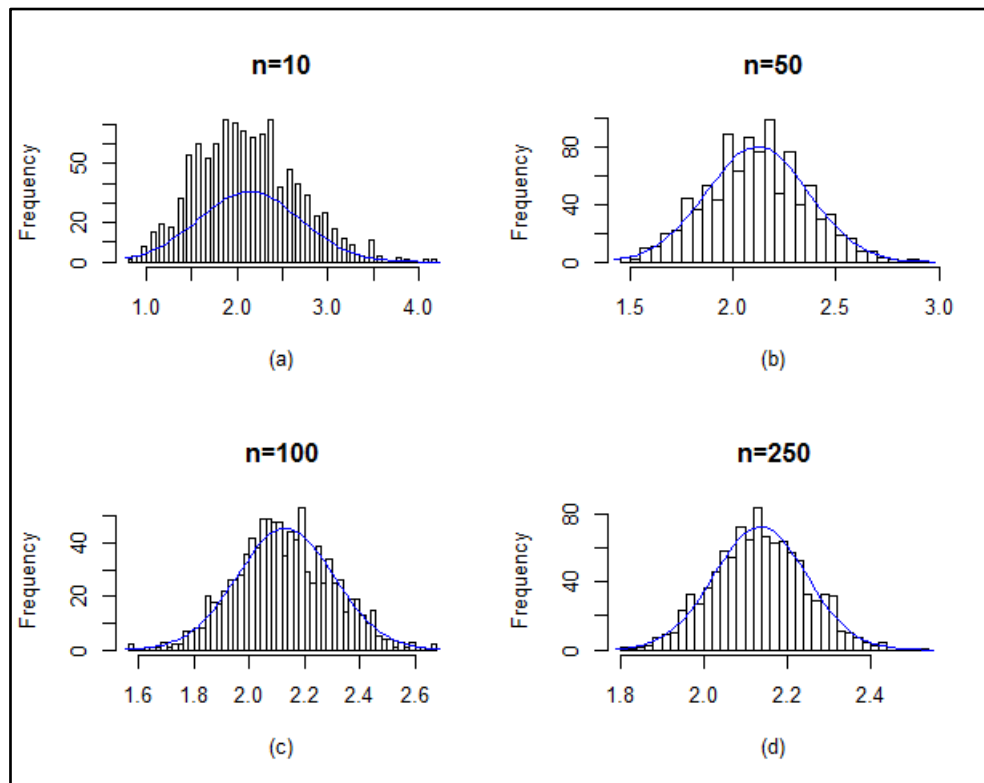


Figure 5: Sampling distribution of sample mean with normal curve of $X \sim NB(5, 0.7)$ for sample of sizes (a) $n = 10$ (b) $n = 50$ (c) $n = 100$ and (d) $n = 250$.

8.3.2 Continuous uniform distribution

Consider X_1, X_2, \dots, X_n is random sample from continuous uniform distribution in the interval $(0, 10)$. I have written the following R-Program 6 for studying sampling distribution of sample mean and to demonstrate the CLT where $X \sim U(0, 10)$.

R-Program 6: R code for demonstration of CLT of $X \sim U(0, 10)$

```
set.seed(50) #for producing the same sequence of random variable every time
n=250;      #sample size
rep=1000;   #repetition
x1=runif(rep*n,0,10); #random sample from Negative Binomial k=5, p=0.7
x=matrix(x1,rep,n);   #arrangement of random numbers in matrix
s.mean10=apply(x[,1:10],1,mean); #sample mean n=10
s.mean50=apply(x[,1:50],1,mean); #sample mean n=50
s.mean100=apply(x[,1:100],1,mean); #sample mean n=100
s.mean250=apply(x[,1:250],1,mean); #sample mean n=250
nt10=shapiro.test(s.mean10); #Normality test of sample mean n=10
nt50=shapiro.test(s.mean50); #Normality test of sample mean n=50
nt100=shapiro.test(s.mean100); #Normality test of sample mean n=100
nt250=shapiro.test(s.mean250); #Normality test of sample mean n=250
```

```

p.value=c(nt10$p.value,nt50$p.value,nt100$p.value,nt250$p.value) #P-value of
the normality test
#Function from plotting Histogram with Normal curve
hist_curve<-function(x){
  N=length(x);H=hist(x,breaks=50,xlab="",main="");dx=(H$breaks[2]-
H$breaks[1]);
  x0=H$breaks;x1=c(x0[1]-dx/2,x0+dx/2);
  lines(x1,N*dnorm(x1,mean(x),sd(x))*dx,col="blue")
}
par(mfrow=c(2,2));
hist_curve(s.mean10);title(main="n=10",xlab="(a)");
hist_curve(s.mean50);title(main="n=50",xlab="(b)");
hist_curve(s.mean100);title(main="n=100",xlab="(c)");
hist_curve(s.mean250);title(main="n=250",xlab="(d)")

```

Table 6 shows the P-value of Shapiro test of normality.

Table 6: P-value for Shapiro test of normality

Sample size(n)	10	50	100	250
P-value	0.0348	0.2414	0.2984	0.3321

CLT hold for $n = 50, 100, 250$ which can be confirmed from P-value given in Table 6. In Figure 6, I used to draw histogram with normal curve. One can see the normal curve fits well for (b) $n=50$, (c) $n=100$ and (d) $n=250$. As sample size increases normal curve fits well.

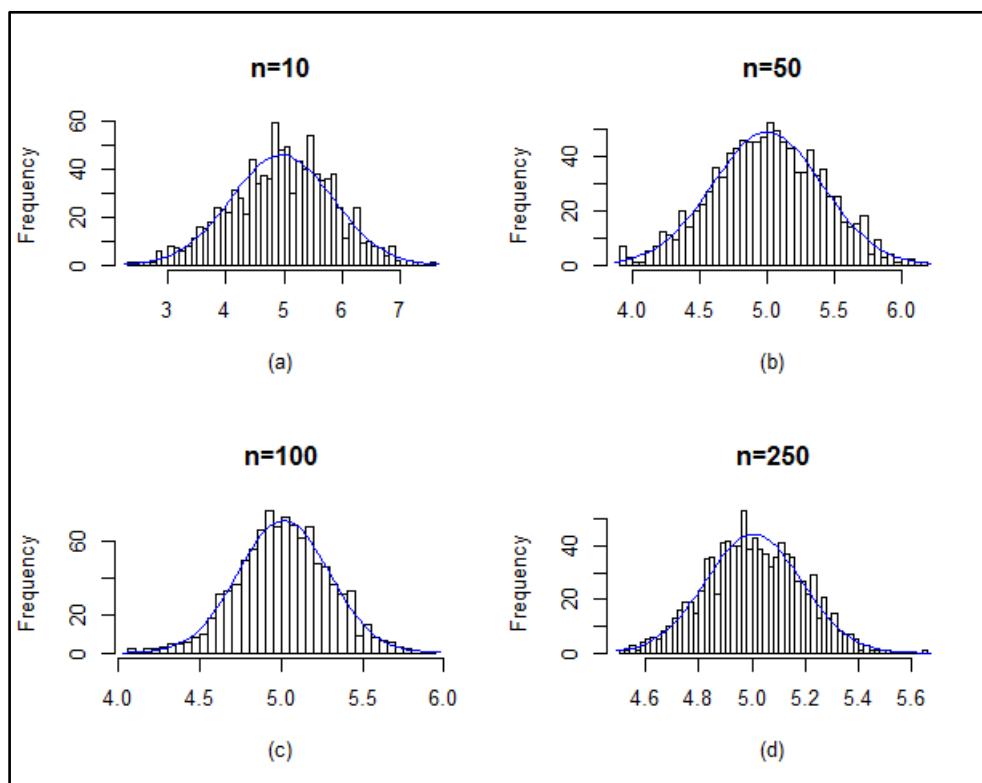


Figure 6: Sampling distribution of sample mean with normal curve of $X \sim U(0, 10)$ for sample of sizes (a) $n = 10$ (b) $n = 50$ (c) $n = 100$ and (d) $n = 250$.

8.4 Some important notes

- One can extend the study of sampling distributions with other sample statistic and distributions.
- This sampling distributions can be used for determining empirical probabilities.
- One can verify the other results like CLT.
- Sampling distributions of complicated statistic can be studied.

8.5 References

- Verzani, J. (2014). *Using R for introductory statistics*. CRC Press.
-

Chapter 9

Statistical Tests Using R

Dr. Rajendra Nana Chavhan, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

9.1 Introduction

In this chapter, I have demonstrated the one sample t-test, two sample t-test, paired t-test, chi-square test for variance, F-test for equality of two variances with example in R programming. This article is useful for students, teachers and researchers in applied sciences.

9.2 t-test

9.2.1 One sample t-test

One sample t-test is used to investigate whether population mean (μ) is regarded as some specified value μ_0 , based on a random sample. That is, to test the significance of the difference between the sample mean (\bar{X}) and the assumed population mean μ_0 . We assume population from which, the sample of size n drawn is Normal distribution whose population mean is unknown. We test one of the following null hypothesis (H_0) and alternative hypothesis (H_1) at α level of significance.

- a) H_0 : There is no significant difference between the sample mean \bar{X} and the assumed population mean μ . i.e., $H_0 : \mu = \mu_0$ vs $H_1 : \mu \neq \mu_0$
- b) $H_0 : \mu \leq \mu_0$ vs $H_1 : \mu > \mu_0$
- c) $H_0 : \mu \geq \mu_0$ vs $H_1 : \mu < \mu_0$

The test statistic for testing the above hypothesis is

$$t = \frac{\bar{X} - \mu}{\hat{\sigma}/\sqrt{n}}$$

Where $\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$ and $\hat{\sigma}^2 = S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}$

Under H_0 , the test statistic follows t distribution with $(n - 1)$ degrees of freedom. We take the decision whether to reject the null hypothesis or not based on P-value. If P-value $< \alpha$ then we reject the null hypothesis and if P-value $\geq \alpha$ then we do not have enough evidence to reject the null hypothesis. The P-value is calculated as

- For
- a) $H_1 : \mu \neq \mu_0$, P-value = $2 \times P(T > |t|)$
 - b) $H_1 : \mu > \mu_0$, P-value = $P(T > t)$
 - c) $H_1 : \mu < \mu_0$, P-value = $P(T < t)$

where T follows t distribution with $(n - 1)$ degrees of freedom.

9.2.2 Two sample t-test

Two sample t-test is used to investigate the null hypothesis of the difference between mean of the two populations is some constant value, based on two random samples. We assume that the populations from which, the two samples drawn, are Normal distributions which have unknown and same variance. A random sample of size m observations X_1, X_2, \dots, X_m be drawn from population with unknown mean μ_1 and a random sample of size n observations Y_1, Y_2, \dots, Y_n be drawn from population with unknown mean μ_2 . We assume that both the populations have equal variances. We test one of the following null hypothesis (H_0) and alternative hypothesis (H_1) at α level of significance.

- a) H_0 : The difference between two population mean is some constant value c . i.e. H_0 :
 $\mu_1 - \mu_2 = c$ vs $H_1 : \mu_1 - \mu_2 \neq c$
- b) H_0 : $\mu_1 - \mu_2 \leq c$ vs $H_1 : \mu_1 - \mu_2 > c$
- c) H_0 : $\mu_1 - \mu_2 \geq c$ vs $H_1 : \mu_1 - \mu_2 < c$

The test statistic for testing the above hypothesis is

$$t = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{S \times \sqrt{\left(\frac{1}{m} + \frac{1}{n}\right)}}$$

where $\bar{X} = \frac{\sum_{i=1}^m X_i}{m}$, $\bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}$ and $S^2 = \frac{\sum_{i=1}^m (X_i - \bar{X})^2 + \sum_{i=1}^n (Y_i - \bar{Y})^2}{m+n-2}$

Under H_0 , the test statistic follows t distribution with $(m + n - 2)$ degrees of freedom. The P-value is calculated as

- For
- a) $H_1 : \mu_1 - \mu_2 \neq c$, P-value = $2 \times P(T > |t|)$
 - b) $H_1 : \mu_1 - \mu_2 > c$, P-value = $P(T > t)$
 - c) $H_1 : \mu_1 - \mu_2 < c$, P-value = $P(T < t)$

where T follows t distribution with $(m + n - 2)$ degrees of freedom.

If the assumption of equality of variance of two samples does not hold then the test statistics for testing the null hypothesis is

$$t = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{\sqrt{\left(\frac{S_1^2}{m} + \frac{S_2^2}{n}\right)}}$$

$$\text{where } \bar{X} = \frac{\sum_{i=1}^m X_i}{m}, \bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}, S_1^2 = \frac{\sum_{i=1}^m (X_i - \bar{X})^2}{m-1} \text{ and } S_2^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n-1}$$

Under $H_0 : \mu_1 - \mu_2 = c$, the test statistic follows t distribution with ν degrees of freedom

$$\text{where } \nu = \frac{\left(\frac{S_1^2}{m} + \frac{S_2^2}{n}\right)}{\frac{S_1^4}{m^2(m-1)} + \frac{S_2^4}{n^2(n-1)}}. \text{ This t-test commonly known as } \mathbf{Welch \ Two \ Sample \ t-test}.$$

Method of calculation of P-value is same as per two sample t-test.

9.2.3 Paired t-test

Paired t-test is used to investigate the significance of the difference between before and after the treatment in the sample. Let X_1, X_2, \dots, X_n be the observations made initially from n individuals as a random sample of size n . A treatment is applied to the above individuals and observations are made after the treatment and are denoted by Y_1, Y_2, \dots, Y_n . That is, (X_i, Y_i) denotes the pair of observations obtained from the i^{th} individual, before and after the treatment applied. Let μ_X is unknown population mean before the treatment and μ_Y is the unknown population mean after the treatment. We assume that the populations from which, the two samples drawn, are Normal distribution and observations are collected in a pair. We test one of the following null hypothesis (H_0) and alternative hypothesis (H_1) at α level of significance.

- a) $H_0 : \text{There is no significant difference between before and after the treatment applied. i.e. treatment applied, is ineffective. i.e., } H_0 : \mu_d = \mu_X - \mu_Y = c \text{ vs } H_1 : \mu_d \neq c$
- b) $H_0 : \mu_d \leq 0 \text{ vs } H_1 : \mu_d > c$
- c) $H_0 : \mu_d \geq 0 \text{ vs } H_1 : \mu_d < c$

The test statistic for testing the above hypothesis is

$$t = \frac{\bar{d} - \mu_d}{S_d / \sqrt{n}}$$

$$\text{where } \bar{d} = \frac{\sum_{i=1}^n d_i}{n}, d_i = X_i - Y_i \text{ and } S_d^2 = \frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n-1}$$

Under H_0 , the test statistic follows t distribution with $(n - 1)$ degrees of freedom. The P-value is calculated as

For a) $H_1 : \mu_d \neq c$,	P-value = $2 \times P(T > t)$
b) $H_1 : \mu_d > c$,	P-value = $P(T > t)$
c) $H_1 : \mu_d < c$	P-value = $P(T < t)$

where T follows t distribution with $(n - 1)$ degrees of freedom.

In R programming, the **t.test()** function produces the variety of t-tests. We will discuss the different t-tests by following Example 1, 2 and 3.

Example 1 (One Sample t-test): A sample of 13 students from a government school has the following scores in a test.

89 88 78 76 78 78 86 83 82 76 72 77 92.

Do this data support that i) the mean mark of the school students is 80? Test at 5% level.

ii) the mean mark of the school students is more than 75? Test at 1% level.

iii) the mean mark of the school students is less than 85? Test at 10% level.

Solution:

i) Here we test, $H_0 : \mu = 80$ against $H_1 : \mu \neq 80$.

```
x=c(89,88,78,76,78,78,86,83,82,76,72,77,92) #data
t.test(x,mu=80) #by default alternative is two sided and level is 5%
```

Output

```
One Sample t-test
data: x
t = 0.68885, df = 12, p-value = 0.504
alternative hypothesis: true mean is not equal to 80
95 percent confidence interval:
77.50427 84.80342
sample estimates:
mean of x
81.15385
```

R Output gives the test statistic t , degrees of freedom and P-value.

Here P-value is $0.504 > 0.05$, hence we do not have enough evidence to reject H_0 (i.e. Accept H_0). Output also gives additional information about the confidence interval with sample estimate of μ . Here 95% confidence interval is (77.50427, 84.80342) which also support the decision taken from P-value as 80 is included in the confidence interval.

ii) Here we test, $H_0 : \mu \leq 75$ against $H_1 : \mu > 75$.

```
x=c(89,88,78,76,78,78,86,83,82,76,72,77,92) #data
t.test(x,mu=75,alternative = "greater",cof.level=0.99)
```

Output

```
One Sample t-test

data: x
t = 3.6739, df = 12, p-value = 0.001592
alternative hypothesis: true mean is greater than 75
95 percent confidence interval:
 78.16846      Inf
sample estimates:
mean of x
81.15385
```

Here P-value is $0.001592 < 0.01$, hence we reject H_0 (i.e. Accept H_1). Output also gives one sided confidence interval with sample estimate of μ which support the decision taken from P-value.

iii) Here we test, $H_0 : \mu \geq 85$ against $H_1 : \mu < 85$.

```
x=c(89,88,78,76,78,78,86,83,82,76,72,77,92)
t.test(x,mu=85,alternative = "less",cof.level=0.9)
```

Output:

```
One Sample t-test

data:  x
t = -2.2962, df = 12, p-value = 0.02024
alternative hypothesis: true mean is less than 85
95 percent confidence interval:
 -Inf 84.13923
sample estimates:
mean of x
 81.15385
```

Here P-value is $0.02024 < 0.1$, hence we reject H_0 (i.e. Accept H_1). Output also gives one sided confidence interval with sample estimate of μ which support the decision taken from P-value.

Example 2 (Two Sample t-test): The yield of two varieties of mango (in tons) on two independent sample of 10 and 12 plants are given below.

Variety-A: 22 24 26 23 26 30 32 34

Variety-B: 28 25 26 30 32 30 33 28 30 35

- i) Test whether the yield of Variety-A is not equal to Variety-B at 2% level of significance.
- ii) Test whether the difference between yield of Variety-A is less than Variety-B by 2 tones at 5% level of significance.
- iii) Test whether the difference between yield of Variety-A is more than Variety-B by 0.5 tones at 10% level of significance.
- iv) Test whether the yield of Variety-A is not equal to Variety-B at 5% level of significance assume unequal variances of both samples.

Solution:

- i) Here we test, $H_0: \mu_1 - \mu_2 = 0$ against $H_1: \mu_1 - \mu_2 \neq 0$

```
x=c(22,24,26,23,26,30,32,34)          #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)    #second sample data
t.test(x,y,var.equal = TRUE, conf.level = 0.98)
#by default c=0 and alternative
#hypothesis is two sided
```

Output:

```
Two Sample t-test
data:  x and y
t = -1.4607, df = 16, p-value = 0.1634
alternative hypothesis: true difference in means is not equal to 0
98 percent confidence interval:
 -7.129169  1.979169
sample estimates:
mean of x mean of y
 27.125    29.700
```

Here P-value is $0.1634 > 0.02$, hence we do not have enough evidence to reject H_0 (i.e. Accept H_0). Output also give confidence interval of difference of means with sample estimates of μ_1 and μ_2 which support the decision taken from P-value.

ii) Here we test, $H_0: \mu_1 - \mu_2 \geq 2$ against $H_1: \mu_1 - \mu_2 < 2$

```
x=c(22,24,26,23,26,30,32,34)          #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)    #second sample data
t.test(x,y,var.equal = TRUE, mu=2,alternative = "less", conf.level = 0.95)
```

Output:

```
Two Sample t-test

data: x and y
t = -2.5953, df = 16, p-value = 0.009763
alternative hypothesis: true difference in means is less than 2
95 percent confidence interval:
 -Inf 0.5026423
sample estimates:
mean of x mean of y
 27.125    29.700
```

Here P-value is $0.009763 < 0.05$, hence we reject H_0 (i.e. Accept H_1). Output also gives one sided confidence interval of difference of means with sample estimates of μ_1 and μ_2 which support the decision taken from P-value.

iii) Here we test, $H_0: \mu_1 - \mu_2 \leq 0.5$ against $H_1: \mu_1 - \mu_2 > 0.5$

```
x=c(22,24,26,23,26,30,32,34)          #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)    #second sample data
t.test(x,y,var.equal = TRUE, mu=0.5,alternative = "greater", conf.level = 0.9)
```

Output:

```
Two Sample t-test

data: x and y
t = -1.7444, df = 16, p-value = 0.9499
alternative hypothesis: true difference in means is greater than 0.5
90 percent confidence interval:
 -4.931434      Inf
sample estimates:
mean of x mean of y
 27.125    29.700
```

Here P-value is $0.9499 > 0.1$, hence we do not have enough evidence to reject H_0 (i.e. Accept H_0). Output also give confidence interval of difference of means with sample estimates of μ_1 and μ_2 which support the decision taken from P-value.

iv) Here we test, $H_0: \mu_1 - \mu_2 = 0$ against $H_1: \mu_1 - \mu_2 \neq 0$ where assumption of equality of variance of two sample does not hold.

```
x=c(22,24,26,23,26,30,32,34)          #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)    #second sample data
t.test(x,y) #by default c=0, alternative hypothesis is two sided and los=5%
             #by default variances are not equal
```

Output:

```
Welch Two Sample t-test

data:  x and y
t = -1.4037, df = 12.172, p-value = 0.1854
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.565645  1.415645
sample estimates:
mean of x mean of y
  27.125   29.700
```

Here P-value is $0.1854 > 0.05$, hence we do not have evidence to reject H_0 (i.e. Accept H_0). Output also give confidence interval of difference of means with sample estimates of μ_1 and μ_2 which support the decision taken from P-value.

Example 3 (Paired t-test): A new variety of health drink in the market for weight of infants. A sample of 10 babies was selected and was given the above diet for a month and the weights were observed before (X) and after (Y) the diet given.

```
X: 6.6 6.85 6.75 7.2 6.75 6.65 6.7 7.3 6.9 6.6
Y: 6.9 7.3 7 7.6 6.85 7.3 6.7 7.45 7.3 6.5
```

- Examine whether there is significant difference between before and after the healthy drink diet at 5% level of significance.
- Examine whether the weight gain after the healthy drink diet is more than 0.2 kg at 1% level of significance.
- Examine whether the weight loss after the healthy drink diet is less than 0.5 kg at 10% level of significance.

Solution:

i) Here we test, $H_0: \mu_d = \mu_X - \mu_Y = 0$ against $H_1: \mu_d \neq 0$

```
x=c(6.6,6.85,6.75,7.2,6.75,6.65,6.7,7.3,6.9,6.6) #Before Treatment Data
y=c(6.9,7.3,7,7.6,6.85,7.3,6.7,7.45,7.3,6.5)    #After Treatment Data
t.test(x,y,paired = TRUE) #by default c=0, alternative is two sided and los=5%
```

Output:

```
Paired t-test
data:  x and y
t = -3.6211, df = 9, p-value = 0.005563
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.42242786 -0.09757214
sample estimates:
mean of the differences
      -0.26
```

Here P-value is $0.005563 < 0.05$, hence we reject H_0 (i.e. Accept H_1). Output also gives confidence interval and sample estimate of μ_d which also support the decision taken from P-value.

ii) Here we test, $H_0: \mu_d = \mu_X - \mu_Y \leq 0.2$ against $H_1: \mu_d > 0.2$

```
x=c(6.6,6.85,6.75,7.2,6.75,6.65,6.7,7.3,6.9,6.6) #Before Treatment Data
y=c(6.9,7.3,7,7.6,6.85,7.3,6.7,7.45,7.3,6.5)      #After Treatment Data
t.test(x,y,paired = TRUE,mu=0.2,conf.level = 0.99,alternative = "greater")
```

Output:

```
Paired t-test

data:  x and y
t = -6.4065, df = 9, p-value = 0.9999
alternative hypothesis: true difference in means is greater than 0.2
99 percent confidence interval:
 -0.4625854      Inf
sample estimates:
mean of the differences
      -0.26
```

Here P-value is $0.9999 > 0.01$, hence we do not have evidence to reject H_0 (i.e. Accept H_0). Output also gives confidence interval and sample estimate of μ_d which also support the decision taken from P-value.

iii) Here we test, $H_0: \mu_d = \mu_X - \mu_Y \geq 0.5$ against $H_1: \mu_d < 0.5$

```
x=c(6.6,6.85,6.75,7.2,6.75,6.65,6.7,7.3,6.9,6.6) #Before Treatment Data
y=c(6.9,7.3,7,7.6,6.85,7.3,6.7,7.45,7.3,6.5)      #After Treatment Data
t.test(x,y,paired = TRUE,mu=0.5,conf.level = 0.9,alternative = "less")
```

Output:

```
Paired t-test

data:  x and y
t = -10.585, df = 9, p-value = 1.113e-06
alternative hypothesis: true difference in means is less than 0.5
90 percent confidence interval:
 -Inf -0.1606955
sample estimates:
mean of the differences
      -0.26
```

Here P-value is < 0.1 , hence we reject H_0 (i.e. Accept H_1). Output also gives confidence interval and sample estimate of μ_d which also support the decision taken from P-value.

9.3 Chi-square Test for Variance:

Chi-square test for variance is used to test the population variance σ^2 regarded as σ_0^2 based on a random sample of size n which is drawn from normal population with mean μ and variance σ_0^2 (both μ and σ^2 are unknown). We investigate the significance of the difference between the assumed population variance σ_0^2 and the sample variance. We test one of the following null hypothesis (H_0) and alternative hypothesis (H_1) at α level of significance.

- a) H_0 : There is no significant difference between the sample variance S^2 and the assumed population variance σ_0^2 . i.e., $H_0 : \sigma^2 = \sigma_0^2$ vs $H_1 : \sigma^2 \neq \sigma_0^2$

$$b) H_0 : \sigma^2 \leq \sigma_0^2 \text{ vs } H_1 : \sigma^2 > \sigma_0^2$$

$$c) H_0 : \sigma^2 \geq \sigma_0^2 \text{ vs } H_1 : \sigma^2 < \sigma_0^2$$

The test statistic for testing the above hypothesis is

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2}$$

$$\text{Where } \bar{X} = \frac{\sum_{i=1}^n X_i}{n} \text{ and } S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}$$

Under H_0 , the test statistic follows χ^2 distribution with $(n-1)$ degrees of freedom. We take the decision whether to reject the null hypothesis or not based on P-value. If $P\text{-value} < \alpha$ then we reject the null hypothesis and if $P\text{-value} \geq \alpha$ then we do not enough evidence to reject the null hypothesis. The P-value is calculated as

$$\text{For a) } H_1 : \sigma^2 \neq \sigma_0^2, \quad P\text{-value} = 2 \times (1 - P(\chi_{n-1}^2 < \chi^2))$$

$$b) H_1 : \sigma^2 > \sigma_0^2, \quad P\text{-value} = P(\chi_{n-1}^2 > \chi^2)$$

$$c) H_1 : \sigma^2 < \sigma_0^2, \quad P\text{-value} = P(\chi_{n-1}^2 < \chi^2)$$

Where χ^2 follows χ^2 distribution with $(n-1)$ degrees of freedom (i.e. χ_{n-1}^2).

If μ is known then test statistic is $\chi^2 = \frac{\sum_{i=1}^n (X_i - \mu)^2}{\sigma_0^2}$ and is follows χ^2 distribution with n degrees of freedom (i.e. χ_n^2).

In R programming, there is no inbuilt function for chi-square test for variance testing. Here we write the code in R, as per discussed procedure. We discuss the code with the following example 4 and 5.

Example 4: A lifetime of a certain brand of bulb (in hours) produced by his company is as follows

3360 3720 3300 3420 3240 3420 3450 3540 3750 3780

- Test whether the variance is 30000 or not at 5% level.
- Test whether the variance is more than 20000 at 10% level.
- Test whether the variance is less than 33000 at 2% level.

Solution:

i) Here we test, $H_0 : \sigma^2 = \sigma_0^2 = 30000$ against $H_0 : \sigma^2 \neq \sigma_0^2 = 30000$

```
x=c(3360,3720,3300,3420,3240,3420,3450,3540,3750,3780) #data
s.2=33000; #assumed population variance
n=length(x) #size of data
chisqare.stat=(n-1)*var(x)/s.2; #test statistic
#Calculation of p-value here alternative is two sided
if (qchisq(alp/2,n-1)<chisqare.stat)
{p.value=pchisq(chisqare.stat,n-1)}else
{p.value=pchisq(chisqare.stat,n-1)}
# Output
cat("\t \t Chi-square Test for Variance\n",
    "alternative hypothesis: true variance is not equal to" , s.2,"\n",
    "test statistic=",chisqare.stat, "\t", "df=",n-1,"\t","p-value=",p.value);
```

Output:

```

Chi-square Test for Variance
alternative hypothesis: true variance is not equal to 33000
test statistic= 10.10182      df= 9      p-value= 0.6846111

```

Here P-value is $0.6846111 > 0.05$, hence we do not have enough evidence to reject H_0 (i.e. Accept H_0).

ii) Here we test, $H_0 : \sigma^2 \leq \sigma_0^2 = 20000$ against $H_0 : \sigma^2 > \sigma_0^2 = 20000$

```

x=c(3360,3720,3300,3420,3240,3420,3450,3540,3750,3780) #data
s.2=20000; #assumed population variance
n=length(x) #size of data
chisqare.stat=(n-1)*var(x)/s.2; #test statistic
#Calculation of p-value here alternative is greater than type
p.value=1-pchisq(chisqare.stat,n-1);
# Output
cat("\t\t Chi-square Test for Variance\n",
    "alternative hypothesis: true variance greater than" , s.2,"\n",
    "test statistic=",chisqare.stat, "\t", "df=",n-1,"\t","p-value=",p.value);

```

Output:

```

Chi-square Test for Variance
alternative hypothesis: true variance greater than 20000
test statistic= 16.668      df= 9      p-value= 0.05417611

```

Here P-value is $0.05417611 < 0.1$, hence we reject H_0 (i.e. Accept H_1).

iii) Here we test, $H_0 : \sigma^2 \geq \sigma_0^2 = 35000$ against $H_0 : \sigma^2 < \sigma_0^2 = 35000$

```

x=c(3360,3720,3300,3420,3240,3420,3450,3540,3750,3780) #data
s.2=40000; #assumed population
variance
n=length(x) #size of data
chisqare.stat=(n-1)*var(x)/s.2; #test statistic
#Calculation of p-value here alternative is less than type
p.value=pchisq(chisqare.stat,n-1);
# Output
cat("\t\t Chi-square Test for Variance\n",
    "alternative hypothesis: true variance less than" , s.2,"\n",
    "test statistic=",chisqare.stat, "\t", "df=",n-1,"\t","p-value=",p.value);

```

Output:

```

Chi-square Test for Variance
alternative hypothesis: true variance less than 40000
test statistic= 8.334      df= 9      p-value= 0.4991312

```

Here P-value is $0.4991312 > 0.02$, hence we do not have enough evidence to reject H_0 (i.e. Accept H_0).

Example 5: A average yield of mango is 650 per mango tree and random sample of 10 mango trees has the following yield in a year:

760 650 640 560 580 540 620 680 760 780

i) Test whether variance is 6500 or not at 1% level of significance.

- ii) Test whether variance is more than 7500 at 5% level of significance.
- iii) Test whether variance is less than 4500 at 10% level of significance.

Solution:

i) Here μ is known and we test, $H_0 : \sigma^2 = \sigma_0^2 = 6500$ against $H_0 : \sigma^2 \neq \sigma_0^2 = 6500$

```
x=c(760,650,640,560,580,540,620,680,760,780) #data
mu=650; #population mean
s.2=6500; #assumed population variance
n=length(x) #size of data
chisqare.stat=sum((x-mu)^2)/s.2; #test statistic
#Calculation of p-value here alternative is two sided
p.value=2*(1-pchisq(chisqare.stat,n))
# Output
cat("\t \t Chi-square Test for Variance\n",
    "alternative hypothesis: true variance is not equal to" , s.2,"\n",
    "test statistic=",chisqare.stat, "\t", "df=",n,"\t","p-value=",p.value);
```

Output:

```
Chi-square Test for Variance
alternative hypothesis: true variance is not equal to 6500
test statistic= 10.47692      df= 10      p-value= 0.7993858
```

Here P-value is $0.7993858 > 0.01$, hence we do not have evidence to reject H_0 (i.e. Accept H_0).

ii) Here μ is known and we test, $H_0 : \sigma^2 \leq \sigma_0^2 = 7500$ against $H_0 : \sigma^2 = \sigma_0^2 > 7500$

```
x=c(760,650,640,560,580,540,620,680,760,780) #data
mu=650; #population mean
s.2=7500; #assumed population variance
n=length(x) #size of data
chisqare.stat=sum((x-mu)^2)/s.2; #test statistic
#Calculation of p-value here alternative is greater than type
p.value=1-pchisq(chisqare.stat,n);
# Output
cat("\t \t Chi-square Test for Variance\n",
    "alternative hypothesis: true variance is greater than" , s.2,"\n",
    "test statistic=",chisqare.stat, "\t", "df=",n,"\t","p-value=",p.value);
```

Output:

```
Chi-square Test for Variance
alternative hypothesis: true variance is greater than 7500
test statistic= 9.08      df= 10      p-value= 0.5245285
```

Here P-value is $0.5245285 > 0.05$, hence we do not have evidence to reject H_0 (i.e. Accept H_0).

iii) Here μ is known and we test, $H_0 : \sigma^2 \geq \sigma_0^2 = 4500$ against $H_0 : \sigma^2 = \sigma_0^2 < 4500$

```
x=c(760,650,640,560,580,540,620,680,760,780) #data
mu=650; #population mean
s.2=4500; #assumed population variance
n=length(x) #size of data
chisqare.stat=sum((x-mu)^2)/s.2; #test statistic
#Calculation of p-value here alternative is less than type
p.value=pchisq(chisqare.stat,n);
# Output
cat("\t \t Chi-square Test for Variance\n",
```

```
"alternative hypothesis: true variance is less than" , s.2,"\\n",
"test statistic=",chisqare.stat, "\\t", "df=",n,"\\t","p-value=",p.value);
```

Output:

```
Chi-square Test for Variance
alternative hypothesis: true variance is less than 4500
test statistic= 15.13333      df= 10      p-value= 0.8727241
```

Here P-value is $0.8727241 > 0.1$, hence we do not have evidence to reject H_0 (i.e. Accept H_0).

9.4 F-test for equality of two variances:

F-test is used to test the variances of the two populations are equal, based on two random samples. We assume that the populations from which, the two samples drawn, are Normal distributions. A random sample of size m observations X_1, X_2, \dots, X_m be drawn from population with unknown variance σ_1^2 and a random sample of size n observations Y_1, Y_2, \dots, Y_n be drawn from population with unknown variance σ_2^2 . We test one of the following null hypothesis (H_0) and alternative hypothesis (H_1) at α level of significance.

- a) H_0 : There is no difference between two population variance i.e. $H_0: \sigma_1^2 = \sigma_2^2$ vs $H_1: \sigma_1^2 \neq \sigma_2^2$
- b) $H_0: \sigma_1^2 \leq \sigma_2^2$ vs $H_1: \sigma_1^2 > \sigma_2^2$
- c) $H_0: \sigma_1^2 \geq \sigma_2^2$ vs $H_1: \sigma_1^2 < \sigma_2^2$

The test statistic for testing the above hypothesis is $F = \frac{S_1^2}{S_2^2}$

Where $S_1^2 = \frac{\sum_{i=1}^m (X_i - \bar{X})^2}{m-1}$, $S_2^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n-1}$, $\bar{X} = \frac{\sum_{i=1}^m X_i}{m}$, and $\bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}$,

Under $H_0: \sigma_1^2 = \sigma_2^2$, the test statistic F follows F distribution with $(m-1, n-1)$ degrees of freedom. We take the decision whether to reject the null hypothesis or not based on P-value. If P-value $< \alpha$ then we reject the null hypothesis and if P-value $\geq \alpha$ then we do not enough evidence to reject the null hypothesis. The P-value is calculated as

- For a) $H_1: \sigma_1^2 \neq \sigma_2^2$, P-value = $2 \times (1 - P(F_{(m-1, n-1)} < F))$
- b) $H_1: \sigma_1^2 > \sigma_2^2$, P-value = $P(F_{(m-1, n-1)} > F)$
- c) $H_1: \sigma_1^2 < \sigma_2^2$, P-value = $P(F_{(m-1, n-1)} < F)$

Where F follows F distribution with $(m-1, n-1)$ degrees of freedom.

In R programming, there is inbuilt function **var.test()** for F test for testing equality of two variances. We will demonstrate the var.test() function by **Example 6**.

Example 6: The yield of two varieties of mango (in tons) on two independent sample of 10 and 12 plants are given below.

Variety-A: 22 24 26 23 26 30 32 34

Variety-B: 28 25 26 30 32 30 33 28 30 35

- i) Test whether the variance of variety-A is not equal to Variety-B at 5% level of significance.

- ii) Test whether the variance of variety-A is greater than Variety-B at 10% level of significance.
- iii) Test whether the variance of variety-A is less than Variety-B at 1% level of significance.

Solution:

i) Here we test $H_0 : \sigma_1^2 = \sigma_2^2$ against $H_1: \sigma_1^2 \neq \sigma_2^2$

```
x=c(22,24,26,23,26,30,32,34)           #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)      #second sample data
var.test(x,y)                          #by default alternative is two sided and los=5%
```

Output:

```
F test to compare two variances

data:  x and y
F = 2.0141, num df = 7, denom df = 9, p-value = 0.3238
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4798759 9.7142569
sample estimates:
ratio of variances
 2.014062
```

Here P-value is $0.3238 > 0.05$, Hence we do not have enough evidence to reject H_0 . (i.e. Accept H_0). Output also gives 95% confidence interval for ratio of variance with their sample estimates which also support the decision taken from P-value.

ii) Here we test $H_0 : \sigma_1^2 \leq \sigma_2^2$ against $H_1: \sigma_1^2 > \sigma_2^2$

```
x=c(22,24,26,23,26,30,32,34)           #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)      #second sample data
var.test(x,y,alternative = "greater",conf.level = 0.9)
```

Output:

```
F test to compare two variances

data:  x and y
F = 2.0141, num df = 7, denom df = 9, p-value = 0.1619
alternative hypothesis: true ratio of variances is greater than 1
90 percent confidence interval:
 0.8039161      Inf
sample estimates:
ratio of variances
 2.014062
```

Here P-value is $0.1639 > 0.10$, Hence we do not have enough evidence to reject H_0 . (i.e. Accept H_0).

iii) Here we test $H_0 : \sigma_1^2 \geq \sigma_2^2$ against $H_1: \sigma_1^2 < \sigma_2^2$

```
x=c(22,24,26,23,26,30,32,34)           #first sample data
y=c(28,25,26,30,32,30,33,28,30,35)      #second sample data
var.test(x,y,alternative = "less",conf.level = 0.99)
```

Output:

```
F test to compare two variances

data:  x and y
F = 2.0141, num df = 7, denom df = 9, p-value = 0.8381
alternative hypothesis: true ratio of variances is less than 1
99 percent confidence interval:
 0.00000 13.53198
sample estimates:
ratio of variances
 2.014062
```

Here P-value is $0.8381 > 0.01$, Hence we do not have enough evidence to reject H_0 . (i.e. Accept H_0).

9.5 References:

- Verzani, J. (2014). *Using R for introductory statistics*. CRC Press.
 - Rajagopalan V. (2006). *Selected Statistical Tests*. New Age International (P) limited, Publishers
-

Chapter 10

Simple Random Sampling

Mrs. Shailaja J. Rane, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

10.1 Introduction

Sampling is the process of selecting units (e.g., people, organizations) from a population of interest so that by studying the **sample** we may fairly generalize our results back to the population from which they were chosen.

Sampling methods are classified as either *probability* or *nonprobability*. In probability samples, each member of the population has a known non-zero probability of being selected. Probability methods include random sampling, systematic sampling, and stratified sampling. In nonprobability sampling, members are selected from the population in some nonrandom manner. These include convenience sampling, judgment sampling, quota sampling, and snowball sampling. The advantage of probability sampling is that sampling error can be calculated. Sampling error is the degree to which a sample might differ from the population. When inferring to the population, results are reported plus or minus the sampling error. In nonprobability sampling, the degree to which the sample differs from the population remains unknown.

10.2 Simple Random sampling

Random sampling is the purest form of probability sampling. Each member of the population has an equal and known chance of being selected. When there are very large populations, it is often difficult or impossible to identify every member of the population, so the pool of available subjects becomes biased. This process and technique is known as **simple random sampling**.

In small populations and often in large ones, such sampling is typically done "**without replacement**", i.e., one deliberately avoids choosing any member of the population more than once. Although simple random sampling can be conducted with replacement instead, this is less common and would normally be described more fully as simple random sampling **with replacement**. Sampling done without replacement is no longer independent, but still satisfies exchangeability, hence many results still hold. Further, for a small sample

from a large population, sampling without replacement is approximately the same as sampling with replacement, since the odds of choosing the same individual twice is low.

Result 1: *In simple random sampling with replacement and without replacement (SRSWOR), the sample mean is an unbiased estimate of population mean. (i.e., $E(\bar{y}) = \bar{Y}$, \bar{y} is the sample mean and \bar{Y} is the Population mean.)*

Result 2: *In simple random sampling without replacement (SRSWOR), the variance of sample mean is given as $V(\bar{y}) = \frac{N-n}{Nn} S^2$ (Where S^2 is Population mean square given as $S^2 = \sum_1^n (Y_i - \bar{Y})^2 / N$)*

Result 3: *In simple random sampling with replacement (SRSWOR), the variance of sample mean is given as $V(\bar{y}) = \sigma^2/n$*

10.3 Examples

R programming helps us in selecting different samples with replacement and without replacement using inbuilt functions. Also it helps us check the above results as follows:

1. A population contains 4 units with values 8,3,1,11. List out all possible values of sample size 2 using SRS (i) With replacement (ii) Without replacement.
2. A population contains 10 units with values 8, 3, 1, 11, 22, 15, 37, 50, 99, 82. List out all possible values of sample size 3 using SRS (i) With Replacement (ii) Without replacement.
3. A population contains 10 units with values 8, 3, 1, 11, 12, 15, 22, 40, 52, 70. List out all possible values of sample size 2 using SRS (i) With Replacement (ii) Without replacement.

#Q1 ## WITHOUT REPLACEMENT ##

```
s=c(0,0) #Initialization
for (i in 1:3)
{
  for (j in 2:4)
  {
    if (i<j)
    {
      s=rbind(s,c(i,j)) #The Coop generates all possible samples starting with (0,0)
    }
  }
}
s=s[-1,] #Removes combination (0,0)
Y=c(8,3,1,11)
y=c(1,1)
for(i in 1:6)
{
```

```

y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y) #Calculate means of y
Ybar=mean(Y) #Calculate means of Y
Eybar=mean(ybar) #Calculate means of ybar
Vybar=5/6*var(ybar)
cat("Ybar=",Ybar, "VY=",2/8*var(Y), "Eybar=",Eybar, "Vybar=",Vybar)

```

OUTPUT:

```
Ybar= 5.75 VY= 5.229167 Eybar= 5.75 Vybar= 5.229167>
```

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

#Q2 ## WITH REPLACEMENT ##

```

s=c(0,0)
for(i in 1:4)
{
  for(j in 1:4)
  {
    #if(i<=j)
    {
      s=rbind(s,j)
    }
  }
}
s=s[-1,]
Y=c(8,3,1,11)
y=c(1,1)
for(i in 1:16)
{
  y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y)
Ybar=mean(Y)
Eybar=mean(ybar)
Vybar=var(ybar)
cat("Ybar=",Ybar, "VY=",3/4*var(Y)/2, "Eybar=",Eybar, "Vybar=",15/16*Vybar/2)

```

```
Output: Ybar= 5.75 VY= 7.84375 Eybar= 5.75 Vybar= 7.84375>
```

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

#Q3 ## WITHOUT REPLACEMENT WITH n=2,N=10 ##

```
s=c(0,0)
for(i in 1:10)
{
  for(j in 1:10)
  {
    if(i<j)
    {
      s=rbind(s,c(i,j))
    }
  }
}
s=s[-1,]
Y=c(8,3,1,11,22,15,37,50,99,82)
y=c(1,1)
for(i in 1:45)
{
  y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y)
Ybar=mean(Y)
Eybar=mean(ybar)
Vybar=var(ybar)
cat("Ybar=",Ybar, "VY=",8/20*var(Y), "Eybar=",Eybar, "Vybar=", 44/45*Vybar)
```

Output: Ybar= 32.8 VY= 468.4267 Eybar= 32.8 Vybar= 468.4267>

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

Q4. #WITH REPLACEMENT WITH n=2,N=10

```
s=c(0,0)
```



```

for(i in 1:10)
{
  for(j in 1:10)
  {
    #if(i<j)
    {
      s=rbind(s,c(i,j))
    }
  }
}
s=s[-1,]
Y=c(8,3,1,11,22,15,37,50,99,82)
y=c(1,1)
for(i in 1:100)
{
  y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y)
Ybar=mean(Y)
Eybar=mean(ybar)
Vybar=var(ybar)
cat("Ybar=",Ybar, "VY=",9/10*var(Y)/2,"Eybar=",Eybar,"Vybar=",99/100*Vybar)

```

Output: Ybar= 32.8 VY= 526.98 Eybar= 32.8 Vybar= 526.

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

```
#Q5 ## WITHOUT REPLACEMENT# n=3 N=10 ##
```

```

s=c(0,0)
for(i in 1:10)
{
  for(j in 1:10)
  {
    for(k in 1:10)
    {
      if(i<j & j<k)
      {
        s=rbind(s,c(i,j,k))
      }
    }
  }
}

```

```

s=s[-1,]
Y=c(8,3,1,11,12,15,22,40,52,70)
y=c(1,1)
for(i in 1:1000)
{
y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y)
Ybar=mean(Y)
Eybar=mean(ybar)
Vybar=var(ybar)
cat("Ybar=",Ybar, "VY=",7/30*var(Y),"Eybar=",Eybar,"Vybar=",119/120*Vybar)

```

Output: Ybar= 23.4 VY= 123.8326 Eybar= 23.4 Vybar= 123.8326>

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

#Q6 ## WITH REPLACEMENT n= 3, N = 10 ##

```

s=c(0,0,0)
for(i in 1:10)
{
for(j in 1:10)
{
for(k in 1:10)
{
#if(i<j & j<k)
{
s=rbind(s,c(i,j,k))
}
}
}
}
s=s[-1,]
Y=c(8,3,1,11,12,15,22,40,52,70)
y=c(1,1,1)
for(i in 1:1000)
{
y=rbind(y,Y[s[i,]])
}
y=y[-1,]
ybar=rowMeans(y)
Ybar=mean(Y)

```

```

Eybar=mean(ybar)
Vybar=var(ybar)
cat("Ybar=",Ybar, "VY=",9/10*var(Y)/3,"Eybar=",Eybar,"Vybar=",999/1000*Vybar)

```

```

Output: Ybar= 23.4 VY= 159.2133 Eybar= 23.4 Vybar= 159.2133>

```

From the above output we can see that $E(\bar{y}) = \bar{Y}$

and $V(\bar{y}) = \frac{N-n}{Nn} S^2$

where $E(\bar{y})$ is Eybar

\bar{Y} is Ybar

S^2 is Var(Y) and $V(\bar{y})$ is Vybar

Vybar is calculated using the formula $\frac{\sum(y-\bar{Y})^2}{n-1}$ instead of the formula $\frac{\sum(y-\bar{Y})^2}{n}$. Hence Vybar is multiplied by n-1 and divided by n.

10.4 Advantages Simple Random Sampling

Ease of use represents the biggest advantage of simple random sampling. Unlike more complicated sampling methods such as stratified random sampling and probability sampling, no need exists to divide the population into subpopulations or take any other additional steps before selecting members of the population at random.

A simple random sample is meant to be an unbiased representation of a group. It is considered a fair way to select a sample from a larger population, since every member of the population has an equal chance of getting selected.

10.5 Disadvantages Simple Random Sampling:

A sampling error can occur with a simple random sample if the sample does not end up accurately reflecting the population it is supposed to represent. For example, in our simple random sample of 25 employees, it would be possible to draw 25 men even if the population consisted of 125 women and 125 men. For this reason, simple random sampling is more commonly used when the researcher knows little about the population. If the researcher knew more, it would be better to use a different sampling technique, such as stratified random sampling, which helps to account for the differences within the population, such as age, race or gender.

Chapter 11

Stratified Random Sampling

Dr. Asha A. Jindal, Associate Professor and Head, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

11.1 Introduction

In Stratified Random Sampling, heterogeneous population is divided into number of strata. Each strata is homogenous in its characteristics. We select a sample of specified size or using particular method of allocation from each stratum.

Command used for drawing a sample from given population is

```
sample (x, size, replace = FALSE, prob = NULL)
```

Arguments

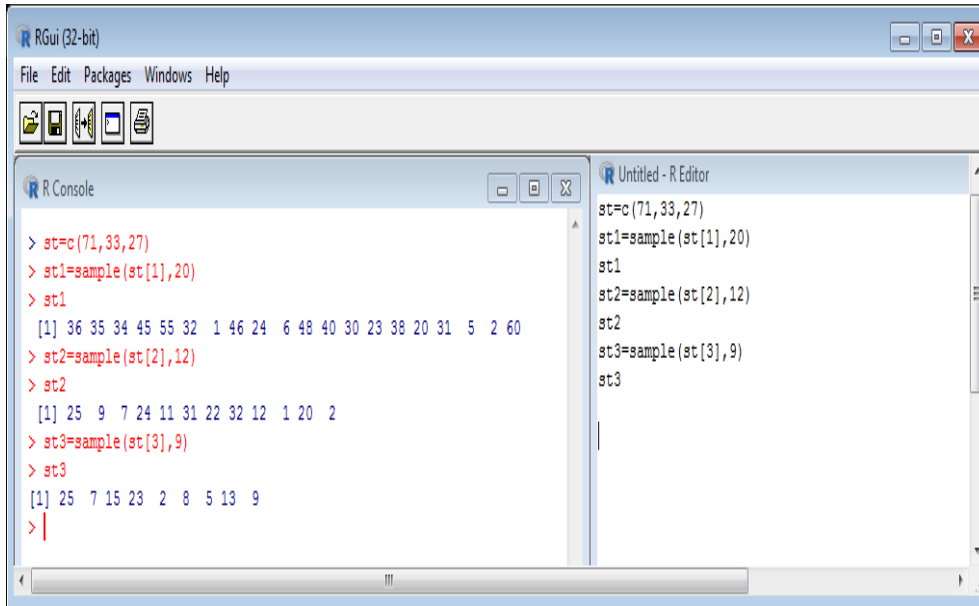
- 'x' either a vector of one or more elements from which to choose, or a positive integer.
- 'Size' a non-negative integer giving the number of items to choose.
- 'replace' should sampling be with replacement?
- 'Prob' a vector of probability weights for obtaining the elements of the vector being sampled.

11.2 Examples

1) Following are data on number of Students enrolled in 3years in Statistics Department in K. C. College:

2)	Class	F.Y.B.Sc.	S.Y.B.Sc.	T.Y.B.Sc.
	No. of Students	71	33	27

Draw a stratified sample of size 20, 12 and 9 from each of the class.

Solution:


```

RGui (32-bit)
File Edit Packages Windows Help

R Console
> st=c(71,33,27)
> st1=sample(st[1],20)
> st1
[1] 36 35 34 45 55 32 1 46 24 6 48 40 30 23 38 20 31 5 2 60
> st2=sample(st[2],12)
> st2
[1] 25 9 7 24 11 31 22 32 12 1 20 2
> st3=sample(st[3],9)
> st3
[1] 25 7 15 23 2 8 5 13 9
>

Untitled - R Editor
st=c(71,33,27)
st1=sample(st[1],20)
st1
st2=sample(st[2],12)
st2
st3=sample(st[3],9)
st3

```

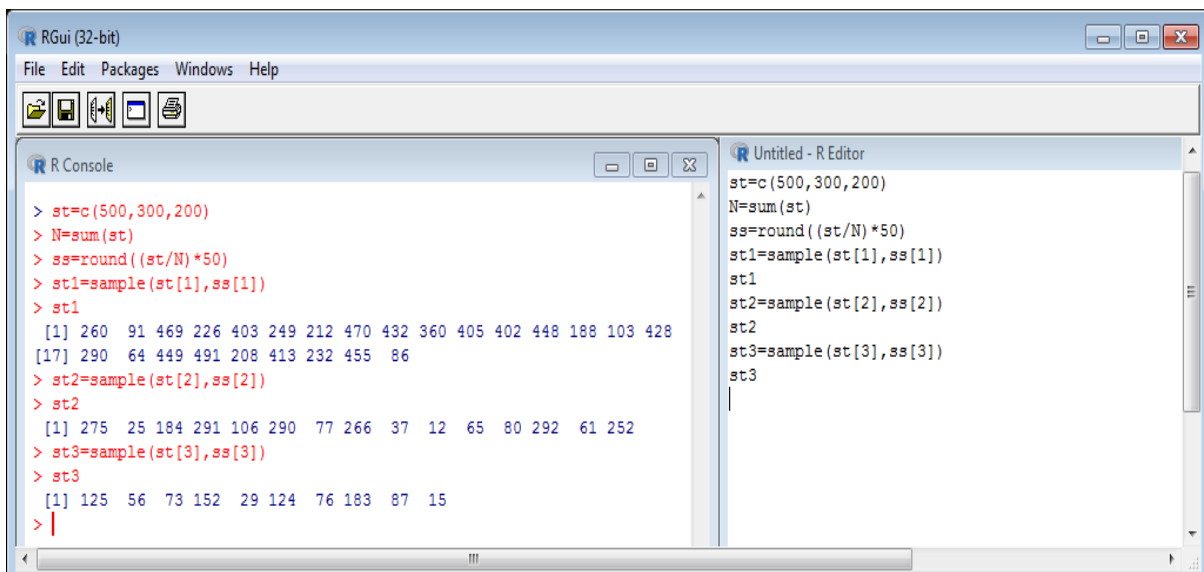
```

> st=c(71,33,27)
> st1=sample(st[1],20)
> st1
[1] 36 35 34 45 55 32 1 46 24 6 48 40 30 23 38 20 31 5 2 60
> st2=sample(st[2],12)
> st2
[1] 25 9 7 24 11 31 22 32 12 1 20 2
> st3=sample(st[3],9)
> st3
[1] 25 7 15 23 2 8 5 13 9

```

3) Using following data, draw a stratified sample of size 50 from different strata by method of proportional allocation.

Stratum	1	2	3
Size(N_i)	500	300	200

Solution:


```

RGui (32-bit)
File Edit Packages Windows Help

R Console
> st=c(500,300,200)
> N=sum(st)
> ss=round((st/N)*50)
> st1=sample(st[1],ss[1])
> st1
[1] 260 91 469 226 403 249 212 470 432 360 405 402 448 188 103 428
[17] 290 64 449 491 208 413 232 455 86
> st2=sample(st[2],ss[2])
> st2
[1] 275 25 184 291 106 290 77 266 37 12 65 80 292 61 252
> st3=sample(st[3],ss[3])
> st3
[1] 125 56 73 152 29 124 76 183 87 15
>

Untitled - R Editor
st=c(500,300,200)
N=sum(st)
ss=round((st/N)*50)
st1=sample(st[1],ss[1])
st1
st2=sample(st[2],ss[2])
st2
st3=sample(st[3],ss[3])
st3

```

```
> st=c(500,300,200)
> N=sum(st)
> ss=round((st/N)*50)
> st1=sample(st[1],ss[1])
> st1
 [1] 260  91 469 226 403 249 212 470 432 360 405 402 448 188 103 428
[17] 290  64 449 491 208 413 232 455  86
> st2=sample(st[2],ss[2])
> st2
 [1] 275  25 184 291 106 290  77 266  37  12  65  80 292  61 252
> st3=sample(st[3],ss[3])
> st3
 [1] 125  56  73 152  29 124  76 183  87  15
```

Chapter 12

Analysis of Varince (ANOVA) using R

Dr. Kalpana Dilip Phal, Associate Professor and Head,
B.N.Bandodkar College of Science, Thane, Chendani Thane (West) 400601.

12.1 Introduction

In this chapter a very popular statistical tool namely Analysis of variance(ANOVA) has been explained . Statistical analysis of i)One way classified data or ii)Two way classified data is explained and with the help of R code the execution is shown, together with interpretations of R output.

12.2 ANOVA

Test of significance for the difference between two population means can be carried out using t-test, under certain set of assumptions .But in many situations like biological or agricultural experiments we come upon a problem of comparing more than 2 population means. For example effect of different conditions on seed germination is same or does it differ significantly? Different types of feed on animals do have same gain in weight? etc. We are also interested in knowing what is the effect of various independent factors on the response or dependent variable. For example How yield of paddy crop responses towards different fertilizers used such as vermi compost, bio compost or chemical fertilizers. Analysis of variance is a powerful tool for both of these purposes.

Variations in observations of a data set is inherited. According to father of Dr. R. A. Fisher the causes of these variations may be broadly classified as assignable and chance causes. In anova the estimate of total variations are split up into variations due to various independent factors Some of which are assignable and remaining variation is due to chance factor. The variation is due to chance factor are experimental error

In anova following assumptions are made

- i) Model applied is linear ii) Various effects influencing response variable are additive
- iii) Observations are independent and iv) Errors are normally distributed IID r.v.s

According to the number of factors variations those influence response variable experiment yields are considered as i)One way classified data or ii)Two way classified data etc.

12.2.1 One way ANOVA

Here Y the response variable is influenced by one factor, usually called as treatments

Model : y_{ij} is the response of j^{th} experimental unit receiving i^{th} treatment

$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$ where $i=1$ to p and $j=1$ to r_i , $n = \sum r_i$

Assumptions 1) Model is additive 2) μ is general mean

3) ε_i follows $IN(0, \sigma^2)$ 4) ε_{ii} are independent 5) α_i effect of i^{th} treatment is fixed effect.

The hypothesis we want to test regarding homogeneity of various treatment means in population which reduces to

$H_0: \alpha_1 = \alpha_2 = \dots \alpha_p = 0$ against H_1 : They differ significantly.

ANOVA table

Source	d.f	S.S	MSS	F ratio
Between/treatment	$p-1$	$SS_{\text{treatment}} = \sum_{i=1}^p \frac{y_i^2}{r_i} - \frac{y_{..}^2}{N}$	$\frac{SS_{\text{treatment}}}{p-1}$	$\frac{MS_{\text{treatment}}}{MS_{\text{error}}}$
Within/error	$n-p$	+		
Total	$n-1$	$\sum_i \sum_j y_{ij}^2 - \frac{y_{..}^2}{n}$		

If calculated F ratio $> F_{\alpha, p-1, n-p}$, then H_0 is rejected. We conclude that treatments differ significantly at confidence level $\alpha\%$ (usually $\alpha = 5\%$ or 1%). MS_{error} is treated as an unbiased estimate of σ^2 . The test of significance of all treatments simultaneously may exhibit significant differences in the means of treatment, but multiple comparison test for pairs of treatments guarantees which treatment means differ significantly.

I) Critical difference C.D:

$t_{(n-p), \frac{\alpha}{2}}$ is two tailed $\alpha\%$ value of t distribution with $n-p$ d.f. C.D = $t_{(n-p), \frac{\alpha}{2}} \cdot \sqrt{MS_{\text{error}}}$

$|\bar{y}_i - \bar{y}_j| > \text{C.D}$ Then we conclude i^{th} treatment shows significant difference from j^{th} treatment

II) Tukey's Honest significant difference test : $|\bar{y}_i - \bar{y}_j| > q_{\alpha, p, n-p} \sqrt{\frac{MS_{\text{error}}}{n}}$ Where $q_{\alpha, p, n-p}$ is studentised range for which tables are available.

12.2.2 Two way ANOVA (r observations per cell)

Here there are two factors A and B say, influencing Y variable. The case with r observation per cell is discussed here.

Model : y_{ijk} is the response of k^{th} experimental unit receiving i^{th} level of factor A and j^{th} level of factor B

$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk}$ where $i=1$ to p , $j=1$ to q , $k=1$ to r

Assumptions 1) Model is additive 2) μ is general mean

3) ε_{ijk} follows $IN(0, \sigma^2)$ 4) ε_{ijk} are independent 5) α_i effect of i^{th} level of factor A and β_j is effect of j^{th} level of factor B . γ_{ij} is interaction effect between i^{th} level of factor A and j^{th} level of and are fixed effects.

$$SS_{\text{total}} = SSA + SSB + SSAB + SS_{\text{error}}$$

ANOVA

Source	d.f	S.S	MSS	F ratio
Factor A	p-1	$SSA = \sum_{i=1}^p \frac{y_{i..}^2}{qr} - \frac{y_{...}^2}{pqr}$	$\frac{SSA}{p-1}$	F_A
Factor B	q-1	$SSB = \sum_{j=1}^q \frac{y_{.j.}^2}{pr} - \frac{y_{...}^2}{pqr}$	$\frac{SSB}{q-1}$	F_B
Factor AB	(p-1)(q-1)	$SSAB = \sum_{j=1}^q \sum_{i=1}^p \frac{y_{ij.}^2}{r} - \sum_{i=1}^p \frac{y_{i..}^2}{qr} - \sum_{j=1}^q \frac{y_{.j.}^2}{pr} + \frac{y_{...}^2}{pqr}$	$\frac{SSAB}{(p-1)(q-1)}$	F_{AB}
Residual	pq(r-1)	$\sum_k \sum_i \sum_j y_{ijk}^2 - \sum_i \sum_{j=1}^q \frac{y_{ij.}^2}{r}$	MSresidual	
Total	pqr-1	$\sum_k \sum_i \sum_j y_{ijk}^2 - \frac{y_{...}^2}{n}$		

First test 1) $H_0 : \gamma_{ij} = 0$ for all i,j

$F_{AB} = MSAB / MS_{\text{error}}$, $F_{AB} > F_{\alpha, (p-1)(q-1), pq(r-1)}$, then conclude that there is interaction between two factors. It makes no sense in carrying out following test. Rather we must held one level of factor A constant and test H_{0B} using one way ANOVA. And we must held one level of factor B constant and test H_{0A} using one way ANOVA .

2) $H_{0A} : \alpha_1 = \alpha_2 = \dots \alpha_p = 0$ against H_{1A} : They differ significantly . $F_A = MSA / MS_{\text{residual}}$

3) $H_{0B} : \beta_1 = \beta_2 = \dots \beta_q = 0$ against H_{1B} : They differ significantly . $F_B = MSB / MS_{\text{residual}}$

12.2.3 Two way ANOVA (one observations per cell)

Model : y_{ij} is the response unit receiving i^{th} level of factor A and j^{th} level of factor B

$y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$ where $i=1$ to p and $j=1$ to q , $n=pq$

$$SS_{\text{total}} = SSA + SSB + SS_{\text{error}}$$

ANOVA

Source	d.f	S.S	MSS	F ratio
Factor A	p-1	$SSA = \sum_{i=1}^p \frac{y_{i.}^2}{q} - \frac{y_{..}^2}{pq}$	$\frac{SSA}{p-1}$	$\frac{MSSA}{MS_{\text{error}}}$
Factor B	q-1	$SSB = \sum_{j=1}^q \frac{y_{.j}^2}{p} - \frac{y_{..}^2}{pq}$	$\frac{SSB}{q-1}$	$\frac{MSSB}{MS_{\text{error}}}$

Error	$(p-1)(q-1)$	$\sum_k \sum_i \sum_j y_{ijk}^2 - \sum_i \sum_{j=1}^q \frac{y_{ij..}^2}{r}$	$\text{MSerror} = \frac{\text{SSerror}}{(p-1)(q-1)}$	
Total	$pq-1$	$\sum_k \sum_i \sum_j y_{ijk}^2 - \frac{y_{...}^2}{n}$		

The hypothesis we want to test regarding homogeneity of various means of

i)factor A and ii)factor B in population which reduces to

i) $H_{0A}: \alpha_1 = \alpha_2 = \dots \alpha_p = 0$ against H_{1A} : They differ significantly .

ii)If calculated F ratio $> F_{\alpha, p-1, n-1}$,then H_{0A} is rejected. We conclude that means of levels of factor A differ significantly at $\alpha \%$.

i) $H_{0B}: \beta_1 = \beta_2 = \dots \beta_q = 0$ against H_{1B} : They differ significantly .

ii)If calculated F ratio $> F_{\alpha, q-1, n-1}$,then H_{0B} is rejected. We conclude that means of levels of factor B differ significantly at $\alpha \%$.

R code for ANOVA

Examples 1: Th grade point average (GPA-4 point scale) of students participating in college sports program are compared .The data are as under.

Football	Tennis	Hockey
3.2	3.8	2.6
2.6	3.1	1.9
2.4	2.6	1.7
2.4	3.9	2.5
1.8	3.2	1.9

Do different sports have significant effect on GPA? .Apply Tuckey's multiple comparison test.

Solution . Here we apply ANOVA on way as GPA are classified according to one factor = sports

```
#data should be read treatment wise #To read treatments
>GPA=c(3.2,2.6,2.4,2.4,1.8,3.8,3.1,2.6,3.9,3.3,2.6,1.9,1.7,2.5,1.9)
>Sport=c(rep("Football",5),rep("Tennis",5),rep("Hockey",5))
>d=data.frame(Sport,GPA)
# anova oneway
>av1=aov(GPA~Sport,data=d)
>summary(av1)
```

Output:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Sport	2	3.929	1.9647	8.456	0.00511**
Residuals	12	2.788	0.2323		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Interpretation: As F calculated is highly significant(**)Treatments differ significantly sports person's GPA differ according sport. We apply Tuckey's test for comparing sports pairwise.

```
>TukeyHSD(av1,"Sport",ordered=F,conf.level=0.95)
```

```
# One can also use plot(TukeyHSD(av1,"Sport"))
```

Output:

Tukey multiple comparisons of means 95% family-wise confidence level

Fit: aov(formula = GPA ~ Sport, data = d)\$

Sport	diff	lwr	upr	p adj
Hockey-Football	-0.36	-1.17329741	0.4532974	0.4860718
Tennis-Football	0.86	0.04670259	1.6732974	0.0381404
Tennis-Hockey	1.22	0.40670259	2.0332974	0.0046180

Interpretation: No sport shows significant difference in GPA means

Example2 : Four varieties of wheat are planted at 3 different locations and their yields (units per plot)are recorded as below.:

Variety↓ Location→	Location 1	Location 2	Location 3
Variety1	14.3	7.6	19.2
Variety2	13.4	3.9	12.6
Variety3	18.4	13.4	15.1

Carry out analysis to check whether different locations or different varieties have significant effect on yield of wheat?..

Solution:

```
#data should be read variety wise
```

```
>yield=c(14.3,13.4,18.4,7.6,3.9,13.4,19.2,12.6,15.1)
```

```
>loc=c(rep("L1",3),rep("L2",3),rep("L3",3))
```

```
>variety=c("V1","V2","V3","V1","V2","V3","V1","V2","V3")
```

```
>result=aov(yield~ loc+variety)
```

```
>summary(result)
```

Output:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
loc	2	103.79	51.89	6.389	0.0568
variety	2	49.79	24.89	065	0.1559
Residuals	4	32.49	8.12		

Interpretation: The Calculated F ratio are not significant, as p value is > .05 The yield does not change significantly as location changes. Even the differences in varieties do not have significant influence on yield. Varieties do not differ significantly.

Example 3: An engineer suspects that surface finish of a metal part is influenced by type of paint used and drying time.Drying times are selected by him are 20,25,30 minutes. and he

randomly choses paint I, II. Conducted experiment yielded following data analyse it. Is there any interaction present between paint and drying time?

paint↓	Drying Times(minutes)		
	20	25	30
I	74,64,50	73,61,44	78,85,92
II	92,86,68	98,73,88	66,45,85

Solution:

```
> DT=c(74,64,50,92,86,68,73,61,44,98,73,88,78,85,92,66,45,85)
> paint=c(rep("I",3),rep("II",3))
> DRT1=c(paint)
> DRT2=c(paint)
> DRT3=c(paint)
> DRT=c("DRT1","DRT2","DRT3")
> d=data.frame(DT,paint,DRT)
> fit=aov(DT~paint*DRT,data=d)
fit=aov(DT~paint*DRT,data=d)
> summary(fit)
```

Output:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
paint	1	356	355.6	1.250	0.285
DRT	2	421	210.4	0.740	0.498
paint:DRT	2	315	157.4	0.553	0.589
Residuals	12	3413	284.4		

Interpretation: Interaction between drying time and paint is not significant. we can perform test for equality of paint means or for drying time means. Using error or error +interaction S.S.

i) $H_{0A}: \alpha_1 = \alpha_2 = \dots \alpha_p = 0$ against H_{1A} : paints differ significantly .ii) Since calculated F ratio $< F_{\alpha,p-1,n-1}$, so H_{0A} is not rejected. We conclude that means of paints do not differ significantly at confidence level 5 %.

ii) $H_{0B}: \beta_1 = \beta_2 = \dots \beta_q = 0$ against H_{1B} : Drying times differ significantly .

ii) Here calculated F ratio $< F_{\alpha,q-1,n-1}$, so H_{0B} is not rejected. We conclude that means of Dryng times do not differ significantly at 5 %.

Chapter 13

Designs of Experiment using R

Dr. Kalpana Dilip Phal, Associate Professor and Head,
B.N.Bandodkar College of Science, Thane, Chendani Thane (West) - 400601

13.1 Introduction

Designs of experiment is a logical construction or plan of the experiment. The inferences drawn after analyzing the data on such experiments are valid, and have very small chance of uncertainty, which may be predefined. This theory is based on principles of i) Replication, ii) Randomization and iii) Local control.

As a prerequisite learner should get acquainted with the terminologies such as plots, treatments, blocks, experimental error etc. for clearness of the subject background.

13.2 Completely randomized design (CRD)

Completely randomized design (CRD) is based on randomization and replication. Here p treatment compared for their effect. i^{th} treatment is applied r_i times completely randomly over experimental units.

The analysis of this design is analogous to ANOVA one way. $SS_{\text{total}} = SS(\text{due to treatment}) + SS_{\text{error}}$

$H_0: \alpha_1 = \alpha_2 = \dots = \alpha_p = 0$ against H_1 : They differ significantly.

ANOVA table

Source	d.f	S.S	MSS	F ratio
/treatment	p-1	$SS_{\text{treatment}} = \sum_{i=1}^p \frac{y_i^2}{r_i} - \frac{y_{..}^2}{N}$	$\frac{SS_{\text{treatment}}}{p-1}$	$F = \frac{MS_{\text{treatment}}}{MS_{\text{error}}}$
/error	n-p	+		
Total	n-1	$\sum_i \sum_j y_{ij}^2 - \frac{y_{..}^2}{n}$		

If calculated F ratio $> F_{\alpha, p-1, n-p}$, then H_0 is rejected. We conclude that treatments differ significantly at confidence level α % (usually $\alpha = 5\%$ or 1%).

13.3 Randomized Block design (RBD)

This design is used when experimental units are not uniform, but they can be grouped into homogeneous strata or groups known as replicate or block. All treatments are randomly applied within each block. So if we have q blocks each is complete replicate of p treatments. Here p treatments compared for their effect. Any treatment is applied q times. The analysis of this design is analogous to ANOVA two way.

$SS_{\text{total}} = SS_{\text{due to treatment}} + SS_{\text{due to block}} + SS_{\text{Error}}$ is

Source	d.f	S.S	MSS	F ratio
Treatmnt	$p-1$	$SST = \sum_{i=1}^p \frac{y_{i..}^2}{q} - \frac{y_{...}^2}{pq}$	$\frac{SST}{p-1}$	$F_T = \frac{MSST}{MSError}$
Block	$q-1$	$SSB = \sum_{j=1}^q \frac{y_{.j.}^2}{p} - \frac{y_{...}^2}{pq}$	$\frac{SSB}{q-1}$	$F_B = \frac{MSSB}{MSError}$
Error	$(p-1)(q-1)$	$\sum_k \sum_i \sum_j y_{ijk}^2 - \sum_i \sum_{j=1}^q \frac{y_{ij.}^2}{r}$	$\frac{MS_{\text{Error}}}{(p-1)(q-1)}$	
Total	$pq-1$	$\sum_k \sum_i \sum_j y_{ijk}^2 - \frac{y_{...}^2}{n}$		

The hypothesis we want to test regarding homogeneity of various means of
i) treatments and ii) Blocks in population which reduces to

i) $H_{0A}: \alpha_1 = \alpha_2 = \dots = \alpha_p = 0$ against H_{1A} : Treatments differ significantly.

ii) $H_{0B}: \beta_1 = \beta_2 = \dots = \beta_q = 0$ against H_{1B} : Blocks differ significantly.

i) If calculated F_T ratio $> F_{\alpha, p-1, n-1}$, then H_{0B} is rejected. We conclude that treatment means differ significantly at α %.

ii) If calculated F_B ratio $> F_{\alpha, q-1, n-1}$, then H_{0B} is rejected. We conclude that block means differ significantly at α %.

13.4 Latin Square design (LSD)

In RBD randomization is restricted. If we come across homogeneous blocks then only we can plan RBD. LSD assumes that variation in treatment should be studied in both perpendicular directions and not to be studied only within block (one direction) in LSD if there are p treatments we plan experiment in $p \times p$ plots and treatments are applied randomly in both directions such that every (horizontal) row arrangement is a complete replicate of p treatments as well as every (vertical) column arrangement is a complete replicate of p treatments.

Model: y_{ijk} is the response of i^{th} row, j^{th} column receiving k^{th} treatment

$y_{ijk} = \mu + \alpha_i + \beta_j + \tau_k + \varepsilon_{ijk}$ where $i=1$ to p , $j=1$ to p , $k=1$ to p

Assumptions 1) Model is additive 2), μ is general mean

3) ε_{ijk} follows $IN(0, \sigma^2)$ 4) ε_{ijk} are independent 5) α_i effect of i^{th} row β_j is effect of j^{th} column.

τ_k is effect due to k^{th} treatment.

$SS_{\text{total}} = SS_{\text{due to treatment}} + SS_{\text{due to row}} + SS_{\text{due to column}} + SS_{\text{error}}$.

ANOVA

Source	d.f	S.S	MSS	F ratio
Row	$p-1$	$SS_{\text{row}} = \sum_{i=1}^p \frac{y_{i..}^2}{p} - \frac{y_{...}^2}{p^2}$	$\frac{SSR}{p-1}$	$F_R = \frac{MSSR}{MSS_{\text{error}}}$
Column	$p-1$	$SS_{\text{column}} = \sum_{j=1}^q \frac{y_{.j.}^2}{p} - \frac{y_{...}^2}{p^2}$	$\frac{SSC}{p-1}$	$F_C = \frac{MSSC}{MSS_{\text{error}}}$
Treatment	$p-1$	$SS_{\text{treat}} = \sum_{k=1}^p \sum_{j=1}^p \frac{y_{.jk}^2}{p} - \frac{y_{...}^2}{p^2}$	$\frac{SST}{(p-1)}$	$F_T = \frac{MSST}{MSS_{\text{error}}}$
Error	$(p-1)(p-1)$	+	M Error	
Total	$n-1$	$\sum_k \sum_i \sum_j y_{ijk}^2 - \frac{y_{...}^2}{n}$		

The hypothesis we want to test regarding homogeneity of various means of i) treatments and ii) Rows iii) Columns in population which reduces to

i) $H_{0A}: \alpha_1 = \alpha_2 = \dots = \alpha_p = 0$ against H_{1A} : Rows differ significantly.

ii) $H_{0B}: \beta_1 = \beta_2 = \dots = \beta_q = 0$ against H_{1B} : Columns differ significantly.

iii) $H_{0C}: \tau_1 = \tau_2 = \dots = \tau_p = 0$ against H_{1C} : Treatments differ significantly.

i) If calculated F_R ratio $> F_{\alpha, p-1, n-1}$, then H_{0A} is rejected. We conclude that row means differ significantly at $\alpha\%$.

ii) If calculated F_C ratio $> F_{\alpha, p-1, n-1}$, then H_{0B} is rejected. We conclude that block means differ significantly at $\alpha\%$.

iii) If calculated F_T ratio $> F_{\alpha, p-1, n-1}$, then H_{0C} is rejected. We conclude that treatment means differ significantly at $\alpha\%$.

Example 1 Athletes are fed three types of diets diet A, diet B and diet C. After the experimentation gain in weights are measured and recorded as below.

Diet Type	Gain in Weight
A	3,6,7,4
B	10,12,11,14,8,6
C	8,3,2,5

Solution:

```
# it is CRD with 3 treatments and unequal no of observations per treatment.
>gainwt=c(3,6,7,4,10,12,11,14,8,6,8,3,2,5)
>diet=c(rep("A",4),rep("B",6),rep("C",4))
```

```
>d=data.frame(gainwt,diet)
>fit=aov(gainwt~diet,data=d)
>summary(fit)
```

Output:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
diet	2	101.10	50.55	7.74	0.00797**
Residuals	11	71.83	6.53		

Signif. codes: 0 0.001 0.01 0.05 0.1 1

Interpretation: The three types of diets have significant effect on gain in weight of athletes.

Example 2 : A researcher wishes to see four brands of gasoline and three types of automobile used have any effect on consumption of gasoline. Following are data on miles per gallon are recorded.

Automobile Brand of Gas.	Automobile		
	Car	Bus	Truck
A	10	11	15
B	10	9	12
C	8	10	11
D	8	8	11

Solution:

```
>miles=c(10,10,8,8,11,9,10,8,15,12,11,11)
>AM=c(rep("Car",4),rep("Bus",4),rep("Truck",4))
>brand=c(rep(A,3),rep(B,3),rep(C,3),rep(D,3))
>d=data.frame(miles,AM,brand)
>fit=aov(miles~brand+AM,data=d)
>summary(fit)
```

Output:

	Df	R	Mean Sq	F value	Pr(>F)
brand	3	10.250	3.417	2.795	0.131
AM	2	26.667	13.333	10.909	0.010*
Residuals	6	7.333	1.222		

Signif. codes: 0 0.001 0.01 0.05 0.1 1

Interpretation: Automobiles differ significantly in mileage at 5% level. Whereas brand means do not differ significantly.

Example3 : Following are results of an experiment in L.S.D. Analyse and comment on your findings.

A	C	B	D
12	19	10	8
C	B	D	A
18	12	6	7

B	D	A	C
22	10	5	21
D	A	C	B
12	7	27	17

Solution:

#LSD with p treatments is conducted over $n=p^2$ plots, p rows p columns.

```
>y=c(12,19,10,8,18,12,6,7,22,10,5,21,12,7,27,17)
>colu=c(1,2,3,4)
>cmn=c(rep(colu,4))
>treatment=c("A","C","B","D","C","B","D","A","B","D","A","C","D","A","B","C")
r>row=c(rep(1,4),rep(2,4),rep(3,4),rep(4,4))
>d=data.frame(y,treatment,row,cmn)
>ft=aov(y~ row+ cmn+ treatment)
summary(fit)
```

Output:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Row	3	94.5	31.50	0.648	0.612
Cmn	3	81.5	27.17	0.559	0.661
Ttreatment	3	315.4	105.13	2.163	0.194
Residuals	6	291.6	48.6		

Interpretation:

i) $H_{0A}: \alpha_1 = \alpha_2 = \dots = \alpha_p = 0$ against H_{1A} : Rows differ significantly .

ii) $H_{0B}: \beta_1 = \beta_2 = \dots = \beta_q = 0$ against H_{1B} : Columns differ significantly .

iii) $H_{0C}: \tau_1 = \tau_2 = \dots = \tau_p = 0$ against H_{1C} : Treatments differ significantly .

i) Here calculated F_R ratio $< F_{\alpha, 3, 6}$, then H_{0A} is rejected. We conclude that row means differ significantly at 5 %.

ii) Here calculated F_C ratio $< F_{\alpha, 3, 6}$, then H_{0B} is rejected. We conclude that block means differ significantly at 5 %.

iii) Here calculated F_T ratio $< F_{\alpha, 3, 6}$, then H_{0C} is rejected. We conclude that treatment means differ significantly at 5 %.

Chapter 14

Linear Programming Problem, Transportation Problem and Assignment problem using R-Software

Dr. S. B. Muley, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020

14.1 General LPP

Objective Function

S.T.

Constraints

Non-Negativity restriction

Package used: lpSolve

14.2 Linear Programming Problem

14.2.1 Usage

```
lp (direction = "min", objective.in, const.mat, const.dir, const.rhs,  
    transpose.constraints = TRUE, int.vec, presolve=0, compute.sens=0,  
    binary.vec, all.int=FALSE, all.bin=FALSE, scale = 196, dense.const,  
    num.bin.solns=1, use.rw=FALSE)
```

14.2.2 Arguments

direction	Character string giving direction of optimization: "min" (default) or "max."
objective.in	Numeric vector of coefficients of objective function
const.mat	Matrix of numeric constraint coefficients, one row per constraint, one column per variable (unless transpose.constraints = FALSE; see below).
const.dir	Vector of character strings giving the direction of the constraint: each value should be one of "<," "<=," "=", "==" , ">," or ">=". (In each pair the two values are identical.)

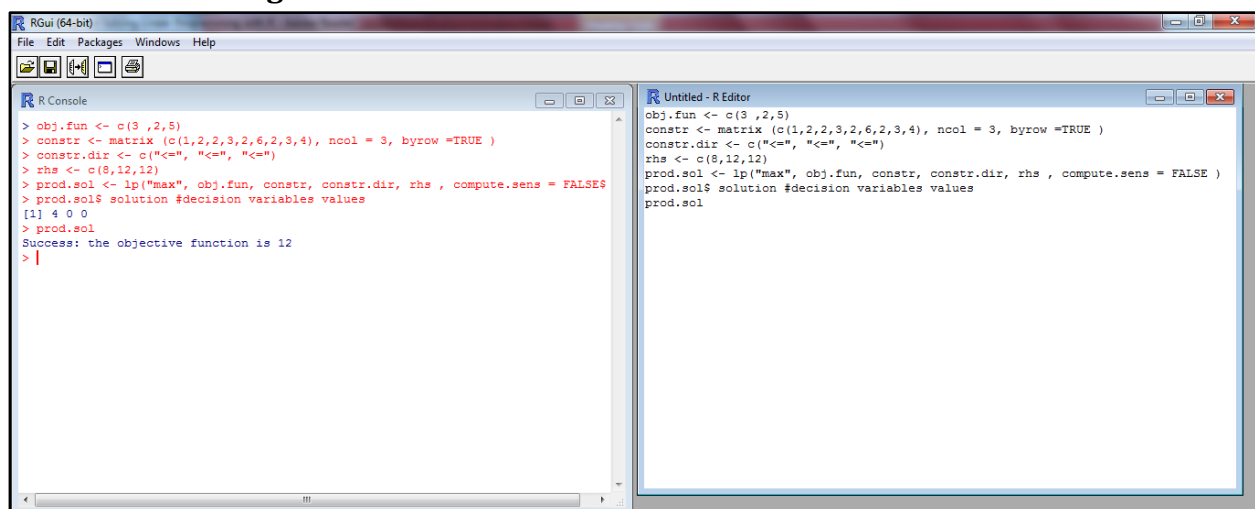
const.rhs	Vector of numeric values for the right-hand sides of the constraints.
transpose.constraints	By default each constraint occupies a row of const.mat, and that matrix needs to be transposed before being passed to the optimizing code. For very large constraint matrices it may be wiser to construct the constraints in a matrix column-by-column. In that case set transpose.constraints to FALSE.

Example 1:

Solve the following LPP upto second simplex table and check whether the solution obtained at the second table is optimum or not.

$$\begin{aligned} \text{Max } Z &= 3x_1 + 2x_2 + 5x_3 \\ \text{Subject to } x_1 + 2x_2 + 2x_3 &\leq 8; \quad 3x_1 + 2x_2 + 6x_3 \leq 12; \quad 2x_1 + 3x_2 + 4x_3 \leq 12; \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

```
# defining parameters
obj .fun <- c(3 ,2,5)
constr <- matrix (c(1,2,2,3,2,6,2,3,4) , ncol = 3, byrow =TRUE )
constr .dir <- c(" <=", " <=", "<=")
rhs <- c(8 , 12 , 12)
#Solving model
prod .sol <- lp("max", obj.fun , constr , constr.dir , rhs , compute . sens = FALSE )
```

R window showing execution:**Output for the LPP:**

```
> obj.fun <- c(3 ,2,5)
> constr <- matrix (c(1,2,2,3,2,6,2,3,4) , ncol = 3, byrow =TRUE )
> constr.dir <- c(" <=", " <=", "<=")
> rhs <- c(8,12,12)
> prod.sol <- lp("max", obj.fun, constr, constr.dir, rhs , compute.sens = FALSE )
```

```
> prod.sol$ solution #decision variables values (x1,x2,x3)
[1] 4 0 0
> prod.sol : Success: the objective function is 12
```

14.3 Transportation Problem

14.3.1 Usages

```
lp.transport (cost.mat, direction="min", row.signs, row.rhs, col.signs,
              col.rhs, presolve=0, compute.sens=0, integers = 1:(nc*nr) )
```

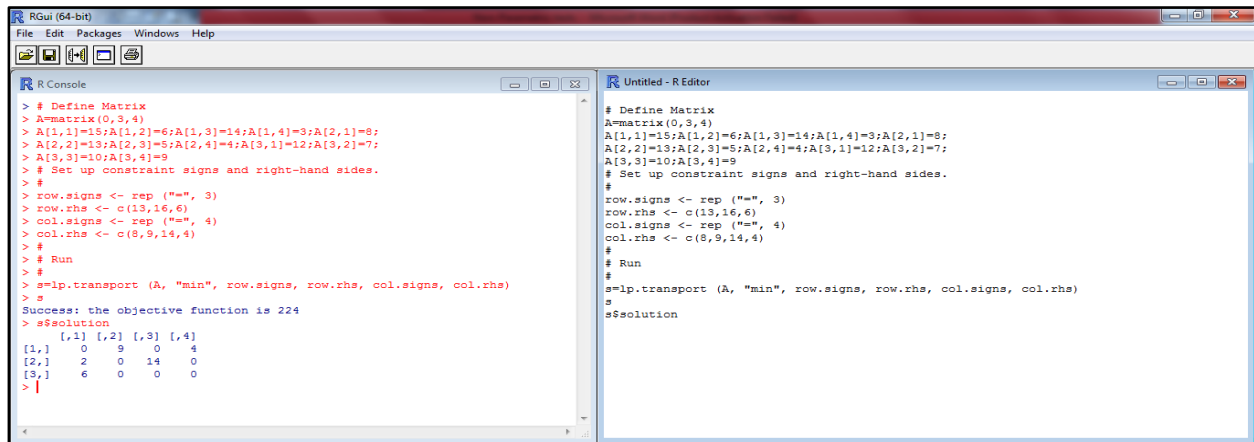
14.3.2 Arguments

cost.mat	Matrix of costs; ij-th element is the cost of transporting one item from source i to destination j.
Direction	Character, length 1: "min" or "max"
row.signs	Vector of character strings giving the direction of the row constraints: each value should be one of "<," "<=," "=", "==" , ">," or ">=." (In each pair the two values are identical.)
row.rhs	Vector of numeric values for the right-hand sides of the row constraints.
col.signs	Vector of character strings giving the direction of the column constraints: each value should be one of "<," "<=," "=", "==" , ">," or ">=."
col.rhs	Vector of numeric values for the right-hand sides of the column constraints.
Presolve	Numeric: presolve? Default 0 (no); any non-zero value means "yes." Currently ignored.
compute.sens	Numeric: compute sensitivity? Default 0 (no); any non-zero value means "yes."
Integers	Vector of integers whose ith element gives the index of the ith integer variable. Its length will be the number of integer variables. Default: all variables are integer. Set to NULL to have no variables be integer.

Example 2:

Obtain the optimum solution for the following transportation problem

Factory	Warehouse				Capacity
	W ₁	W ₂	W ₃	W ₄	
F ₁	15	6	14	3	13
F ₂	8	13	5	4	16
F ₃	12	7	10	9	6
Requirement	8	9	14	4	35

Define matrix of the size row X column**R window showing execution:****Output for the LPP:**

```
> # Define Matrix
> A=matrix(0,3,4)
> A[1,1]=15;A[1,2]=6;A[1,3]=14;A[1,4]=3;A[2,1]=8;
A[2,2]=13;A[2,3]=5;A[2,4]=4;A[3,1]=12;A[3,2]=7;
> A[3,3]=10;A[3,4]=9
> # Set up constraint signs and right-hand sides.
> row.signs <- rep ("=", 3); row.rhs <- c(13,16,6); col.signs <- rep ("=",
4); col.rhs <- c(8,9,14,4)
> # Run
> s=lp.transport (A, "min", row.signs, row.rhs, col.signs, col.rhs);
> s; Success: the objective function is 224
> s$solution
      [,1] [,2] [,3] [,4]
[1,]    0    9    0    4
[2,]    2    0   14    0
[3,]    6    0    0    0
```

14.4 Assignment Problem: lp.assign {lpSolve}**14.4.1 Usage**

```
lp.assign (cost.mat, direction = "min", presolve = 0, compute.sens = 0)
```

14.4.2 Arguments

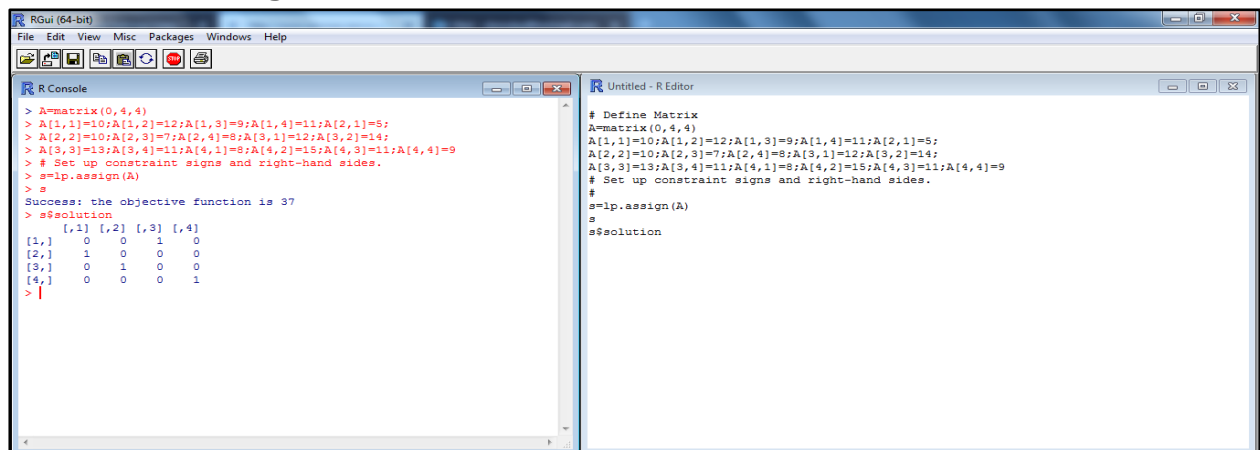
cost.mat	Matrix of costs: the ij-th element is the cost of assigning source i to destination j.
direction	Character vector, length 1, containing either "min" (the default) or "max"

presolve	Numeric: presolve? Default 0 (no); any non-zero value means "yes." Currently ignored.
compute.sens	Numeric: compute sensitivity? Default 0 (no); any non-zero value means "yes." In that case presolving is attempted.

Example 3:

Obtain the optimum solution using Hungarian method for the following assignment table giving costs of doing different jobs on different machines.

Job	Machines			
	M ₁	M ₂	M ₃	M ₄
A	10	12	9	11
B	5	10	7	8
C	12	14	13	11
D	8	15	11	9

R window showing execution:**Output for the LPP:**

```
> A=matrix(0,4,4)
> A[1,1]=10;A[1,2]=12;A[1,3]=9;A[1,4]=11;A[2,1]=5;
> A[2,2]=10;A[2,3]=7;A[2,4]=8;A[3,1]=12;A[3,2]=14;
> A[3,3]=13;A[3,4]=11;A[4,1]=8;A[4,2]=15;A[4,3]=11;A[4,4]=9
> # Set up constraint signs and right-hand sides.
> s=lp.assign(A)
> s
Success: the objective function is 37
> s$solution
      [,1] [,2] [,3] [,4]
[1,]    0    0    1    0
[2,]    1    0    0    0
[3,]    0    1    0    0
[4,]    0    0    0    1
```

Chapter 15

Theory of Estimation

Mrs. Shailaja J. Rane, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

15.1 Introduction

Estimation (or **estimating**) is the process of finding an **estimate**, or approximation, which is a value that is usable for some purpose even if input data may be incomplete, uncertain, or unstable. The value is nonetheless usable because it is derived from the best information available. Typically, estimation involves "using the value of a statistic derived from a sample to estimate the value of a corresponding population parameter". The sample provides information that can be projected, through various formal or informal processes, to determine a range most likely to describe the missing information. An estimate that turns out to be incorrect will be an **overestimate** if the estimate exceeded the actual result, and an **underestimate** if the estimate fell short of the actual result.

Estimation is the process of making inferences from a sample about an unknown population parameter using an estimator. An estimator is a statistic that is used to infer the value of an unknown parameter.

A **point estimate** is the best estimate of the parameter based on a sample. It should be obvious that any point estimate is not absolutely accurate. It is an estimate based on only a single random sample. If repeated random samples were taken from the population, the point estimate would be expected to vary from sample to sample.

A **confidence interval** is an estimate constructed on the basis that a specified proportion of the confidence intervals include the true parameter in repeated sampling. How frequently the confidence interval contains the parameter is determined by the confidence level. 95% is commonly used and means that in repeated sampling 95% of the confidence intervals include the parameter. 99% is sometimes used when more confidence is needed and means that in repeated sampling 99% of the intervals include the true parameter. It is unusual to use a confidence level of less than 90% as too many intervals would fail to include the parameter.

15.2 Uses of estimation

In statistics, an estimator is the formal name for the rule by which an estimate is calculated from data, and estimation theory deals with finding estimates with good properties.

This process is used in signal processing, for approximating an unobserved signal on the basis of an observed signal containing noise.

For estimation of yet-to-be observed quantities, forecasting and prediction are applied. A Fermi problem, in physics, is one concerning estimation in problems which typically involve making justified guesses about quantities that seem impossible to compute given limited available information.

Estimation is important in business and economics, because too many variables exist to figure out how large-scale activities will develop.

Estimation in project planning can be particularly significant, because plans for the distribution of labour and for purchases of raw materials must be made, despite the inability to know every possible problem that may come up. A certain amount of resources will be available for carrying out a particular project, making it important to obtain or generate a cost estimate as one of the vital elements of entering into the project. The U.S. Government Accountability Office defines a cost estimate as, "the summation of individual cost elements, using established methods and valid data, to estimate the future costs of a program, based on what is known today", and reports that "realistic cost estimating was imperative when making wise decisions in acquiring new systems". Furthermore, project plans must not underestimate the needs of the project, which can result in delays while unmet needs are fulfilled, nor must they greatly overestimate the needs of the project, or else the unneeded resources may go to waste.

15.3 Point estimate and Confidence interval:

Case 1 :

Point estimate for μ is \bar{x} .

100(1- α)% confidence interval for Population mean(μ) can be found using the formula

$$P(\bar{x} \pm Z_{\alpha/2} \sigma / \sqrt{n}) = 1 - \alpha$$

R programming helps us to calculate this using inbuilt functions as follows:

Mean(\bar{x}) can be found using the function

```
mean (data)
```

Standard deviation can be found using

```
sd(data)
```

To find $Z_{\alpha/2}$

```
z = qnorm(alp/2, mean = 0, sd = 1, lower.tail = 0)
```

If sample size is small and standard deviation is unknown use t statistic value instead of $Z_{\alpha/2}$.

This can be found using the following inbuilt function.

```
t = qt(alp/2,d.o.f,lower.tail = 0)
```

d.o.f = n-1

Lower limit and Upper limit can be found using R as follows:


```
l=xbar-z*s/sqrt(n);
u=xbar+z*s/sqrt(n)
```

Note: Confidence interval for population mean can also be found using Rmisc . Install Rmisc from packages, cran cloud. Then load and use the following inbuilt function.

CI(X) gives default lower and upper limit

CI(X, ci = 0.95) gives 95% confidence interval for mean

Example:

#Q1

```
xbar=3.5;s=2.61;n=900;alp=0.05
```

```
> z=qnorm(alp/2,0,1,lower.tail=0)
> l=xbar-z*s/sqrt(n);u=xbar+z*s/sqrt(n)
> #95% CI for mean
> paste("(",l,",",u,")")
```

Output

```
[1] "(3.32948313334502, 3.67051686665498)"
```

#Q2

```
x=c(20,16,26,27,23,22,18,24,25,19,18,28,25,27,22)
```

```
> xbar=mean(x);n=length(x);s=sd(x)
> alp=.1
> t=qt(alp/2,n-1,lower.tail=0)
> l=xbar-t*s/sqrt(n);u=xbar+t*s/sqrt(n)
> #90% CI for mean
> paste("(",l,",",u,")")
```

Output

```
[1] "(20.9506711391254, 24.3826621942079)"
```

Case 2:

Point estimation for variance (σ^2) is s^2

100(1- α)% confidence interval for variance using the formula

$P((n-1)s^2/\chi^2(1-\alpha/2, n-1) < \sigma^2 < (n-1)s^2/\chi^2(\alpha/2, n-1)) = 1 - \alpha$

R programming commands used to calculate s^2 and χ^2 table values are as follows:

```
s=sd(x)
c1=qchisq(1-alp/2,n-1,lower.tail=1)
c2=qchisq(alp/2,n-1,lower.tail=1)
l = (n-1)*s^2/c1;
u = (n-1)*s^2/c2
paste("(",l,",",u,")")
```

Example:

```
x=c(20,16,26,27,23,22,18,24,25,19,18,28,25,27,22)
```

```
xbar=mean(x);n=length(x);s=sd(x)
alp=.05
> c1=qchisq(1-alp/2,n-1,lower.tail=1)
```

```
> c2=qchisq(alp/2,n-1,lower.tail=1)
> l=(n-1)*s^2/c1;u=(n-1)*s^2/c2
> #95% CI for population variance
> paste("(",l,",",u,")")
```

Output

```
[1] "(7.63175197521812, 35.4135784339703)"
```

Case 3:**Point estimation for $\mu_1 - \mu_2$ is $\bar{x}_1 - \bar{x}_2$** **100(1- α)% confidence interval for difference of means.**

Formula: $P((\bar{x}_1 - \bar{x}_2) \pm Z_{\alpha/2} (s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}})) 1 - \alpha$

R programming commands used:

```
n1=length(x)
n2=length(y)
xbar=mean(x)
ybar=mean(y)
s1=sd(x) #SD of X
s2=sd(y)
s=sqrt(((n1-1)*s1^2+(n2-1)*s2^2)/(n1+n2-2))
t=qt(.05/2,n1+n2-2,lower.tail=0)
l=xbar-ybar-t*s*sqrt(1/n1+1/n2)
u=xbar-ybar+t*s*sqrt(1/n1+1/n2)
```

Example:**Q1.**

```
> x=c(74,77,74,73,79,76,82,72,75,78,77,78,76,76)
> y=c(70,75,74,70,69,72,76,72,72,77,77,72,75,78,72,74,75)
> n1=length(x)
> n2=length(y)
> xbar=mean(x)
> ybar=mean(y)
> s1=sd(x) #SD of X
> s2=sd(y)
> s=sqrt(((n1-1)*s1^2+(n2-1)*s2^2)/(n1+n2-2))
> t=qt(.05/2,n1+n2-2,lower.tail=0)
> #i)
> l=xbar-ybar-t*s*sqrt(1/n1+1/n2)
> u=xbar-ybar+t*s*sqrt(1/n1+1/n2)
> #95% CI for difference of means
> paste("(",l,",",u,")")
```

Output

```
[1] "(0.733919803307959, 4.63582809585169)"
```

Q2.

```
> n1=7;n2=8;xbar=1234;ybar=1036;s1=34;s2=40
> alp=.05;t=qt(alp/2,n1+n2-2,lower.tail=0)
> l=ybar-t*s2/sqrt(n2);u=ybar+t*s2/sqrt(n2)
```

```
> #90% CI for difference of means
> paste("(",l,"",",",u,"")")
```

Output

```
[1] "(1005.44777346305, 1066.55222653695)"
```

Case 4:

Point estimation for σ_1^2 / σ_2^2 is s_1^2 / s_2^2

100(1- α)% confidence interval for Ratio of variances

Formula : $P(s_1^2 / (s_2^2 F(\alpha/2, n_1 - 1, n_2 - 1)) \leq \sigma_1^2 / \sigma_2^2 \leq s_1^2 F(\alpha/2, n_1 - 1, n_2 - 1) / s_2^2) = 1 - \alpha$

R programming commands used:

```
f=qf(alp/2,n1-1,n2-1,lower.tail=0)
l=s1^2/(s2^2*f)
u=(s1^2*f)/s2^2
paste("(",l,"",",",u,"")")
```

Example:**Q1.**

```
>x=c(74,77,74,73,79,76,82,72,75,78,77,78,76,76)
>y=c(70,75,74,70,69,72,76,72,72,77,77,72,75,78,72,74,75)
>n1=length(x)
>n2=length(y)
>xbar=mean(x)
>ybar=mean(y)
>s1=sd(x) #SD of X
>s2=sd(y)
>s=sqrt(((n1-1)*s1^2+(n2-1)*s2^2)/(n1+n2-2))
>t=qt(.05/2,n1+n2-2,lower.tail=0)
> alp=.01;f=qf(alp/2,n1-1,n2-1,lower.tail=0)
> l=s1^2/(s2^2*f);u=(s1^2*f)/s2^2
> #99% C.I for ratio of variance
> paste("(",l,"",",",u,"")")
```

Output

```
[1] "(0.236138371356271, 3.83768581270189)"
```

#Q2

```
>n1=7;n2=8;xbar=1234;ybar=1036;s1=34;s2=40
> alp=0.1;f=qf(alp/2,n1-1,n2-1,lower.tail=0)
> l=s1^2/(s2^2*f);u=(s1^2*f)/s2^2
> #99% C.I for ratio of variance
> paste ("(",l,"",",",u,"")")
```

Output

```
[1] "(0.18688717562124, 2.79316249638198)"
```

Chapter 16

Financial Functions

Mrs. Pratiksha M. Kadam, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

16.1 Introduction

In this chapter we discuss basic financial functions in R:
To use the financial functions in R, we need the package “FinCal” to be installed from CRAN.
Before we start executing financial functions listed below, we must load package “FinCal”.

Procedure to install “FinCal” Package in R:

In R Gui, Click on Packages menu and select the option “Install package(s)”, Select 0-cloud [https] from the country options and click on OK. Then a list of functions will be displayed. From that list select function “FinCal” and click on Install.

Procedure to Load “FinCal” package in R:

In R Gui, Click on Packages menu and select the option “Load package”. List of installed packages will be shown. From that list select “FinCal” and click on OK.

16.2 Effective Annual Rate (ear() function)

Effective Annual Rate is the actual rate of interest actually earned on an investment or paid on a loan as a result of compounding interest over the given period of time.

Formula:

$$EAR = (1 + r/n)^n$$

r = Nominal rate of interest

n = Number of compounding periods.

Example: Given the nominal rate of 8%, compute the effective annual rates for annual, semiannual, quarterly, monthly, daily and continuous compounding.

R code:

```
> #For Annual
> ear(0.0425, 1)
[1] 0.0425
> #For Semiannual
> ear(0.0425, 2)
[1] 0.04295156
> #For Quarterly
```

```
> ear(0.0425, 4)
[1] 0.04318215
> #For Monthly
> ear(0.0425, 12)
[1] 0.04333772
> #For Daily
> ear(0.0425, 365)
[1] 0.04341347
> #For Continuous
> ear.continuous(0.0425)
[1] 0.04341606
```

16.3 PRESENT VALUE

Present Value: Present value (PV) is the current worth of a future sum of money or stream of cash flows given a specified rate of return.

16.3.1 Present value of a single sum (pv.simple() function):

Formula:

$$PV = FV / ((1 + r)^n)$$

PV = present value

FV = future value required

r = rate per period

n = number of periods

Example: Mr. X wants to save money such that at the end of 15 years he has 10000000 Rs. in his account. He gets 6% interest rate per year. How much amount should he put in the account now?

R code:

```
> # rate= 6% per annum, fv= 1000000, period= 15 years
> pv.simple(r = 0.06, n = 15, fv = 1000000)
[1] -417265.1
```

16.3.2 Present value of an ordinary annuity (pv.annuity() function):

In ordinary annuity, payment is made at the end of the period.

Formula:

$$PV(ordinary) = PMT \left(\frac{(1 - (1 + r)^{-n})}{r} \right)$$

PMT = payment per period

r = rate per period

n = Number of periods

Example: Calculate the PV of an annuity that pays 15000 per annum at the end of each of the next 15 years, given a 8% per annum discount rate?

R code:

```
> #rate=8% per annum, payment per period= 15000, period= 15 years, type=0(as
payment made at the end of the period)
> pv.annuity(r = 0.08, n = 15, pmt = -15000, type = 0)
[1] 128392.2
```

16.3.3 Present value of an annuity due (pv.annuity() function):

In annuity due, payment is made at the end of the period.

Formula:

$$PV(annuity\ due) = PMT \left(\frac{(1 - (1 + r)^{-n})}{r} \right) (1 + r)$$

PMT = payment per period

r = rate per period

n = Number of periods

Example: Given a rate of interest 10% per annum, find the present value of 5-year annuity that makes payment of 2000Rs. at the beginning of every month starting today.

R code:

```
> #rate=10% per annum= .1/12 per month, payment per month= 2000, period= 5
years= 60 months, type=1(as payment made at the beginning of the period)
> pv.annuity(r = 0.1/12, n = 60, pmt = -2000, type = 1)
[1] 94915.16
```

16.3.4 Present value of a perpetuity (pv.perpetuity() function):

Perpetuity is an infinite series of periodic payments of equal face value.

Formula:

$$PV(perpetuity) \frac{PMT}{r}$$

PMT = payment per period

r = rate per period

Example: Calculate the present value of the perpetuity paying 5000 Rs. at the end of every quarter. The yearly discount rate is 12%.

R code:

```
> #rate=12% per annum= .12/4 per qtr, payment per qtr= 2000,type=0(as
payment is made at the end of theperiod)
> pv.perpetuity(r = 0.12/4, pmt = -5000, type = 0)
[1] 166666.7
```

16.3.5 Present value of uneven cash flow (pv.uneven() function):

Formula:

$$PV(uneven\ cf) = \sum_{i=0}^n \frac{cf_i}{(1+r)^i}$$

 cf_i = i^{th} cash flow r = rate per period n = number of periods

Example: Given rate of return 8% per annum, calculate the present value of the following 5-year cash flow stream occurred at the end of each year:

Year	1	2	3	4	5
Cash Flow	10000	-6000	8000	-3000	-2000

R code:

```
> #rate=8% per annum, cash flow:10000, -6000, 8000, 3000, 2000
> pv.uneven(r = 0.08, cf = c(10000, -6000, 8000, 3000, 2000))
[1] -14032.14
```

16.4 FUTURE VALUE

The future value (FV) is the value of a current asset at a specified date in the future based on an assumed rate of growth over time.

16.4.1 Future value of a single sum (fv.simple() function):

Formula:

$$FV = PV (1 + r)^n$$

 PV = present value FV = future value required r = rate per period n = number of periods

Example: Calculate the Future Value of an investment of 1000 Rs. at the end of five years if it earns an annually compounded rate of return of 7.2%.

R code:

```
> # rate= 7.2% per annum, pv= 1000, period= 5 years
> fv.simple(r = 0.072, n = 5, pv = -1000)
[1] -417265.1
```

16.4.2 Future value of an ordinary annuity (fv.annuity() function):

Formula:

$$FV(ordinary) = PMT \left(\frac{((1 + r)^n - 1)}{r} \right)$$

PMT= payment per period*r*= rate per period*n*= Number of periods

Example: Calculate the Future Value of an ordinary annuity that pays 15000 per quarter at the end of each quarter for the next 10 years, Expected rate of return is 8% per year.

R code:

```
> # rate= 8% per annum= .8/4 per qtr, pv= 15000, period= 10 years= 40 qtrs,
type=0(as payment made at the end of the period)
> fv.annuity(r = 0.08/4, n = 40, pmt = -15000, type = 0)
[1] 906029.7
```

16.4.3 Future value of an annuity due (fv.annuity() function):

Formula:

$$FV(annuity\ due) = PMT \left(\frac{((1 + r)^n - 1)}{r} \right) (1 + r)$$

PMT= payment per period*r*= rate per period*n*= Number of periods

Example: Given a rate of interest 12% per annum, find the future value of an annuity that pays 1000Rs. at the beginning of every month starting today for the next 6 years.

R code:

```
> #rate=12% per annum= 0.01 per month, payment per month= 1000, period= 6
years= 72 months, type=1(as payment made at the beginning of the period)
> fv.annuity(r = 0.01, n = 72, pmt = -1000, type = 1)
[1] 105757
```

16.4.4 Future value of uneven cash flow (fv.uneven() function):

Formula:

$$FV(uneven\ cf) = \sum_{i=0}^n cf_i(1 + r)^i$$

cf_i= ith cash flow*r*=rate per period*n*= number of periods

Example: Given rate of return 9% per annum, calculate the future value of the following 5-year cash flow stream occurred at the end of each year:

Year	1	2	3	4	5
Cash Flow	10000	-6000	8000	-3000	-2000

R code:

```
> #rate=9% per annum, cash flow:10000, -6000, 8000, 3000, 2000
> fv.uneven(r = 0.09, cf = c(10000, -6000, 8000, 3000, 2000))
[1] -21120.44
```

16.5 Loan payment calculation (pmt() function)

To find the amount per period

Formula:

$$PMT = \frac{r(PV)}{1 - (1 + r)^{-n}}$$

PMT = payment per period

PV = present value

r =rate per period

n = number of periods

Example: A person decides to take a loan of 500000Rs. for 4 years. Bank lends the money at the rate of 6% per annum and requires that the loan should be paid off in the equal end-of-month payments in 4 years. Calculate the amount the person has to pay at the end of every month to repay the loan.

R code:

```
> # rate= 6% per annum=.06/12 per month, fv= 500000, period= 4 years=48 months
> pmt(r = 0.06/12, n = 48, pv = 500000, fv = 0)
[1] -11742.51
```

16.6 Number of periods of an annuity (n.period() function)

Formula:

$$n = \frac{\ln\left(1 + \frac{FV \times r}{PMT}\right)}{\ln(1 + r)}$$

PMT = payment per period

FV = future value

r =rate per period

n = number of periods

Example: How many years one has to pay the yearly installment of 20000Rs. at the end of each year to get accumulated value as 1000000, if the rate per annum is 9%?

R Code:

```
> #rate=9%per annum, fv=1000000, payment at the end of every year=20000
> n.period(r = 0.09, pv = 0, fv = 1000000, pmt = -20000, type = 0)
[1] 19.78178
```

16.7 Rate of return (discount.rate() function)

Function discount.rate is used to calculate rate of return of the given investment.

Example: Mr. Y want to invests 300Rs. at the end each month for the next 2 years. He gets 500000 Rs. at the end of second year. Calculate annual rate of return of this investment.

R code:

```
> discount.rate(n = 24, fv = 50000, pmt = -300, pv = 0, type = 0)*12
[1] 1.719706
> #rate by the function given by the above function is per month. So to
convert it to per year we multiply by 12
```

16.8 Net Present Value (npv() function)

Net Present Value (NPV) is the difference between the present value of cash inflows and the present value of cash outflows.

Formula:

$$NPV = -cf_0 + \frac{cf_1}{(1+r)} + \frac{cf_2}{(1+r)^2} + \dots + \frac{cf_n}{(1+r)^n} = \sum_{i=1}^n \frac{cf_i}{(1+r)^i} - cf_0$$

cf_i = i^{th} cash flow

r = rate per period

n = number of periods

Example: Mr. Z does an investment with an initial cost of 600000 Rs. and positive cash flows of 300000 at the end of first year, 230000 Rs. at the end of second year, and 480000Rs. at the end of third year and 120000 Rs. at the end of fourth year. Calculate the net present value of this investment if the discount rate is 8.5%.

R code:

```
> #rate= 8.5%, cash flow: -600000, 300000, 230000, 480000, 120000
> npv(r = 0.085, cf = c(-600000, 300000, 230000, 480000, 120000))
[1] 334257.2
```

16.9 Internal Rate of Return (irr() function)

Internal rate of return (IRR) is the interest rate at which the net present value of all the cash flows (both positive and negative) from a project or investment equal zero.

Example: Consider the example given in Net Present Value calculation and calculate the internal rate of return for the investment mentioned there.

R code:

```
> #cash flow: -600000, 300000, 230000, 480000, 120000  
> irr(cf = c(-600000, 300000, 230000, 480000, 120000))  
[1] 0.3278433
```

16.10 References

1. www.investopedia.com
 2. R for Beginners, Emmanuel Paradis
 3. Package 'FinCal', August 29, 2016.
-

Chapter 17

Non-Parametric Test

Dr. S. B. Muley, Assistant Professor, Department of Statistics,
K. C. College, Churchgate, Mumbai – 400 020.

17.1 Introduction

Hypothesis testing is the key are of elementary statistics. Different statistical tests and their uses to test different types of hypothesis on the basis of different observed data points has always remained at the centre of discussion in the interdisciplinary fields. Statistician categorises the test as parametric and Non-parametric. Parametric methods in elementary statistics that assume the data is quantitative, the population has a normal distribution and the sample size is sufficiently large.

On the contrary a statistical method is called non-parametric if it makes no assumption on the population distribution or sample size. This approach is less powerful yet more frequent, more flexible, more robust, and applicable to non-quantitative data.

- One sample tests:
 - Sign test
 - Wilcoxon Signed rank test
- Two sample tests:
 - Independent sample test
 - Mann-Whitney U test
 - Paired sample test
 - Wilcoxon test:
- More than two sample tests:
 - Independent sample test
 - Kruskal-Wallis Test:
 - Paired sample test:
 - Friedman Test:

The packages used:

- BSDA

17.2 One sample tests

17.2.1 Sign test

Appropriate data

- One-sample data.
- Data are ordinal, interval, or ratio.

Description

This function will test a hypothesis based on the sign test and reports linearly interpolated confidence intervals for one sample problems.

Usage

```
SIGN.test(x, y = NULL, md = 0, alternative = "two.sided", conf.level = 0.95)
```

Arguments

X	numeric vector; NAs and Infs are allowed but will be removed.
Y	optional numeric vector; NAs and Infs are allowed but will be removed.
Md	a single number representing the value of the population median specified by the null hypothesis
alternative	is a character string, one of "greater", "less", or "two.sided", or the initial letter of each, indicating the specification of the alternative hypothesis. For one-sample tests, alternative refers to the true median of the parent population in relation to the hypothesized value of the median.
conf.level	confidence level for the returned confidence interval, restricted to lie between zero and one
Statistic	the S-statistic (the number of positive differences between the data and the hypothesized median), with names attribute "S".

Example 2:

It is known from the past experience that the median length of Sunfish in a particular polluted lake was 3.9 inches. During the past two years the lake was cleaned up and the conjecture is made that now median length is greater than 3.9 inches. A random sample of 10 sunfish selected from this lake showed lengths as 5.2, 4.1, 5.4, 5.7, 3.0, 6.3, 6.6, 2.8, 1.9, 4.5 inches. Will you reject the null hypothesis at 10 % level of significance (l.o.s.) on the basis of Sign Test?

Manual Calculation:

$H_0 : m = 3.9$

$H_1 : m > 3.9$ ($\alpha = 0.10$)

Let T^- be the number of sunfish with lengths < 3.9

$\therefore T^- = 3$

Test statistic is T^-

Under H_0 ; $T^- \sim B(n = 10, P = \frac{1}{2})$

The C.R. is $C \equiv \{T^-; T^- \leq C\}$ where C is such that $P(\text{Type 1 error}) \leq \alpha$

$$P(T^- \leq C/P = 1/2) \leq 0.10$$

$$\sum_{r=0}^C \binom{10}{r} (0.5)^{10} \leq 0.10$$

$$\text{For } C = 2: \sum_{r=0}^2 \binom{10}{r} (0.5)^{10} = 0.0547$$

$$\text{For } C = 3: \sum_{r=0}^3 \binom{10}{r} (0.5)^{10} = 0.1719$$

$$\therefore C = 2$$

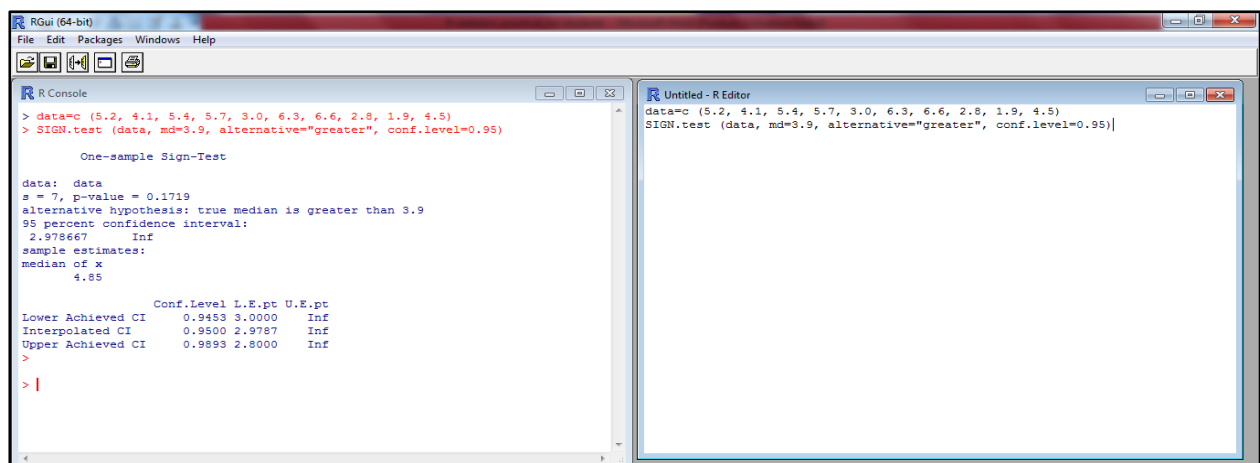
For given sample $T^- = 3 \nless C = 2$

\therefore Do not reject H_0 at 10 % l.o.s.

Note: Test statistics is based on min (No. of positive, No of negative)

R-syntax:

```
> data=c (5.2, 4.1, 5.4, 5.7, 3.0, 6.3, 6.6, 2.8, 1.9, 4.5)
> SIGN.test (data, md=3.9, alternative="greater", conf.level=0.95)
```



One Sample Sign-Test:

```
> data=c (5.2, 4.1, 5.4, 5.7, 3.0, 6.3, 6.6, 2.8, 1.9, 4.5)
> SIGN.test (data, md=3.9, alternative="greater", conf.level=0.90)
```

One-sample Sign-Test

data: data

s = 7, p-value = 0.1719

alternative hypothesis: true median is greater than 3.9

90 percent confidence interval:

3.43 Inf

sample estimates: median of x = 4.85

	Conf.Level	L.E.pt	U.E.pt
Lower Achieved CI	0.8281	4.1000	Inf
Interpolated CI	0.9000	3.4253	Inf
Upper Achieved CI	0.9453	3.0000	Inf

```
> data=c (5.2, 4.1, 5.4, 5.7, 3.0, 6.3, 6.6, 2.8, 1.9, 4.5)
```

```
> SIGN.test (data, md=3.9, alternative="greater", conf.level=0.90)
```

One-sample Sign-Test

```
data: data
s = 7, p-value = 0.1719
alternative hypothesis: true median is greater than 3.9
90 percent confidence interval:
 3.43      Inf
sample estimates: median of x = 4.85
```

	Conf.Level	L.E.pt	U.E.pt
Lower Achieved CI	0.8281	4.1000	Inf
Interpolated CI	0.9000	3.4253	Inf
Upper Achieved CI	0.9453	3.0000	Inf

Note: S the test statistics is no. of positive in the data set.

Interpretation: Since $p\text{-value} = 0.1719 > 0.05$ indicates one should not reject null hypothesis.

17.2.2 Wilcoxon Signed rank test:

Performs one- and two-sample Wilcoxon tests on vectors of data; the latter is also known as 'Mann-Whitney' test.

Usage

```
wilcox.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),
            mu = 0, paired = FALSE, exact = NULL, correct = TRUE,
            conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	numeric vector of data values. Non-finite (e.g. infinite or missing) values will be omitted.
Y	an optional numeric vector of data values: as with x non-finite values will be omitted.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
Mu	a number specifying an optional parameter used to form the null hypothesis. See 'Details'.
Paired	a logical indicating whether you want a paired test.
Exact	a logical indicating whether an exact p-value should be computed.
Correct	a logical indicating whether to apply continuity correction in the normal approximation for the p-value.
conf.int	a logical indicating whether a confidence interval should be computed.

conf.level	confidence level of the interval.
Formula	a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs a factor with two levels giving the corresponding groups.
Data	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
Subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").
...	further arguments to be passed to or from methods.

Example 3:

A random sample of 10 infants showed the following pulse rates per minute: 110,121,125,122,112,117,129,114,124,127. Assuming that the distribution of pulse rates is symmetric. Is there any evidence to suggest that the median pulse rate of infants is more than 120 beats per minute? Use Wilcoxon's signed rank test at 5% l.o.s.

Manual Calculation:

$H_0 : M=120$ $H_1 : M>120$, $\alpha=0.05$ $n = 10$

X_i	$X_i - M_0$	Signed ranks
110	-10	-10
121	1	1
125	5	5
122	2	2
112	-8	-8
117	-3	-3
129	9	9
114	-6	-6
124	4	4
127	7	7

$T^+ = \text{sum of ranks with +ve sign} = 28$

$T^- = \text{sum of ranks with -ve sign} = 27$

Test statistic is T^-

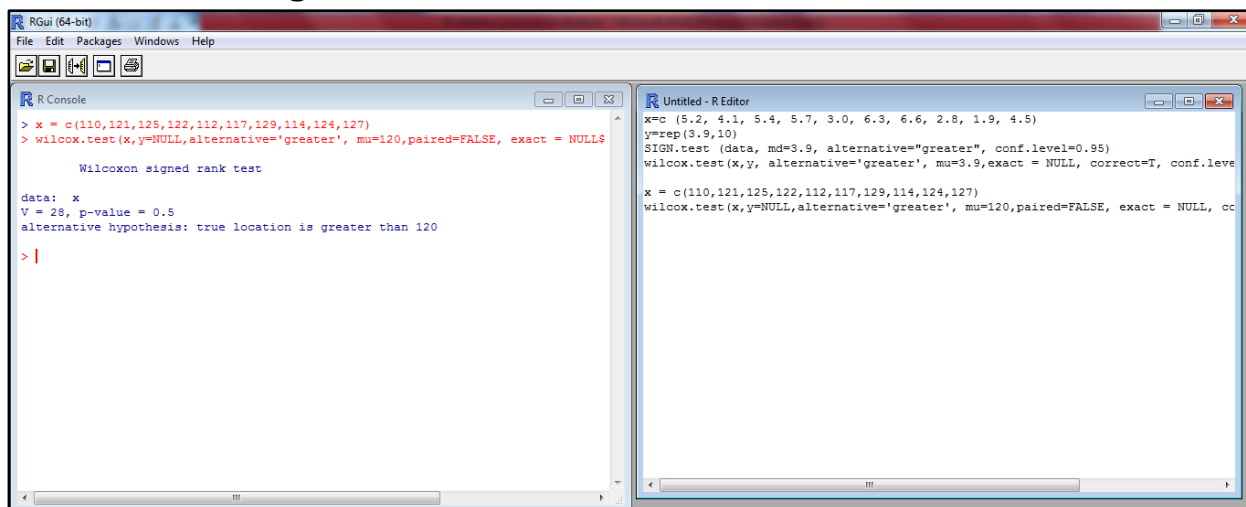
Reject H_0 if $T^- \leq d$ where d is s.t. $P(\text{type I error}) \leq 0.05$

From table A-3 (Daniel) $d = 11$ for $n=10$ and $\alpha=0.05$ (one sided)

Since $27 \nless 11$ do not reject H_0

So median pulse rates of infants is 120 beats per min.

Solution using R-Programming: R-Windows showing the execution:



R-Console output:

```
> x = c(110,121,125,122,112,117,129,114,124,127)
> wilcox.test(x, y=NULL, alternative='greater', mu=120, paired=FALSE, exact = NULL, correct =T, conf.level=0.95)
      Wilcoxon signed rank test
data:  x
V = 28, p-value = 0.5
alternative hypothesis: true location is greater than 120.
```

Interpretation: Since $p\text{-value} = 0.1719 > 0.05$ indicates one should not reject null hypothesis.

17.3 Two sample tests

17.3.1 Independent sample comparison:

Mann-Whitney U test

The two-sample Mann-Whitney U test compares values for two groups. A significant result suggests that the values for the two groups are different. It is equivalent to a two-sample Wilcoxon rank-sum test.

The test is performed with the *wilcox.test* function.

If the distributions of values of each group are similar in shape, but have outliers, then Mood's median test is an appropriate alternative.

Appropriate data

- Two-sample data. That is, one-way data with two groups only.
- Dependent variable is ordinal, interval, or ratio.
- Independent variable is a factor with two levels. That is, two groups.

- Observations between groups are independent. That is, not paired or repeated measures data.
- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread; outliers affect the spread. Otherwise the test is a test of distributions.

Example 4:

Following data represents failure times of certain type of light bulbs produced by two different manufacturers X and Y by testing 10 bulbs selected at random from each of the output. The data are (hundreds of hours used before failures)

X	5.6	4.6	6.8	4.9	6.1	5.3	4.5	5.8	5.4	4.7
Y	7.2	8.1	5.1	7.3	6.9	7.8	5.9	6.7	6.5	7.1

Use Mann-Whitney-Wilcoxon test at 5% l.o.s. to test $H_0 : M_x = M_y$ against $H_1: M_x < M_y$ (use normal approximation.)

Manual Solution:

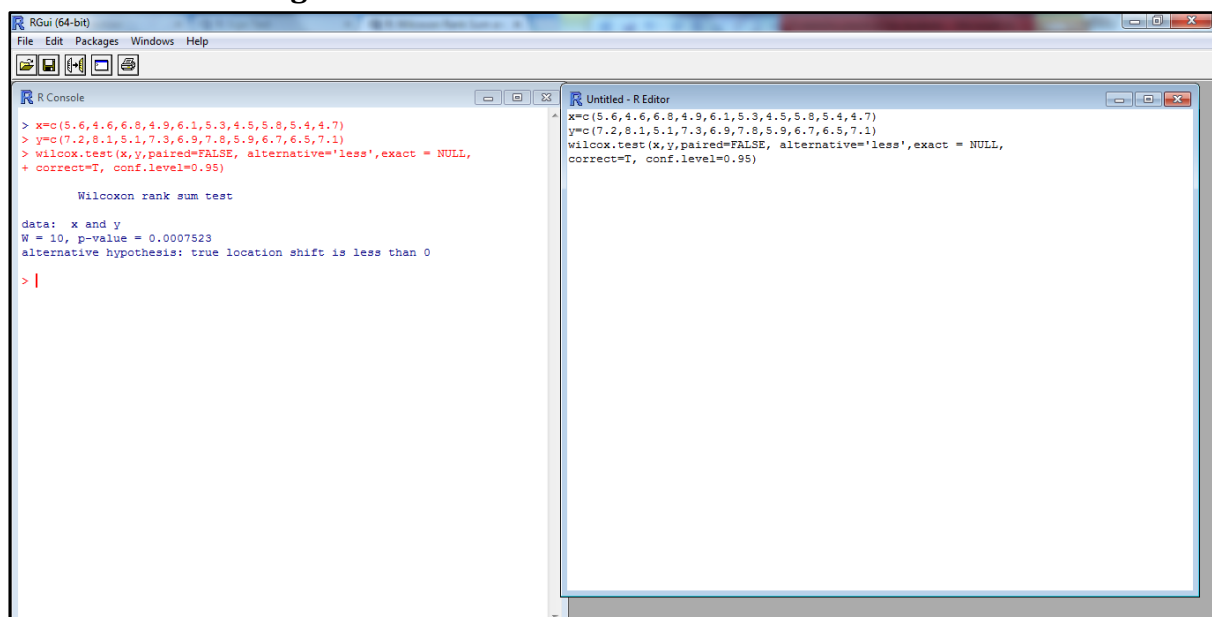
$H_0 : M_x = M_y$; $H_1: M_x < M_y$, $\alpha = 0.05$

m: no. of X observations=10, n : no.of Y observations=10. Arrange 20 observations in increasing order .Below each observation write X or Y depending on whether it comes from set I or set II. Give ranks to 20 observations.

S_x =Sum of ranks of X observations =65

$U = S_x - \frac{m(m+1)}{2} = 10$; $E(U) = \frac{mn}{2} = 50$ & $V(U) = \frac{mn(m+n+1)}{12} = 175$ Under H_0 $U \sim N(50, 175)$

$Z_{cal} = \frac{U - E(U) + 0.5}{\sqrt{V(U)}} = -2.9859 < -1.645 (= -Z_{\alpha})$ So reject H_0

Solution using R-Programming:**R-Windows showing the execution:**

R-Console output:

```
> x=c(5.6,4.6,6.8,4.9,6.1,5.3,4.5,5.8,5.4,4.7)
> y=c(7.2,8.1,5.1,7.3,6.9,7.8,5.9,6.7,6.5,7.1)
> wilcox.test(x,y,paired=FALSE, alternative='less',exact = NULL, correct=T,
conf.level=0.95)
      Wilcoxon rank sum test
data:  x and y
W = 10, p-value = 0.0007523
alternative hypothesis: true location shift is less than 0
```

Interpretation: Since $p\text{-value} = 0.0007523 < 0.05$ indicates one should reject null hypothesis.

17.3.2 Paired Sample comparison:

The two-sample rank-sum test for paired data is used to compare values for two groups where each observation in one group is paired with one observation in the other group. The distribution of differences in the paired samples should be symmetric in shape. The test is performed with the *wilcox.test* function with the *paired=TRUE* option.

Appropriate data

- Two-sample paired data. That is, one-way data with two groups only, where the observations are paired between groups.
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with two levels. That is, two groups
- The distribution of differences in paired samples is symmetric

Example 5:

Test scores of a group of 15 high – school students before & after a training programme are as given below :

Score before	63	75	78	84	58	58	70	76	74	88	74	94	99	79	93
Score after	84	86	75	94	50	95	97	98	72	100	101	98	105	84	90

Use appropriate statistical test at 1%l.o.s to check if the training has any effect on the test scores.

Hypothesis:

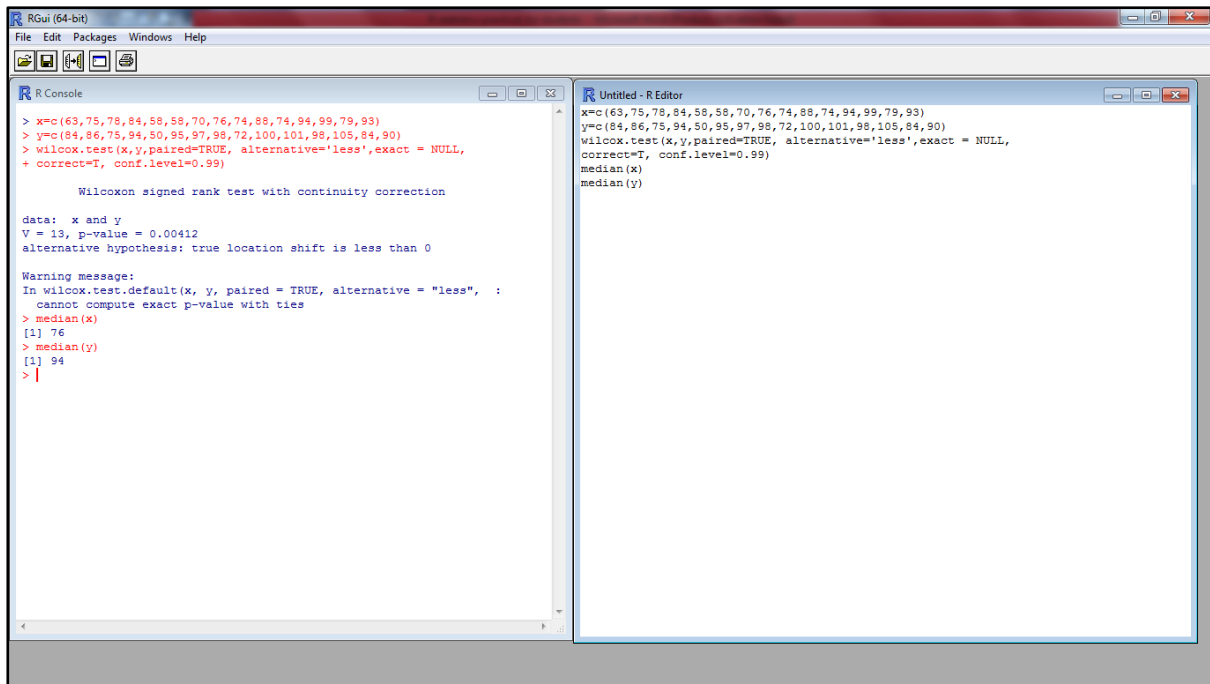
$H_0 : M_x = M_y$

$H_1 : M_x < M_y, \alpha = 0.01$

X: score before training (Median of corresponding distribution is M_x)

Y: score after training (Median of corresponding distribution is M_y)

Solution using R-Programming: R-Windows showing the execution:



R-Console output:

```

> x=c(63,75,78,84,58,58,70,76,74,88,74,94,99,79,93)
> y=c(84,86,75,94,50,95,97,98,72,100,101,98,105,84,90)
> wilcox.test(x,y,paired=TRUE, alternative='less',exact = NULL, correct=T,
conf.level=0.99)

Wilcoxon signed rank test with continuity correction
data: x and y
V = 13, p-value = 0.00412
alternative hypothesis: true location shift is less than 0

Warning message:
In wilcox.test.default(x, y, paired = TRUE, alternative = "less", :
cannot compute exact p-value with ties

```

Interpretation: Since $p\text{-value} = 0.0007523 < 0.05$ indicates one should reject null hypothesis.

Interpretation: Since $p\text{-value} = 0.00412 < 0.01$ indicates one should reject null hypothesis.

17.4 More than two sample tests

17.4.1 Three independent samples comparison:

The Kruskal-Wallis test is a rank-based test that is similar to the Mann-Whitney U test, but can be applied to one-way data with more than two groups.

The test is performed with the *kruskal.test* function.

Post-hoc tests

The outcome of the Kruskal–Wallis test tells you if there are differences among the groups, but doesn't tell you *which* groups are different from other groups. To determine which groups are different from others, post-hoc testing can be conducted. Probably the most common post-hoc test for the Kruskal–Wallis test is the Dunn test, here conducted with the *dunnTest* function in the *FSA* package. An alternative to this is to conduct Mann–Whitney tests on each pair of groups. This is accomplished with *pairwise.wilcox.test* function.

Appropriate data

- One-way data.
- Dependent variable is ordinal, interval, or ratio.
- Independent variable is a factor with two or more levels. That is, two or more groups.
- Observations between groups are independent. That is, not paired or repeated measures data.
- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread. Otherwise the test is a test of distributions.

Kruskal-Wallis Rank Sum Test

Description

Performs a Kruskal-Wallis rank sum test.

Usage

```
kruskal.test(x, ...)
```

```
## Default S3 method:
kruskal.test(x, g, ...)
## S3 method for class 'formula'
kruskal.test(formula, data, subset, na.action, ...)
```

Arguments

X	a numeric vector of data values, or a list of numeric data vectors. Non-numeric elements of a list will be coerced, with a warning.
G	a vector or factor object giving the group for the corresponding elements of x. Ignored with a warning if x is a list.
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data	an optional matrix or data frame (or similar: see <i>model.frame</i>) containing the variables in the formula formula. By default the variables are taken from <i>environment(formula)</i> .
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to <i>getOption("na.action")</i> .
...	further arguments to be passed to or from methods.

Details

`kruskal.test` performs a Kruskal-Wallis rank sum test of the null that the location parameters of the distribution of `x` are the same in each group (sample). The alternative is that they differ in at least one.

If `x` is a list, its elements are taken as the samples to be compared, and hence have to be numeric data vectors. In this case, `g` is ignored, and one can simply use `kruskal.test(x)` to perform the test. If the samples are not yet contained in a list, use `kruskal.test(list(x, ...))`.

Otherwise, `x` must be a numeric data vector, and `g` must be a vector or factor object of the same length as `x` giving the group for the corresponding elements of `x`.

Value

A list with class "htest" containing the following components:

statistic	the Kruskal-Wallis rank sum statistic.
-----------	--

Example 6:

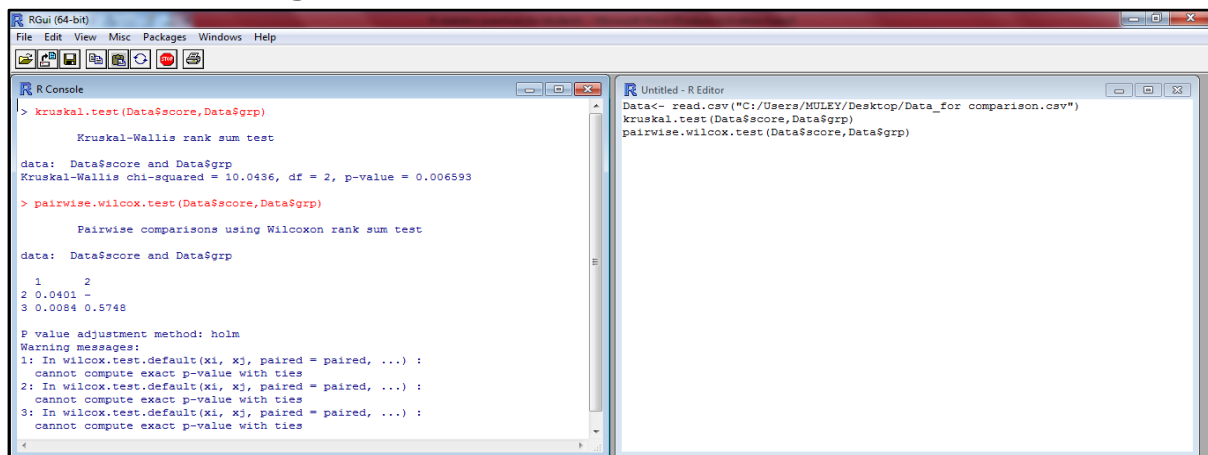
Test if there exists a significance of difference between the scores of three groups when compared against each other for the following given data set. Use 5% l. o. s. Also use post-hoc test to find the exact significance.

Group 1	Group 2	Group 3
63	84	74
75	86	76
78	75	65
84	94	84
58	50	50
58	95	85
70	97	97
76	98	88
74	72	72
88	100	90
74	101	101
94	98	98
99	105	115
79	84	94
93	90	90

Enter the data in the following format:

Group (g)	Score (x)	Group (g)	Score (x)	Group (g)	Score (x)
1	63	2	84	3	74
1	75	2	86	3	76
1	78	2	75	3	100
1	84	2	94	3	84
1	58	2	50	3	110
1	58	2	95	3	85
1	70	2	97	3	97
1	76	2	98	3	88
1	74	2	72	3	95
1	88	2	100	3	90
1	74	2	101	3	105
1	94	2	98	3	98
1	99	2	105	3	115
1	79	2	84	3	94
1	93	2	90	3	90

R-Windows showing the execution:



R-Console output:

```
kruskal.test (Data$score, Data$grp)
      Kruskal-Wallis rank sum test
data:  Data$score and Data$grp
Kruskal-Wallis chi-squared = 10.0436, df = 2, p-value = 0.006593
```

Interpretation: Since $p\text{-value} = 0.006593 < 0.01$ indicates one should reject null hypothesis and conclude that there exists significance of difference between the scores of three group at 1% l.o.s. To find exact significance of difference we used post-hoc test comparison.

R-Console output:

```
> pairwise.wilcox.test(Data$score, Data$grp)
      Pairwise comparisons using Wilcoxon rank sum test
data:  Data$score and Data$grp
```

	1	2
2	0.0401	-
3	0.0084	0.5748

Interpretation:

- p-value for the comparison of Group 1 against Group 2 is less than that of 0.05 indicates significance of difference.
 - p-value for the comparison of Group 1 against Group 3 is less than that of 0.05 indicates significance of difference.
 - p-value for the comparison of Group 3 against Group 2 is greater than that of 0.05 indicates no significance of difference.
-

Chapter 18

Multiple Regression: A Case Study

Gourav Tiwari, Mohammed Saad Qadri, Misbahuddin Saba, Dr. Asha Jindal*

Researcher, Star DBT Scheme, Dept. of Statistics, K.C. College, Mumbai - 20, INDIA.

*Star - DBT Mentor & Associate Professor and Head, Dept. of Statistics, K.C. College, Mumbai, INDIA – 20.

18.1 Data Description

The data consists of 398 records characterizing various car types. For each car type the following attributes are provided: the MPG value (**mpg**) measured for each car model in a test performed in 1982, the number of the engine cylinders (**cyl**), the cylinder displacement in cubic inches (**displ**), the engine power (**power**), the car weight in pounds (**weight**), a number of seconds required to accelerate to the speed of 100 miles per hour (**accel**), the car's production year (**year**), the country of production (**origin**: USA, Europe, or Japan), and the name of the model (**model**).

Objective: - To Identify which variables are influencing the miles per gallon (MPG i.e fuel consumption) of a car.

18.1.1 Data Dictionary

NAME OF THE VARIABLE	VARIABLE DESCRIPTION
MPG	Mileage per gallon
Cylinders	No. of cylinders in the car
Displacement	Displacement of the piston in cubic inches
Horsepower	Engine power
Acceleration	A number of seconds required to accelerate to the speed of 100 miles per hour
Weight	Car's weight in pounds
Year	Year of production of car
Origin	Place of production (for e.g. 1-North America, 2-Europe, 3-Asia)
Name	Name of the car

18.1.2 Data used for Analysis

```
> head(data)
  x  MPG cylinders displacement horsepower weight acceleration year origin
1 1  18         8         307         130   3504         12.0   70    1
2 2  15         8         350         165   3693         11.5   70    1
3 3  18         8         318         150   3436         11.0   70    1
4 4  16         8         304         150   3433         12.0   70    1
5 5  17         8         302         140   3449         10.5   70    1
6 6  15         8         429         198   4341         10.0   70    1
      Name
1 chevrolet chevelle malibu
2 buick skylark 320
3 plymouth satellite
4 amc rebel sst
5 ford torino
6 ford galaxie 500
```

18.1.3 Summary of the Data

```
> summary(data[2:9])
      MPG      cylinders      displacement      horsepower      weight
Min.   : 9.00   Min.   :3.000   Min.   : 68.0   150   : 22   Min.   :1613
1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   90    : 20   1st Qu.:2224
Median :23.00   Median :4.000   Median :148.5   88    : 19   Median :2804
Mean   :23.51   Mean   :5.455   Mean   :193.4   110   : 18   Mean   :2970
3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   100   : 17   3rd Qu.:3608
Max.   :46.60   Max.   :8.000   Max.   :455.0   75    : 14   Max.   :5140
      acceleration      year      origin
Min.   : 8.00   Min.   :70.00   Min.   :1.000
1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000
Median :15.50   Median :76.00   Median :1.000
Mean   :15.57   Mean   :76.01   Mean   :1.573
3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
Max.   :24.80   Max.   :82.00   Max.   :3.000
      (other):288
```

Summary gives us the minimum value, 1st quartile, median, mean, 3rd quartile and maximum value of each variable in the data.

18.1.4 Struture of the Data

```
> str(data)
'data.frame':   398 obs. of  10 variables:
 $ x          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ MPG        : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : int   8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : Factor w/ 94 levels "?","100","102",...: 17 35 29 29 24 42 47 46 48 40 ...
 $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ year        : int  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Name        : Factor w/ 305 levels "amc ambassador brougham",...: 50 37 232 15 162 142 55 224 242 2 ...
```

From the structure of the data we can observe that the data consist of 398 observations and 9 variables. Also it tells us the type of the variable where MPG, cylinders, displacement, weight, acceleration, year and origin are numeric variables and horsepower and name are categorical variables. Also we can see that data for the variable horsepower consist of some missing values denoted by "?", so before starting our analysis we have to convert the variable type of horsepower from factor to integer and impute the missing values.

Conversion of the Variable Horsepower from Factor to Integer

Conversion of the variable type of horsepower from factor to integer can be done in the following way

```
data$horsepower<-as.numeric(as.character(data$horsepower))
class(data$horsepower)
"numeric"
```

18.2 Imputation of the Missing Values

The missing values can be imputed by the mean of the other known values. In the variable horsepower there are 6 missing values.

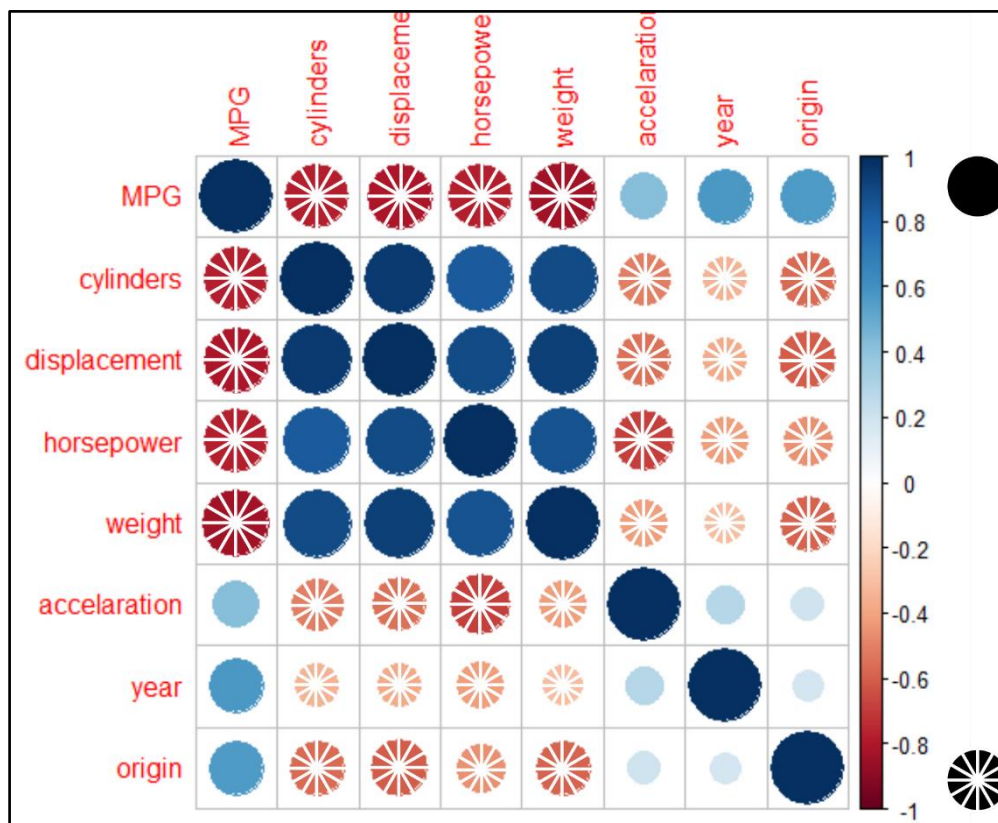
Mean of known values=104.423 which can be approximately taken as 104. Hence replacing all the missing values (denoted by '?') by 104 in the data.

18.3 Bivariate Analysis and Visualization

```
> correlation<- cor(data[,2:9])
> correlation
```

	MPG	cylinders	displacement	horsepower	weight	acceleration	year	origin
MPG	1.0000000	-0.7753963	-0.8042028	-0.7715428	-0.8317409	0.4202889	0.5792671	0.5634504
cylinders	-0.7753963	1.0000000	0.9507214	0.8390609	0.8960168	-0.5054195	-0.3487458	-0.5625433
displacement	-0.8042028	0.9507214	1.0000000	0.8937600	0.9328241	-0.5436841	-0.3701642	-0.6094094
horsepower	-0.7715428	0.8390609	0.8937600	1.0000000	0.8606759	-0.6843761	-0.4117505	-0.4536133
weight	-0.8317409	0.8960168	0.9328241	0.8606759	1.0000000	-0.4174573	-0.3065643	-0.5810239
acceleration	0.4202889	-0.5054195	-0.5436841	-0.6843761	-0.4174573	1.0000000	0.2881370	0.2058730
year	0.5792671	-0.3487458	-0.3701642	-0.4117505	-0.3065643	0.2881370	1.0000000	0.1806622
origin	0.5634504	-0.5625433	-0.6094094	-0.4536133	-0.5810239	0.2058730	0.1806622	1.0000000

```
> corrrplot(correlation)
> data$origin<-as.factor(data$origin)
```



Above schedule and graph show the correlation between the variables. We can observe that cylinders, displacement, horsepower and weight are highly negatively correlated with MPG whereas acceleration and year of production are quite positively correlated. Also we can see that cylinders, displacement, weight and horsepower are positively correlated whereas cylinders & acceleration are positively correlated which means that as we increase the num

ber of cylinders , displacement of the piston, weight of the car, engine power also increases but results in the decrease in the acceleration of the car.

Visualizing the effects of no. of cylinders on MPG of car.

```
> p<-ggplot(data, aes(x=cylinders, y=MPG, fill=cylinders)) + geom_boxplot()
> p
```

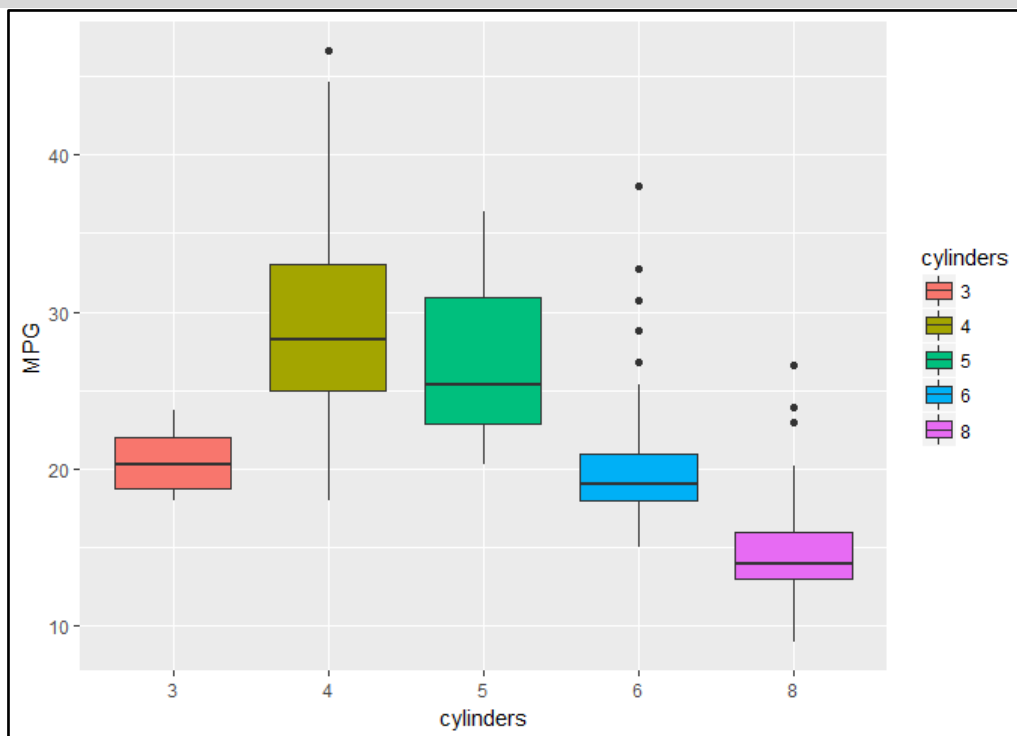


Figure 1

Correlation coefficient between MPG and no. of cylinders is -0.77539 which implies there is negative correlation between the variables which we can see from the above boxplot. It also shows that for 4 cylinders the cars provides the maximum MPG.

Now lets see the change in the MPG of the cars over the years,

```
> tapply(data$MPG,data$year,mean)
 70    71    72    73    74    75    76    77    78    79    80    81    82
17.68966 21.25000 18.71429 17.10000 22.70370 20.26667 21.57353 23.37500 24.06111 25.09310 33.69655 30.33448 31.70968
```

Table 1

Here we can see that over the years there has been increase in the average MPG of the car. This increase in the MPG of the cars can be the effect of positive changes in other variables.

a) Summarization of horsepower over the years:

```
> tapply(data$horsepower,data$year,mean)
 70    71    72    73    74    75    76    77    78    79    80    81    82
147.82759 106.92857 120.17857 130.47500 94.59259 101.06667 101.11765 105.07143 99.69444 101.20690 79.31034 81.82759 82.19355
```

Table 2

From the correlation matrix we can see that horsepower and MPG are highly negatively correlated. And over the years there has been production of the cars with less horsepower which might be a possible reason for the increment in the MPG of the cars.

b) Summarization of acceleration over years:

```
> tapply(data$acceleration,data$year,mean)
 70    71    72    73    74    75    76    77    78    79    80    81    82
12.94828 15.14286 15.12500 14.31250 16.20370 16.05000 15.94118 15.43571 15.80556 15.81379 16.93448 16.30690 16.63871
```

Table 3

From correlation matrix we can see that there is slight positive correlation between acceleration and MPG of the cars. And over the years there has been production of the cars which provides better acceleration and hence contributing to the rise in the average MPG of the cars.

c) Summarization of weight of the car over the years:

```
> tapply(data$weight,data$year,mean)
 70    71    72    73    74    75    76    77    78    79    80    81    82
3372.793 2995.429 3237.714 3419.025 2877.926 3176.800 3078.735 2997.357 2861.806 3055.345 2436.655 2522.931 2453.548
```

Table 4

From the correlation matrix we can see that there is negative correlation between weight and MPG of the cars. And over the years there has been production of the cars with lesser weight and hence contributing to the rise in the MPG of the cars.

18.4 Predictive Modelling

Multiple Regression

It is a statistical tool that allows you to examine how multiple independent variables are related to a dependent variable. Once we have identified how these multiple variables relate to your dependent variable, we can take information about all of the independent variables and use it to make much more powerful and accurate predictions about why things are the way they are. This latter process is called “Multiple Regression”.

A population model for a multiple linear regression model that relates a y -variable to $p - 1$ x -variables is written as

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_{p-1} x_{i(p-1)} + \epsilon_i.$$

Dependent Variable for Modelling

In our given data MPG (Mileage per gallon) is the dependent variable i.e. $Y = \text{MPG}$

Independent Variable

In our data independent variables are

1. Cylinders
2. Displacement
3. Weight
4. Horsepower
5. Acceleration
6. Year of production
7. Origin

Hypothesis

We test,

$H_0: \beta_1 = \beta_2 = \beta_3 = \dots = \beta_p = 0$ i.e. there is no linear regression between our dependent and independent variable

VS

$H_1: \beta_i \neq 0, i=1,2,3\dots p$, i.e. There is a linear regression between our dependent and independent variable

Model Building

Multiple linear regression model between our dependent and independent variables is

$$Y(\text{MPG}) = \beta_0 + \beta_1(\text{cylinders}) + \beta_2(\text{weight}) + \beta_3(\text{displacement}) + \beta_4(\text{horsepower}) + \beta_5(\text{acceleration}) + \beta_6(\text{year}) + \beta_7(\text{origin})$$

Summary of the model

```
call:
lm(formula = MPG ~ cylinders + weight + displacement + horsepower +
    acceleration + year + origin, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-8.9328 -2.0492 -0.0907  1.9473 13.3512

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.887e+01  4.586e+00  -4.115 4.73e-05 ***
cylinders    -4.209e-01  3.192e-01  -1.319 0.18809
weight      -6.965e-03  6.417e-04 -10.854 < 2e-16 ***
displacement  2.362e-02  7.623e-03   3.099 0.00208 **
horsepower   -1.340e-02  1.309e-02  -1.024 0.30640
acceleration  9.937e-02  9.536e-02   1.042 0.29806
year         7.842e-01  5.093e-02  15.396 < 2e-16 ***
origin2       2.783e+00  5.568e-01   4.998 8.77e-07 ***
origin3       2.828e+00  5.454e-01   5.186 3.46e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.308 on 389 degrees of freedom
Multiple R-squared:  0.8245,    Adjusted R-squared:  0.8209
F-statistic: 228.4 on 8 and 389 DF,  p-value: < 2.2e-16
```

18.4.1 ASSUMPTIONS VALIDATION**a) Multicollinearity**

Multicollinearity exists when two or more of the predictors(independent variables) in a regression model are moderately or highly correlated. Unfortunately, when it exists, it can wreak havoc on our analysis and thereby limit the research conclusions we can draw.

- Multicollinearity Detection**

Multicollinearity may be checked multiple ways:

- Correlation matrix – When computing a matrix of Pearson's bivariate correlations among all independent variables, the magnitude of the correlation coefficients should be less than .80.

- Variance Inflation Factor (VIF) – The VIFs of the linear regression indicate the degree that the variances in the regression estimates are increased due to multicollinearity. VIF values higher than 5 indicate that multicollinearity is a problem. There is also one more term called GVIF (Generalized Variance Inflation Factor) which comes into the play for factors and polynomial variables. Variables which require more than 1 coefficient and thus more than 1 degree of freedom are typically evaluated using the GVIF. For one-coefficient terms VIF equals GVIF.
- **Remedies for Multicollinearity**
 - Drop one of the independent variable which is explained by others
 - Use Principal Component Regression in case of severe multicollinearity
 - Use Ridge Regression

In our data we check the multicollinearity by using GVIF. Consider the model,

$$\text{MPG}(Y) = \beta_0 + \beta_1 * (\text{cylinders}) + \beta_2 * (\text{displacement}) + \beta_3 * (\text{horsepower}) + \beta_4 * (\text{weight}) + \beta_5 * (\text{acceleration}) + \beta_6 * (\text{year}) + \beta_7 * (\text{origin})$$

```
> library(car)
> reg1<-lm(MPG~cylinders+weight+displacement+horsepower+acceleration+year+origin,data)
> vif(reg1)
```

	GVIF	Df	GVIF^(1/(2*Df))
cylinders	10.696136	1	3.270495
weight	10.712126	1	3.272938
displacement	22.920730	1	4.787560
horsepower	9.065368	1	3.010875
acceleration	2.508941	1	1.583964
year	1.286670	1	1.134315
origin	2.035532	2	1.194454

Since the GVIF of the displacement is highest and exceeds 5 we will drop this variable and rebuild the model excluding Displacement

$$\text{MPG}(Y) = \beta_0 + \beta_1 * (\text{cylinders}) + \beta_2 * (\text{horsepower}) + \beta_3 * (\text{weight}) + \beta_4 * (\text{acceleration}) + \beta_5 * (\text{year}) + \beta_6 * (\text{origin})$$

```
> reg2<-lm(MPG~cylinders+weight+horsepower+acceleration+year+origin,data)
> vif(reg2)
```

	GVIF	Df	GVIF^(1/(2*Df))
cylinders	6.154188	1	2.480764
weight	8.774478	1	2.962175
horsepower	8.315921	1	2.883734
acceleration	2.475220	1	1.573283
year	1.270440	1	1.127138
origin	1.751514	2	1.150412

Since the GVIF of the weight is highest and exceeds 5 we will drop this variable and rebuild the model excluding weight

$$\text{MPG}(Y) = \beta_0 + \beta_1 * (\text{cylinders}) + \beta_2 * (\text{horsepower}) + \beta_3 * (\text{acceleration}) + \beta_4 * (\text{year}) + \beta_5 * (\text{origin})$$

```
> reg3<-lm(MPG~cylinders+horsepower+acceleration+year+origin,data)
> vif(reg3)
```

	GVIF	Df	GVIF^(1/(2*Df))
cylinders	4.156914	1	2.038851
horsepower	5.068871	1	2.251415
acceleration	1.970180	1	1.403631
year	1.238933	1	1.113074
origin	1.649847	2	1.133342

Since the GVIF of horsepower is highest and exceeds 5 we will drop this variable and rebuild the model excluding horsepower

$$\text{MPG}(Y) = \beta_0 + \beta_1(\text{cylinders}) + \beta_2(\text{acceleration}) + \beta_3(\text{year}) + \beta_4(\text{origin})$$

```
> reg4<-lm(MPG ~ cylinders+acceleration+year+origin,data)
> vif(reg4)
```

	GVIF	Df	GVIF^(1/(2*Df))
cylinders	2.128088	1	1.458797
acceleration	1.391214	1	1.179497
year	1.197183	1	1.094159
origin	1.647669	2	1.132968

Here we can see that the GVIF of the remaining variables are smaller than 5. Hence we say that the model created using these variables is the best model and the problem of multicollinearity is resolved.

Diagnostic Plots

```
> par(mfrow=c(2,2))
> plot(reg4)
```

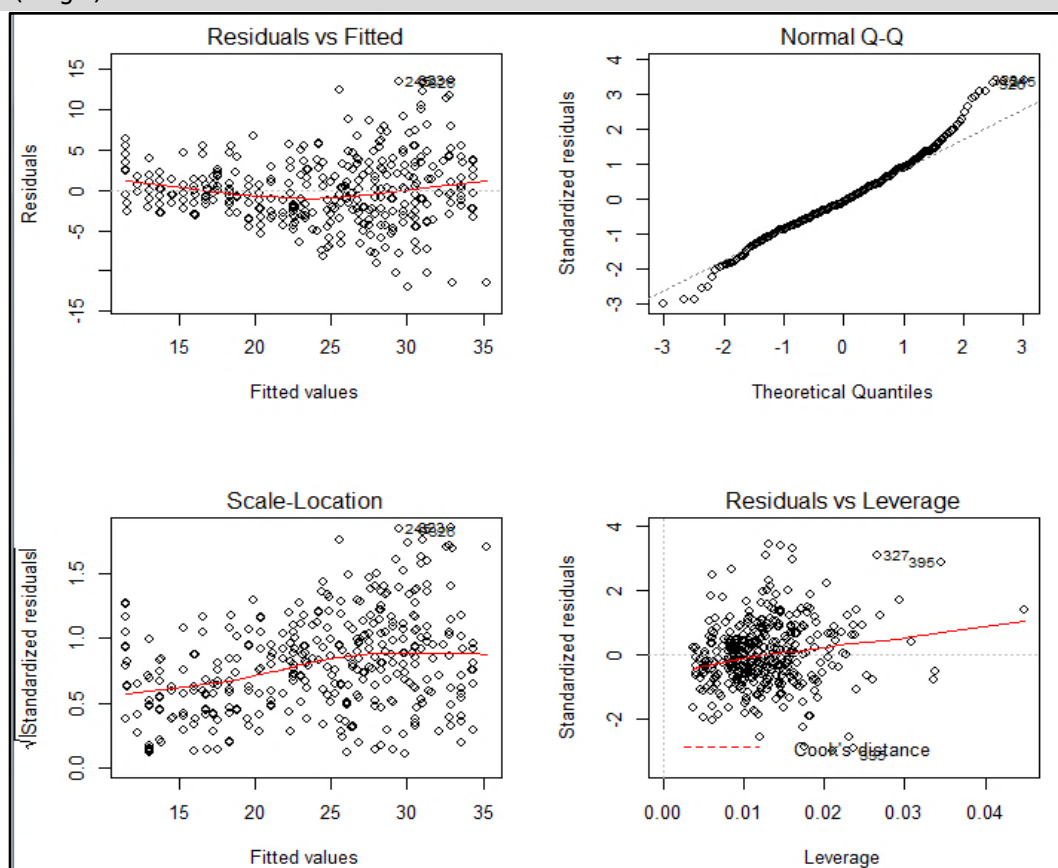


Figure 2

b) Homoscedasticity

The assumption of homoscedasticity (meaning “same variance”) is central to linear regression models. Homoscedasticity describes a situation in which the error term (that is, the “noise” or random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables

- **Detection of homoscedasticity**

Homoscedasticity can be tested statistically by **Breush-Pagan test**.

Consider the hypothesis,

H_0 : The variance of the residuals is constant.

v/s

H_1 : The variance of the residuals is not constant.

Breush Pagan Test,

```
> library(lmtest)
> lmtest::bptest(reg4)

studentized Breusch-Pagan test

data: reg4
BP = 46.909, df = 5, p-value = 5.928e-09
```

This test have a p-value= 5.928×10^{-9} less than a significance level of 0.05, therefore we reject the null hypothesis that the variance of the residuals is constant and infer that the assumption of homoscedasticity is not satisfied.

Alternatively, we can also detect homoscedasticity using the diagnostic plots. The plots we are interested in are at the top-left and bottom-left. The top-left is the chart of residuals vs fitted values, while in the bottom-left one, it is standardised residuals on Y axis. If there is exist homoscedasticity, you should see a completely random, equal distribution of points throughout the range of X axis and a flat red line. But in our case, as you can notice from the top-left plot, the red line is slightly curved and the residuals seem to increase as the fitted Y values increase. So, the inference here is, **homoscedasticity does not exists**.

- **Remedial Measures**

The heteroscedasticity (absence of homoscedasticity) can be fixed by transforming the variables. Log transformation is one of the transformations that can be used. In our case we take log of our dependent variable i.e. log(MPG) and rebuild the model

$$Y[\log(\text{MPG})] = \beta_0 + \beta_1 * (\text{cylinders}) + \beta_2 * (\text{acceleration}) + \beta_3 * (\text{year}) + \beta_4 * (\text{origin})$$

Summary

```
Call:
lm(formula = log(MPG) ~ cylinders + acceleration + year + origin,
    data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.50355 -0.09129 -0.00571  0.09953  0.45767
```

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.3940805  0.1946174   7.163 3.93e-12 ***
cylinders    -0.1234605  0.0067193  -18.374 < 2e-16 ***
acceleration  0.0002592  0.0033511   0.077 0.938385
year         0.0307584  0.0023184  13.267 < 2e-16 ***
origin2      0.0834540  0.0246402   3.387 0.000778 ***
origin3      0.1213613  0.0238270   5.093 5.47e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1561 on 392 degrees of freedom
Multiple R-squared:  0.7914,    Adjusted R-squared:  0.7888
F-statistic: 297.5 on 5 and 392 DF,  p-value: < 2.2e-16

```

Looking at the above summary,

Coefficient of variable cylinder (β_1) = -0.123461, this implies that if we increase the number of cylinders by 1 unit the MPG of the car decreases 12.35% by units.

Coefficient of acceleration (β_2) = 0.0002592, this implies that if we increase the acceleration by 1 unit the MPG of the car also increases 0.02592% by units.

Coefficient of year (β_3) = 0.0307584, this implies that if we increase the year by 1 unit the MPG of the car also increases by 3.07584 units. Which is quite obvious along with time there have been advancements in technologies which helped in better production of cars in terms of MPG.

Adjusted R^2 = 0.7888 = 78.88% of the variance in the values of log(MPG) is explained by the model

VIF of the new model

```

> reg5<-lm(log(MPG)~cylinders+acceleration+year+origin,data)
> vif(reg5)
            GVIF Df GVIF^(1/(2*Df))
cylinders    2.128088 1      1.458797
acceleration 1.391214 1      1.179497
year         1.197183 1      1.094159
origin       1.647669 2      1.132968

```

Since the GVIF of all the variables are less than 5 we can conclude that multicollinearity does not exist.

Now let's check for homoscedasticity using Breush-Pagan test.

```

> lmtest::bptest(reg5)

studentized Breusch-Pagan test

data:  reg5
BP = 10.661, df = 5, p-value = 0.05853

```

Since p-value is greater than 0.05 we accept H_0 and conclude that the residual terms have same variance. Also we look at the diagnostic plots of this model to check the homoscedasticity

Diagnostic Plots for testing Homoscedasticity and Multicollinearity

```
> par(mfrow=c(2,2))
> plot(reg5)
```

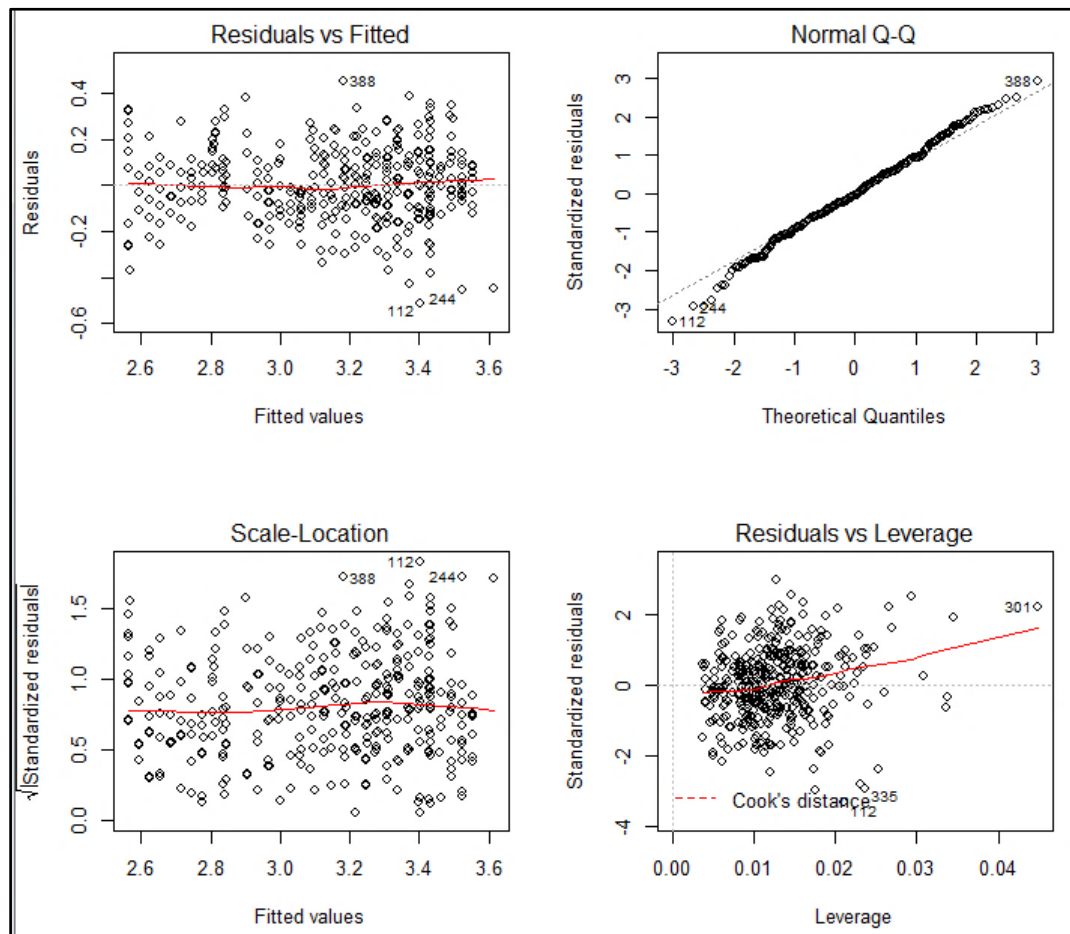


Figure 3

c) Normality of residuals

The residual terms are assumed to be normally distributed. This assumption can be checked by plotting Normal Q-Q graph. In Figure 2 Normal Q-Q plot is shown where the values of the residual are along the diagonal line which means that they are normally distributed. Hence this assumption is also satisfied.

d) Linearity between response and predictors

This assumption says that the response variable (dependent variable) and predictors (independent variables) needs to have linear relationship. This assumption can be checked by plotting fitted values VS residual graph. In figure 2 the red line is almost straight and hence we say that the variables are fairly linearly correlated.

18.5 Conclusions

1. From the summary of the final model we can see that the cylinders, year of production and origin (place of production) are the significantly influencing MPG of the car.
2. From bivariate analysis we can see that over the years there has been increment in MPG of the cars which is the collective effect of,
 - Decrement in horsepower .
 - Decrement in the weight of the car.
 - Rise in the acceleration.

18.6 Recommendations

1. From the figure 1 we can say that 4 is the optimum number of cylinders leading to enhancement in MPG of the cars.
 2. Minimizing the weight and horsepower of the car can effectively increase the MPG of the car.
-

Chapter 19

Logistics Regression: Human Resources Analytics Why do our employees leave prematurely?

Pravesh .S. Tiwari¹, Divya .M. Poojari²

¹Data Analyst in Accenture; ²Statistical Programmer in Cognizant

19.1 Introduction

Employee attrition is one of the biggest challenges that the companies face.

There are several factors that lead to attrition. While it may not be easy to control all the factors, it may not be worthwhile to look into those factors that seem controllable. Factors such as average number of hours spend per month by the employees, salary, promotions, job rotation, number of projects are a few which are easier to manage.

If we are able to extract cut-off levels for some of the above mentioned factors through our analysis, then we should be able to have a better understanding about the factors that are responsible for the employees leaving the company prematurely.

19.2 Role of analytics in Human Resources

In today's competitive world talented people are the most worthwhile treasure for the company and at the same time burdensome to hold down such valuable resources in organization. During last year's, large investments were put into tools and information systems to manage performance, hiring, compliance and employees' development in order to enhance its capabilities.

Using data produced by these tools and systems typically implemented into enterprise HR departments, most companies are able to provide reports at least at some basic level. They are usually able to go through data from several previous periods to assess positive or negative trends, or to create benchmarks comparing their performance against their competitors across time and regions. However, in order to bring real value and help driving the business competitiveness, HR analytics utilization needs to go far beyond.

The biggest struggles in achieving better utilization of data resources and information systems are inefficient use of the data, asking wrong questions and lack of analytical ability in HR environment in general. HR departments are in need for analytically capable people enabled to provide right insights combining reporting skills and domain knowledge.

19.2.1 Problem Statement

- The goal of the case study is to find out which are the most influential factors leading to employees renege.
- Which employee will leave next?

19.2.2 Methodology

- **Visualization:** - The first step is to visualize and perform univariate analysis to explore data to find useful insights.
- **Model:** - Next step is to model the data in order to confirm or reject our hypothesis that certain variables are significant in determining employee departures.
- **Actionable Insights:** - The final step is to review and build onto our analysis by drawing new insights or further enhancing existing insights.

19.2.3 Data Dictionary

Variable Name	Variable Definition
Satisfaction Level	Level of satisfaction (0-1)
Last evaluation	Time since last performance evaluation (in Years)
Projects	Number of projects completed while at work
Average monthly hours	Average monthly hours at workplace
Time spent at company	Number of years spent in the company
Accident	Whether the employee had a workplace accident
Promotion Last 5 yrs	Whether the employee was promoted in the last five years(1 = promoted, 0 = Not Promoted)
Positions	Type of Job Position
Salary	Salary level (1= low, 2= medium, 3= high)
Left	Whether the employee has left (0= remains employed, 1= left)

19.2.4 Structure of the data

```
'data.frame': 14999 obs. of 10 variables:
 $ satisfaction_level : num 0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
 $ last_evaluation : num 0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
 $ number_project : num 2 5 7 5 2 2 6 5 5 2 ...
 $ average_monthly_hours : num 157 262 272 223 159 153 247 259 224 142 ...
 $ time_spent_company : num 3 6 4 5 3 3 4 5 5 3 ...
 $ work_accident : Factor w/ 2 levels "Accident","No Accident": 2 2 2 2 2 2 2 2 2 2 ...
 $ left : Factor w/ 2 levels "Left","Not Left": 1 1 1 1 1 1 1 1 1 1 ...
 $ promotion_last_5years: Factor w/ 2 levels "Not Promoted",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ sales : Factor w/ 10 levels "accounting","hr",...: 8 8 8 8 8 8 8 8 8 8 ...
 $ salary : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...
```

From the structure of the data we can see that sample size of the data is 14999 and there are 10 variables and also whether a variable is categorical or continuous where **num** indicates numeric or continuous and **Factor** indicates categorical.

19.3 Data Analysis

19.3.1 Data used for Analysis

satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spent_company	Work_accident	left	promotion_last_5years	sales	salary
0.38	0.53	2	157	3	No Accident	Left	Not Promoted	sales	low
0.80	0.86	5	262	6	No Accident	Left	Not Promoted	sales	medium
0.11	0.88	7	272	4	No Accident	Left	Not Promoted	sales	medium
0.72	0.87	5	223	5	No Accident	Left	Not Promoted	sales	low
0.37	0.52	2	159	3	No Accident	Left	Not Promoted	sales	low
0.41	0.50	2	153	3	No Accident	Left	Not Promoted	sales	low
0.10	0.77	6	247	4	No Accident	Left	Not Promoted	sales	low
0.92	0.85	5	259	5	No Accident	Left	Not Promoted	sales	low
0.89	1.00	5	224	5	No Accident	Left	Not Promoted	sales	low
0.42	0.53	2	142	3	No Accident	Left	Not Promoted	sales	low

19.3.2 Univariate Analysis and Visualization

a) Correlation Matrix

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spent_company
satisfaction_level	1.00	0.11	-0.14	-0.02	-0.10
last_evaluation	0.11	1.00	0.35	0.34	0.13
number_project	-0.14	0.35	1.00	0.42	0.20
average_monthly_hours	-0.02	0.34	0.42	1.00	0.13
time_spent_company	-0.10	0.13	0.20	0.13	1.00

b) For Employees who left the organization

Average monthly time spent by employees for status of promotion in last 5 years

Not Promoted	Promoted
207.5780	177.7368

Average last evaluation score for by employees for status of promotion in last 5 years

Not Promoted	Promoted
0.7188063	0.5884211

Average number of projects done by employees for status of promotion in last 5 years

Not Promoted	Promoted
3.859797	3.052632

From the above correlation matrix we can see that last_evaluation, number_of_projects, and average_monthly_hours are quite correlated with each other. Which means number of projects and time spent on these projects influence the last evaluation score, also from the remaining three tables we can see that for employees who left numbers of projects and average monthly hours spent to do them are more for those who are not promoted than who are promoted in last 5 years and also last evaluation score is more for not promoted employees, which indicates more workload with less financial growth.

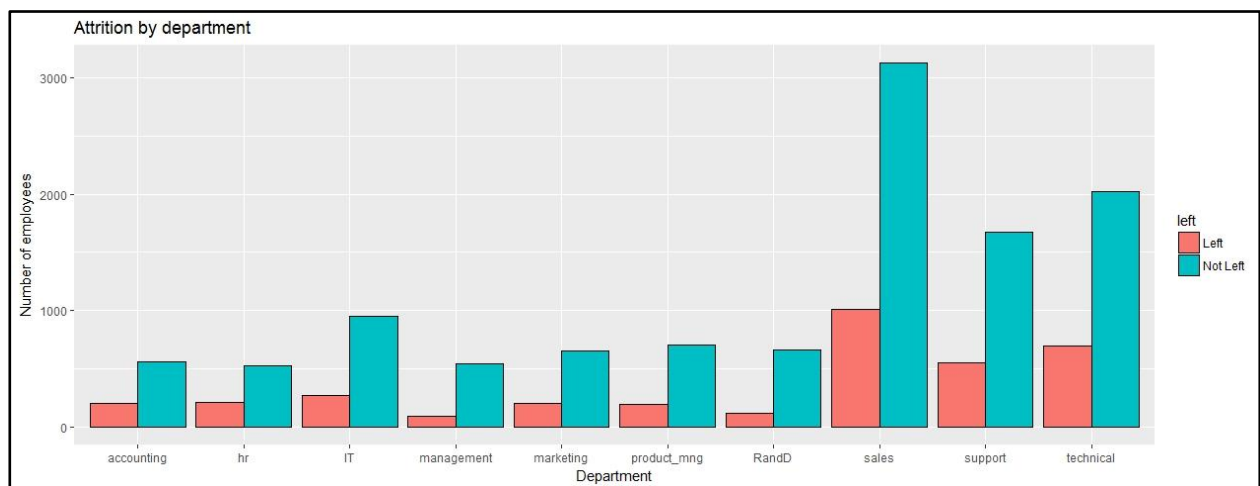


Figure 1

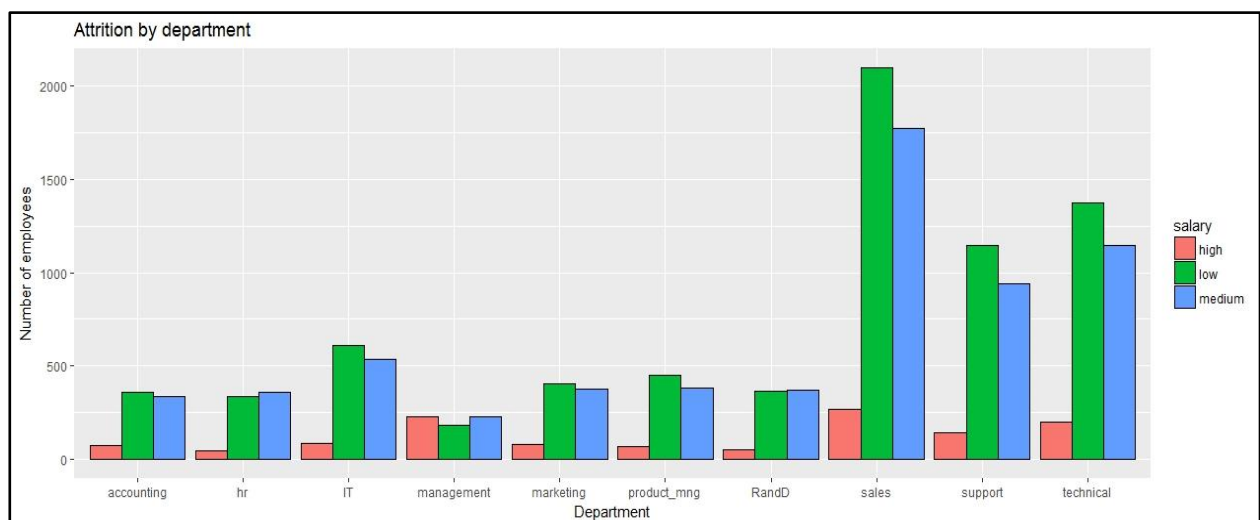


Figure 2

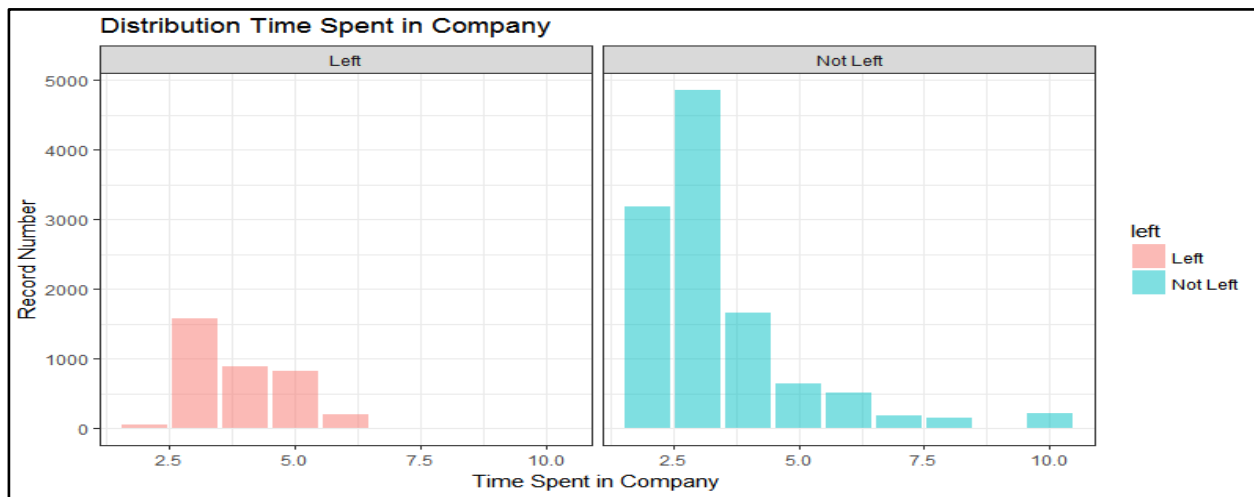


Figure 3

From the Figure 1 we can see that maximum numbers of attritions are from sales, support and technical department and Figure 2 show that sales, support and technical departments employees falls in low range salary. So we can conclude that low salary could be a possible reason for the renege for employees. From figure 3 we can observe that attrition rate is high for 2.5 to 5 years experienced people in a company, Not getting proper hike in last 5 years could be a possible reason for this.

19.4 Predictive Modelling

19.4.1 Logistic Regression

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

The logistic regression model

$$\ln[p/(1-p)] = a + BX + e \text{ or}$$

$$[p/(1-p)] = \exp(a + BX + e)$$

where:

\ln is the natural logarithm, \log_{\exp} , where $\exp=2.71828...$

p is the probability that the event Y occurs, $p(Y=1)$

$p/(1-p)$ is the "odds ratio"

$\ln[p/(1-p)]$ is the log odds ratio, or "logit"

all other components of the model are the same.

The logistic regression model is simply a non-linear transformation of the linear regression. The "logistic" distribution is an S-shaped distribution function which is similar to the standard-normal distribution (which results in a probit regression model) but easier to work

with in most applications (the probabilities are easier to calculate). The logit distribution constrains the estimated probabilities to lie between 0 and 1.

For instance, the estimated probability is:

$$p = 1/[1 + \exp(-a - BX)]$$

With this functional form:

if you let $a + BX = 0$, then $p = .50$

as $a + BX$ gets really big, p approaches 1

as $a + BX$ gets really small, p approaches 0.

19.4.2 Data splitting

Separating data into training and testing sets is an important part of evaluating data mining models. Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing. Analysis Services randomly samples the data to help ensure that the testing and training sets are similar. By using similar data for training and testing, you can minimize the effects of data discrepancies and better understand the characteristics of the model.

After a model has been processed by using the training set, you test the model by making predictions against the test set. Because the data in the testing set already contains known values for the attribute that you want to predict, it is easy to determine whether the model's guesses are correct.

19.4.2 Dependent Variable for Modeling:

Left (whether employee has left the organisation or not)

Independent Variables:

last_evaluation	number_project
average_monthly_hours	time_spend_company
Work_accident	promotion_last_5years
Sales	Salary

19.5 Model Building and Output Interpretation

Step 1: Create a logistic model using `glm()` function in R

```
glm(formula = left ~ last_evaluation + number_project + average_monthly_hours +
    time_spend_company + work_accident + promotion_last_5years +
    sales + salary, family = binomial, data = training.data)
```

Step 2: Global testing

$H_0: b_1 = b_2 = \dots = b_k = 0$ OR (H_0 : None of the variables has significant impact)

v/s

H1: At least one coefficient is not zero

Test Statistic:

$\chi^2 = L1 - L2$ which follows Chi-square distribution with k df.

$L1 = -2 \log L$ with only constant term $L2 = -2 \log L$ with k variables and constant term

Reject H_0 for large value of χ^2 or Reject H_0 if p value < 0.05

```
Model 1: left ~ 1
Model 2: left ~ last_evaluation + number_project + average_monthly_hours +
  time_spend_company + work_accident + promotion_last_5years +
  sales + salary
Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      11999      13173
2      11982      12029 17      1144 < 2.2e-16 ***
```

19.6 Interpretation:

Since P-Value is less than 0.05 we reject H_0 and conclude that atleast one coefficient is not zero or atleast one variable is making impact on dependent variable.

Step 3: Interpreting model Summary

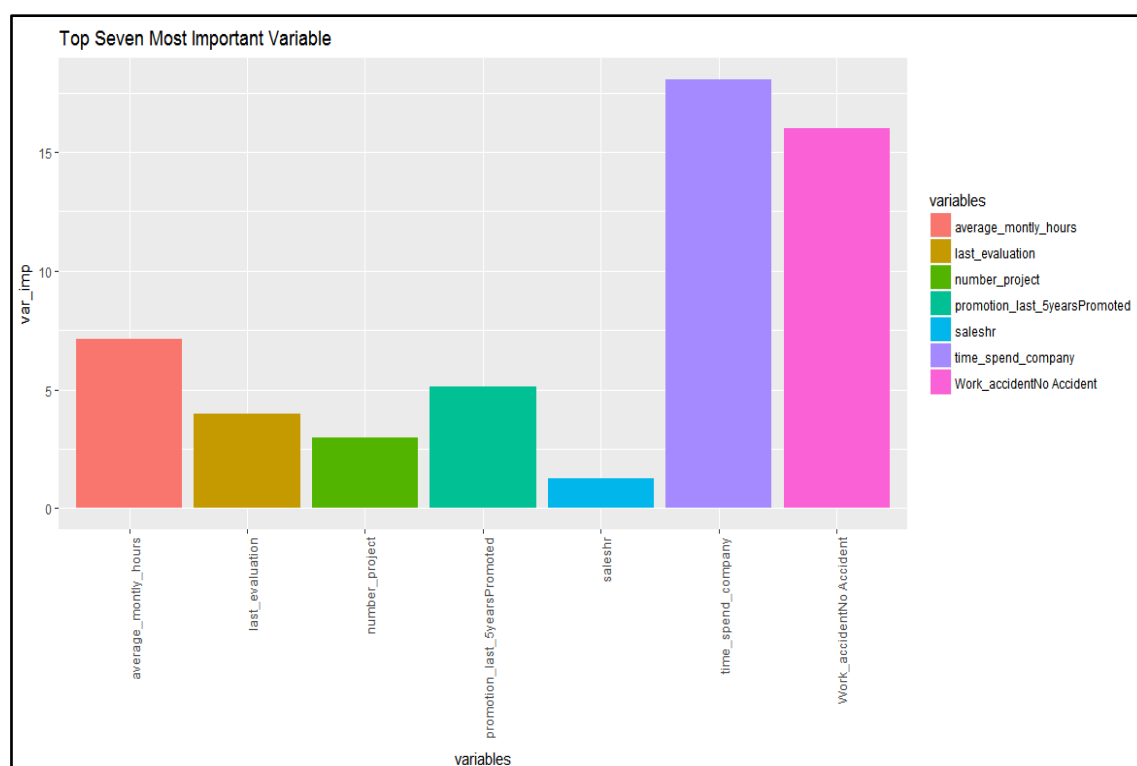
```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.0516472  0.2227755  22.676 < 2e-16 ***
last_evaluation  0.5783116  0.1458020   3.966 7.30e-05 ***
number_project  0.0618086  0.0207772   2.975 0.002932 **
average_monthly_hours -0.0036792  0.0005176  -7.108 1.18e-12 ***
time_spend_company -0.2844187  0.0157298 -18.081 < 2e-16 ***
work_accidentNo Accident -1.5285175  0.0956042 -15.988 < 2e-16 ***
promotion_last_5yearsPromoted 1.3158443  0.2557198   5.146 2.67e-07 ***
saleshr        -0.1730925  0.1353162  -1.279 0.200837
salesIT         0.3448281  0.1250151   2.758 0.005810 **
salesmanagement 0.5970735  0.1668275   3.579 0.000345 ***
salesmarketing  0.1796906  0.1353576   1.328 0.184335
salesproduct_mng 0.3905329  0.1348498   2.896 0.003779 **
salesRandD      0.6999827  0.1504383   4.653 3.27e-06 ***
salessales      0.1614356  0.1048509   1.540 0.123641
salessupport    0.1037433  0.1118951   0.927 0.353850
salestechnical  0.0705335  0.1088828   0.648 0.517119
salarylow      -1.9040250  0.1374129 -13.856 < 2e-16 ***
salarymedium   -1.3685506  0.1382692  -9.898 < 2e-16 ***
```

From the above output we can see that last_evaluation, average_monthly_hours, _work_accident, promotion in last 5 years, salary, are the most significant variables also for sales category salesmanagement and salesRandD are significant categories.

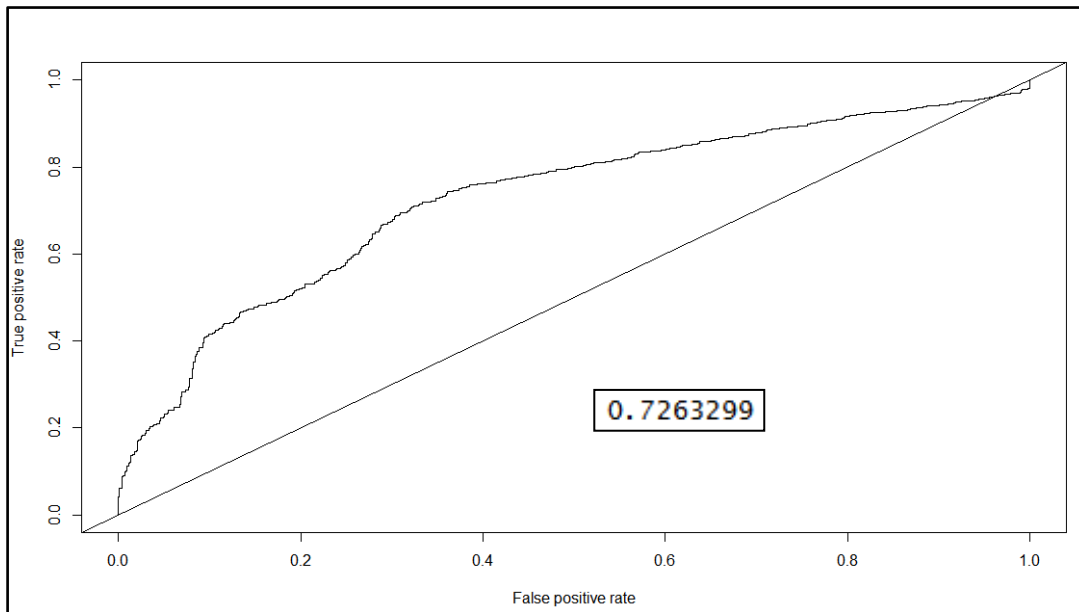
Step 4: Variable importance

From the logistic model we identified significant variables but it is important to find out order of importance below table shows order of significance for each variable

variables	var_imp
time_spend_company	17.5811877
Work_accidentNo Accident	16.1665204
salarylow	13.7205470
salarymedium	9.7219241
average_monthly_hours	6.7030190
promotion_last_5yearsPromoted	5.4063883
salesRandD	4.9719962
last_evaluation	4.6926876
number_project	3.1522544
salesmanagement	2.7371228
salesproduct_mng	2.1024282
salesIT	1.6960385
salessales	1.6348802
salesmarketing	1.2052193
saleshr	0.9681873
salessupport	0.8155086
salestechnical	0.2887437



Step 5: Measuring accuracy of the model



Area inside the curve which indicates accuracy of the model is **72.63%**

19.7 Conclusion

- From the Univariate analysis and Visualization we can conclude that although employees are contributing in more projects and spending long hours to complete these project also evaluation score is high despite of that they are not getting promotion and financial growth.
 - Payscale of Sales, Support and Technology department is low which could be a reason for higher attrition rate in these three departments
 - From Variable Importance plot we can see that Time spend in a company, Work accidents and salary are the three top most influencing factors for attrition of employees
-

Chapter 20

Factor Analysis

Dr. Santosh P. Gite, Associate Professor and Head, Department of Statistics,
University of Mumbai, Mumbai.

20.1 Introduction

Factor analysis is normally used to understand the correlation structure of collected data and identify the most important factors contributing to the data structure. In factor analysis, the relationship among a number of observed quantitative variables are represented in terms of a few underlying, independent variables called factors, which may not be directly measurable.

Factor analysis is related to principal components, but the two methods have different goals. Principal components seeks to identify orthogonal linear combination of the variables, to be used either for descriptive purposes or to substitute a smaller number of uncorrelated components for the original variables. In contrast, factor analysis represents a model for the data, and as such is more elaborate.

The factor model hypothesizes that the response vector X_1, X_2, \dots, X_m can be modeled as linear combinations of a smaller set of k unobserved (latent) random variables F_1, F_2, \dots, F_k , called common factors, along with an error term e_1, e_2, \dots, e_m .

The factor analysis model is

$$X - \mu = L F + e$$

Where, $X - \mu$ is the response vector, centered by the mean vector, L is the $m \times k$ matrix of factor loadings, F is $k \times 1$ vector of unobservable common factors and e is $m \times 1$ error vector. The factor analysis model differs from other models, such as the linear regression model, in that independent variables F_1, F_2, \dots, F_k are unobservable. Because of so many terms are unobserved, further assumption must be made before may uncover the factors from the observed response alone.

These assumptions are that $E(F) = 0$, $Cov(F) = I$, $E(e) = 0$ and $Cov(e)$ is diagonal matrix. Some terms which is useful in factor analysis such as communality, factor loadings, specific variance.

- 1) **Factor loadings:** factor loadings represents the correlation between the i^{th} variable and the j^{th} factor .
- 2) **Communality:** Communality represents the proportion of variance of a particular variable that is shared with other variable. The communalities represents the overall importance of each of the variable in the factor analysis as whole. Communality values are calculated as the sum of squared factor weights for a variable. Communalities less than 0.5 can be considered to be too small , since this would mean that the variable shares less than half of its variability in common with the other variable.
- 3) **Specific variance:** The proportion of variance of particular variance due to the specific factor is often called as specific variance.

20.2 Checking for Adequacy of the Data.

Before go to the perform factor analysis, the adequacy of the data is evaluated on the basis of the results of a Kaiser-Meyer-Olkin (KMO) sampling adequacy test and Bartlett's test of sphericity. The initial step is the determination of adequacy of the data for being use for factor analysis.

20.2.1 Kaiser-Meyer Olkin (KMO) measures

The proportion of variability within the standardized variables which is shared in common, and therefore caused by underlying factors, is measured by Kaiser-Meyer-Olkin measure of sampling adequacy. Values of the KMO statistics less than 0.5 indicate that the correlation between pairs of variable cannot be explained by the other variables and that factor analysis may not be appropriate.

20.2.2 Bartlett's test of sphericity

Bartlett's test for sphericity tests the null hypothesis that the correlation matrix is an identity matrix. Small p-value indicate that evidence against the null hypothesis(i.e. the variables really are correlated). For pvalues much larger than 0.05 indicated that there is insufficient evidence that variables are not correlated, so far factor analysis may not be suitable.

If KMO value is greater than 0.5 and Bartlett's test is significant then go for the factor analysis.

The second step is the estimation of the eigenvalues and factor loadings. Small eigenvalues contribute little to the explanatory capability of the data, only the first few factors are require to account for much of the parameter variability. Following methods are often using to extract important factors.

How many factors should be extract?

The criteria used for deciding how many factors to extract are 1) eigenvalue criterion 2) screeplot criterion.

a) Eigenvalue criterion

The sum of the eigenvalue represents the number of variables entered into the PCA. An eigenvalue of 1 would then mean that the component would explain about “one variables worth” of the variability. The rationale for using eigenvalue criterion is that each factor should explain at least one variables worth of the variability, and therefore eigenvalue criterion states that only factors with eigenvalue greater than 1 should be retained.

b) Scree Plot criterion

A scree plot is a graphical plot of the eigenvalue against the factor number. Scree plots are helpful for finding an upper bound (maximum) for the number of factors that should be retained. To determine the appropriate number of factors to be retained, one looks for an elbow (bend) in the scree plot. The number of factors to be retained is taken to be the point at which the elbow is found.

In factor analysis factor loadings and specific variances are unknown parameters. To estimate these parameters, we use two of the methods of parameter estimation, the Principal Component method and the Maximum Likelihood Method. The solution obtained from these methods can be rotated in order to simplify the interpretation of factors. Here we are using principal component method to estimate factor loadings and specific variances.

20.3 Factor rotation methods

To assist in the interpretation of the factors, the factor rotation may be performed. Factor rotation corresponds to a transformation of the coordinate axes, leading to different set of factor loadings.

20.3.1 Varimax rotation

Varimax rotation prefers to simplify the column of the factor loadings matrix. Varimax rotation maximizes the variability in the loadings for the factors. The rationale for varimax rotation is that we can best interpret the factors when they are strongly associated with some variable and strongly not associated with other variables.

20.3.2 Quartimax Rotation:

Quartimax rotation seeks to simplify the rows of a matrix of a factor loadings. Quartimax rotation tends to rotate the axes so that the variables have high loadings for the first factor and low loadings thereafter. The difficulty is that it can generate a strong “ general “ first factor, in which almost every variable has high loadings.

20.3.3 Equimax rotation

Equimax rotation seeks to compromise between simplifying the columns and rows.

20.3.4 Oblique Rotation

Oblique rotation method available in which the factors may be correlated with each other. In this study, factor extraction is performed by using the method of principal component for factoring. The widely accepted method for deciding the number of factors to use, is eigenvalue criterion(Kaiser Criterion), which retains only those factors with eigenvalues > 1. Factor loadings will be used to measure the correlation between variables and factors. Factor rotation also used to facilitate interpretation by providing simple factor structure. In this study we have used varimax rotation (Orthogonal) to interpret factors.

20.4 Applying Factor Analysis to the Online shopping data set:

513 sample size data and 16 variables were collected to study the factors that influences the customer to prefer online shopping. Following 16 variables were measured on 5 point likert type scale(Strongly agree, Agree, Indifferent, Disagree, Strongly disagree).

- X₁: Comparison of prices on various sites.
- X₂: Wide variety of brand choices.
- X₃: Getting latest product or product information.
- X₄: 24 hour accessibility
- X₅: No access to shop.
- X₆: Description of the product.
- X₇: E-Shopping is secure.
- X₈: Free shipping.
- X₉: Easy payback
- X₁₀: Convenience in time .
- X₁₁: Convenience in place.
- X₁₂: Low prices.
- X₁₃: Offers and discounts.
- X₁₄: Cash on delivery.
- X₁₅: Discreet shopping(privacy)
- X₁₆: Sufficient return time.

We have used R software to perform factor analysis on online shopping data to identify the factors that influences the customer to prefer online shopping. We are using “psych” package for factor analysis .

```
> Install.packages("psych")  
> Library(psych)
```

After installation of the “psych” package, import data from file sources to perform factor analysis. Use following command to import .csv data file.

```
> Online = read.csv(file.choose( ), header=TRUE)
```

Perform KMO test for adequacy of the data. To perform these test use following R code.

```
> KMO(online)
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = online)
## Overall MSA = 0.79
## MSA for each item =
```

Price	Variety	Product	Accesibility	Noacess
0.84	0.80	0.83	0.78	0.85
Product.1	Secure	Shiping	Payback	Convinencetime
0.82	0.84	0.83	0.83	0.60
Convinenceplace	Lowprices	discounts	COD	discreteshoping
0.62	0.78	0.77	0.82	0.84
returntime				
0.84				

The KMO function in the psych package produces an overall measure of sampling adequacy and MSA for each item.

The overall KMO for data is **0.79** , which is acceptable and this suggest that data is appropriate for factor analysis. i.e. variables and sample size are enough to proceed for factor analysis.

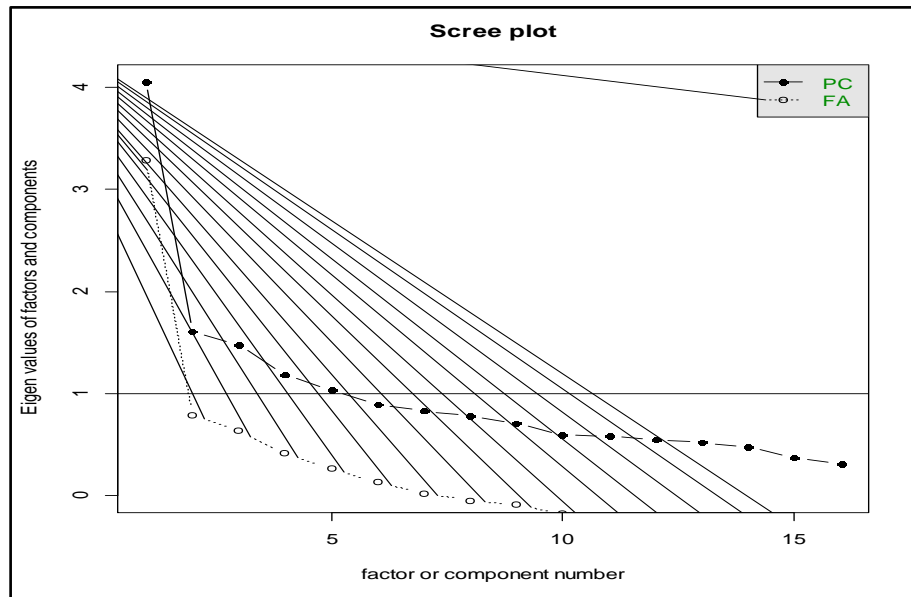
Bartlett’s test for sphericity tests the null hypothesis that the correlation matrix is an identity matrix. To perform Bartlett’s test for sphericity use following R code.

```
> cortest.bartlett(online)
## R was not square, finding R from data
## $chisq
## [1] 1821.756
## $p.value
## [1] 8.051493e-302
## $df
## [1] 120
```

Bartlett test is statistically significant, suggesting that correlation matrix is different from identity matrix. There is enough correlation between variable to proceed for factor analysis.

To demining the number of factors to extract, we use eigenvalue criterion and scree plot to extract important factors. Use the following R code.

```
scree(online)
```



Eigen values are a measure of the amount of variance accounted for by a factor. In the above graph, eigenvalue criterion states that only factors with eigenvalue greater than 1 should be retained. Therefore 5 factors are retained for factor analysis.

To perform factor analysis with 5 factor model and examine the variables that have high loadings on the factors. After examine high loadings, try to think what construct is common to these variable. After that construct, give name to the factors.

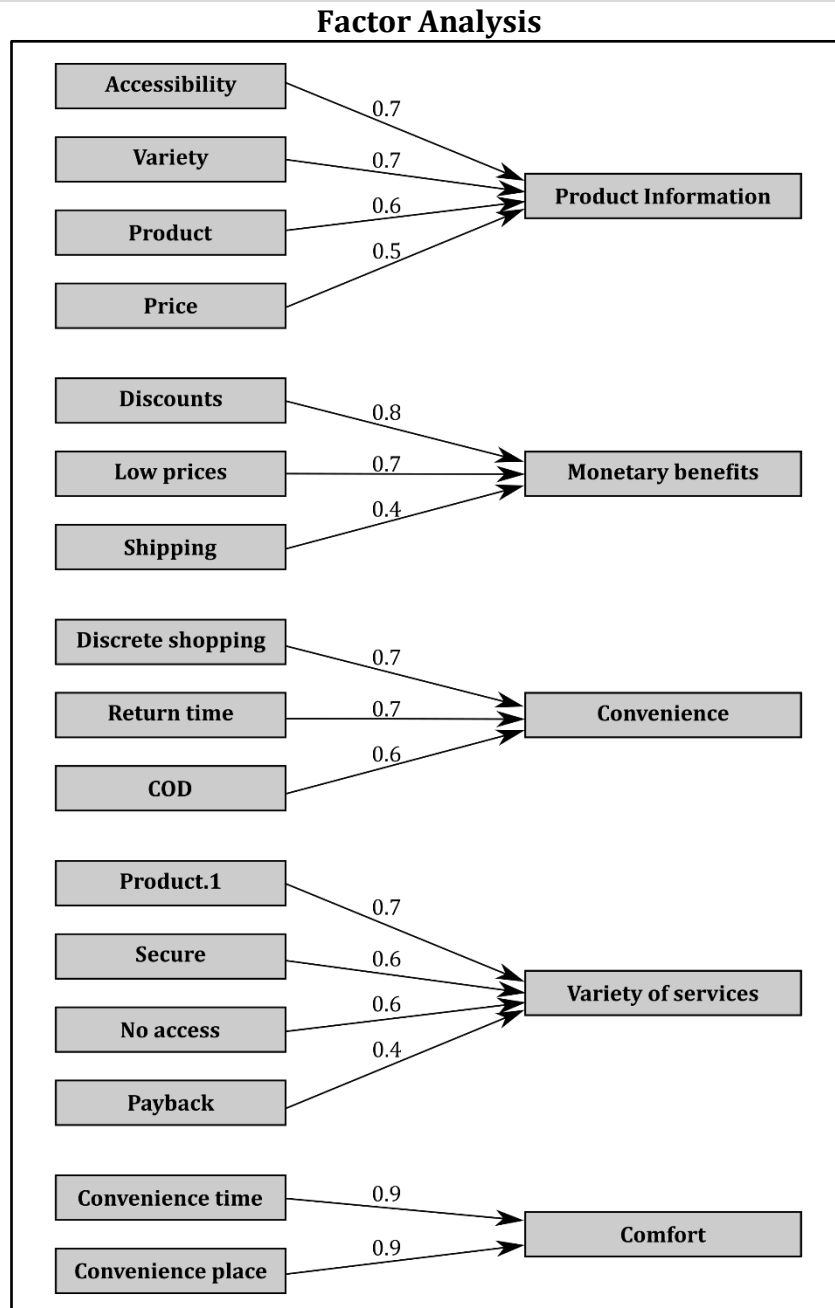
```
> fact.1=principal(online, nfactors=5, rotate="varimax")
colnames(fact.1$loadings)=c("Product Information", "Monetary benefits",
+"convenience", "variety of services", "Comfort")
> print(loadings(fact.1), digits=2, cutoff=0.35, sort=TRUE)
```

	Product Information	Monetary benefits	Convenience	variety of services	Comfort
Price	0.52				
Variety	0.71				
Product	0.64				
Accessability	0.72				
Lowprices		0.74			
discounts		0.80			
COD			0.62		
discreteshoping			0.74		
returntime			0.67		
Noaccess				0.58	
Product.1				0.70	
Secure				0.63	
Convinencetime					0.89
Convinenceplace					0.89
Shiping		0.40			
Payback				0.42	

It seems that there 5 factors. The first factor is something that is common to Accessibility, Variety, Product & price. It seems like a good name for this factor is “ **Product Information**” . The other four factors might be named “**Monetary benefits**”, “ **Convenience**”, “ **Variety of services**” and “ **Comfort**”.

Quick way to visualize your rotated factor solution. Use the following R code.

```
> fa.diagram(online)
```



20.5 Conclusion

Five (**Product Information**, **Monetary benefits**, **convenience**, **variety of services** and **comfort**) key factors are may be influencing to customer prefer online shopping.

Product manufacturing industries should look at the five factors to attract more consumers towards the online shopping.

20.6 References:

1. **Applied Multivariate Statistical Analysis**- Richard A. Johnson and Dean w. wichern.
 2. **A course in Statistics with R**: Tattar, Ramaiah and Manjunath.
 3. **A handbook of Statistical Analysis Using R**: Brian S. Everitt and Torsten Hotthorn
 4. **A Beginner's Guide to R**: Alain F. zuur, Elena N. leno and Meesters.
-

Chapter 21

Sentiment Analysis

Jain Jimit, Karani Hardik, Sen Milankumar, Dr. Asha Jindal*

Researcher, Star DBT Scheme, Dept. of Statistics, K.C. College, Mumbai - 20, INDIA.

*Star - DBT Mentor & Associate Professor and Head, Dept. of Statistics, K.C. College, Mumbai - 20.

21.1 Introduction

The aim of this chapter is to study real time tweets about trending topic #narendramodi. In this paper an attempt is made to classify tweets as positive or negative using a model. This information will be useful in gathering information about the general public response related to honourable PM Narendra Modi's political career and his work till date.

As we are a follower of Modi and he has millions of followers on Twitter, we thought of doing 'Sentiment Analysis' on #NARENDRAMODI keyword. This is interesting as Narendra Modi (often called as Modi) is the current Prime Minister of India and he is very active on social media. But India is a democratic country and everyone has right to speak so let's see how people are reacting to Modi.

a) About Twitter

- It is a social networking and micro blogging service.
- Enables users to send and read messages.
- Messages of length of up to 140 characters known as "tweets".
- Tweet contains rich information about people's preferences.
- People share their thoughts about political events using twitter

Data analysis on twitter data to predict the success of any event.

Social media and web plays a significant role in sentiments analysis.

b) What is Tweezer?

TWEEZER= Tweets + Analyser

This product (Tweezers) introduces a novel approach for automatically classifying the sentiment of twitter messages. These messages are classified as positive, negative or neutral with respect to query term or the keyword entered by a user.

c) What is Sentiment Analysis?

- Sentiment analysis refers to the use of natural language processing (NLP), text analysis to identify and extract subjective information from the source material.
- The main objective is to extract expressed opinions, emotions and sentiments in text.
- It allows us to track attitude and feelings, reviews, tweets of the products.
- Feedbacks of newly launched products.
- Sentiments or acceptance of an event (like-demonetization, movies reviews etc.)

d) Purpose

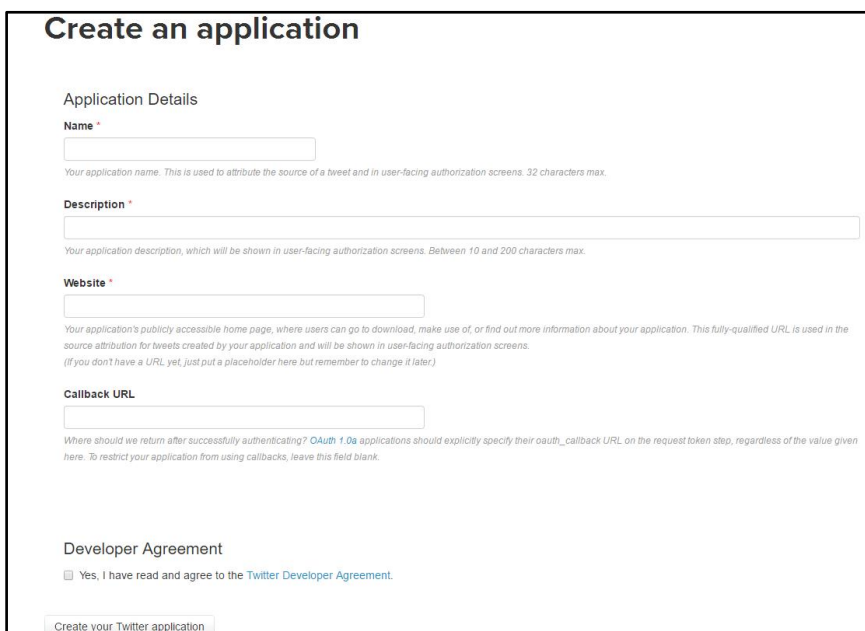
It allows individuals to get an opinion on P. M. Narendra Modi on a global scale.

e) Objective

- Our main objective is to do a real time analysis on twitter sentiments using R.
- Use the statistics of the tweets 'labels' to predict people's sentiment towards #narendramodi.
- Predicting the people's present opinions and views on #narendramodi.

21.2 How to Perform Sentiment Analysis?

- **Step 1**
 - Go to <https://apps.twitter.com/>
 - Create a new Twitter app
 - Enter your desired Application Name, Description and your website address making sure to enter the full address including the http://. You can leave the call back URL empty.



The screenshot shows the 'Create an application' form on the Twitter developer portal. It includes fields for Name, Description, Website, and Callback URL, each with a text input box and a small explanatory note below it. At the bottom, there is a 'Developer Agreement' section with a checkbox and a link to the agreement, followed by a 'Create your Twitter application' button.

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

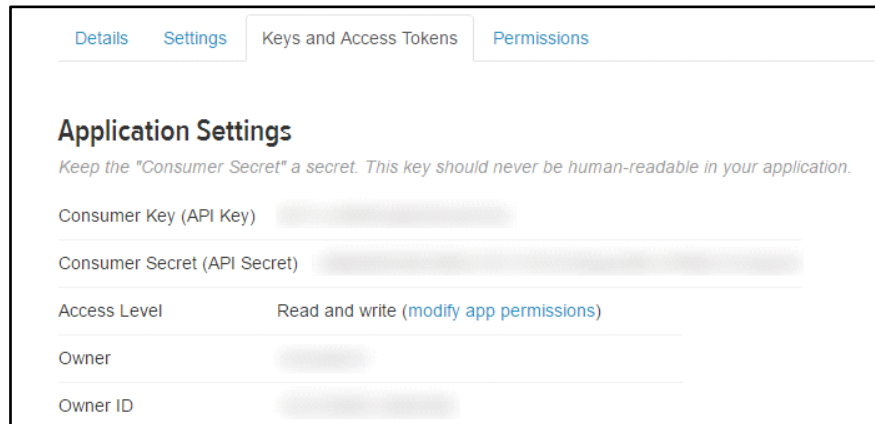
Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

☐ Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create your Twitter application

After registering, create an access token and grab your application's Consumer Key, Consumer Secret, Access token and Access token secret from Keys and Access Tokens tab.



In this article, sentiment analysis is performed using '#narendramodi' with 3000 tweets.

- **Step 2**

Packages that are used:

```
library(twitterR)
library(ROAuth)
library(plyr)
library(dplyr)
library(stringr)
library(ggplot2)
library(httr)
library(wordcloud)
library(tm)
library(Rstem)
library(psych)
#to be installed locally
library(sentiment)
```

- **Step 3**

Connect to API

```
oauth_endpoint(authorize = "https://api.twitter.com/oauth",
               access = "https://api.twitter.com/oauth/access_token")
#connect to API
download.file(url="http://curl.haxx.se/ca/cacert.pem",
             destfile="cacert.pem")
reqURL <- 'https://api.twitter.com/oauth/request_token'
accessURL <- 'https://api.twitter.com/oauth/access_token'
authURL <- 'https://api.twitter.com/oauth/authorize'
consumerKey="6mSD7XiBGxovXCT"
consumerSecret="KKuncF624KYNADGQiF9Wllysp0ylxGtBsxts"
accesstoken="921723452167294978-RJx2wwR8m168SB0lK9WHbqQhMj22Iby"
accesssecret="3mpaDDLcNXTQrT78DNqGeRvfjTk2hFt6W"
Cred <- OAuthFactory$new(consumerKey=consumerKey,
                        consumerSecret=consumerSecret,
                        requestURL=reqURL,
```



```

                                accessURL=accessURL,
                                authURL=authURL)
Cred$handshake(cainfo = system.file('CurlSSL', 'cacert.pem', package =
  'RCurl')) #There is URL in Console. You need to go to it, get code and
  enter it on Console
Once you launch the code first time, you can start from this line in the
  future (libraries should be connected)
setup_twitter_oauth(consumer_key=consumerKey,
                    consumer_secret=consumerSecret,
                    access_token=accesstoken,
                    access_secret=accesssecret)

```

- **Step 4**

```

# Harvest the tweets
all_tweets = searchTwitter("#narendramodi", n=3000, lang="en")
tweets_df <- twListToDF(all_tweets)
write.csv(tweets_df, "AllTweets.csv")
# get the text
tweet_txt=apply(all_tweets, function(x) x$getText())
# remove retweet entities
tweet_txt = gsub("(RT|via)((?:\\b\\W*@[\\w+)+)",
  "", tweet_txt)

```

- **Step 5**

```

class_emo = classify_emotion(tweet_txt,
                             algorithm="bayes", prior=1.0)
# get emotion best fit
emotion = class_emo[,7]
# substitute NA's by "unknown"
emotion[is.na(emotion)] = "unknown"
# classify polarity
class_pol = classify_polarity(tweet_txt,
                              algorithm="bayes")
# get polarity best fit
polarity = class_pol[,4]

```

- **Step 6**

```

# plot distribution of Emotions
ggplot(sent_df, aes(x=emotion)) +
  geom_bar(aes(y=..count.., fill=emotion)) +
  scale_fill_brewer(palette="Dark2") +
  labs(x="emotion categories", y="number of tweets",
       title="classification based on emotion")
## plot distribution of Polarity
ggplot(sent_df1, aes(x=polarity)) +
  geom_bar(aes(y=..count.., fill=polarity)) +
  scale_fill_brewer(palette="Dark2")
K, labs(x="polarity categories",
       y="number of tweets",
       title="classification based on polarity")

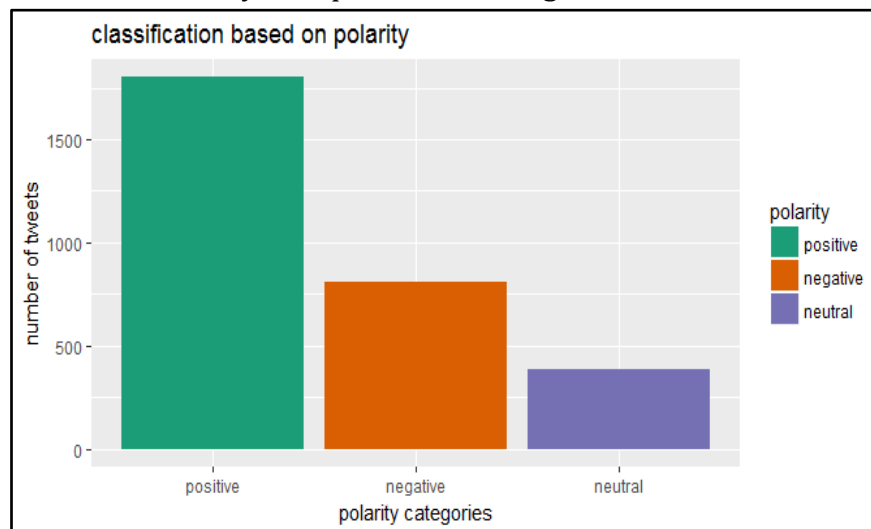
```

- **Step 7**

```
# First, separate the words according to emotions
tweet_emotions = levels(factor(sent_df$emotion))
n_tweet_emotions = length(tweet_emotions)
tweet_emotions_docs = rep(" ", n_tweet_emotions)
for (I in 1:n_tweet_emotions){
  tmp = tweet_txt[emotion == tweet_emotions[i]]
  tweet_emotions_docs[i] = paste(tmp, collapse=" ")
}
# Remove stopwords- Data Cleaning Step
tweet_emotions_docs=removeWords(tweet_emotions_docs,stopwords(kind = "en"))
TweetData.corpus = Corpus(VectorSource(tweet_emotions_docs))
TweetData.tdm = TermDocumentMatrix(TweetData.corpus)
TweetData.tdm = as.matrix(TweetData.tdm)
colnames(TweetData.tdm) = tweet_emotions
# creating, comparing and plotting the words on the cloud
comparison.cloud(TweetData.tdm, colors = brewer.pal(n_tweet_emotions,
```

21.3 Findings

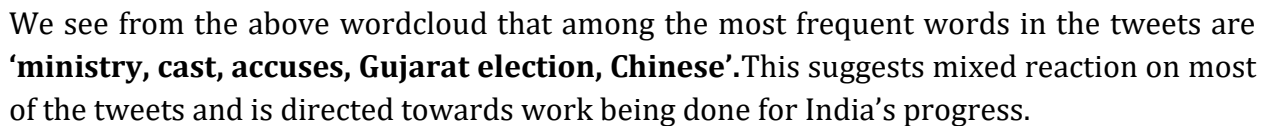
In this article, sentiment analysis is performed using ‘#narendramodi’ with 3000 tweets.



Looking at the above graph we can see that #narendramodi have more positive sentimental response from the tweets

Wordcloud

Wordclouds are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. The larger the word in the visual the more common word was in the document. This type of visualization can assist evaluators, with exploratory textual analysis by identifying words that frequently appear in a set of documents or other text. It can also be used for communicating the most salient points or themes in the reporting stage.



This paper tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. Twitter reviews are selected as data used for the study. A sentiment polarity categorization has been proposed along with detailed descriptions of each step. Experiments for both sentence level categorization and review level categorization have been performed which lead to the conclusion that there has been more positive sentimental response from the tweets. It shows that there is a strong possibility of him getting elected again for the 2019 elections.

Further work can be done in this area on understanding local language/ domain/ context to understand the data better. E.g. somebody using sarcasm/ double negative/ pun-related-to-some other context, etc. Further work using Machine Learning can be used here.

Chapter 22

Discriminant Analysis

Dr. Suresh Kumar Sharma, Professor, Department of Statistics & Coordinator,
Centre for Systems Biology & Bioinformatics, Panjab University, Chandigarh-India

22.1 Introduction

Discriminant analysis is a technique for analyzing data when the criterion or dependent variable is categorical and the predictor or independent variables are interval in nature.

The objectives of discriminant analysis are as follows:

- Development of discriminant functions, or linear combinations of the predictor or independent variables, which will best discriminate between the categories of the criterion or dependent variable (groups).
- Examination of whether significant differences exist among the groups, in terms of the predictor variables.
- Determination of which predictor variables contribute to most of the intergroup differences.
- Classification of cases to one of the groups based on the values of the predictor variables.
- Evaluation of the accuracy of classification.
- When the criterion variable has two categories, the technique is known as **two-group discriminant analysis**.
- When three or more categories are involved, the technique is referred to as **multiple discriminant analysis**.
- The main distinction is that, in the two-group case, it is possible to derive only one discriminant function. In multiple discriminant analysis, more than one function may be computed. In general, with G groups and k predictors, it is possible to estimate up to the smaller of $G - 1$, or k , discriminant functions.
- The first function has the highest ratio of between-groups to within-groups sum of squares. The second function, uncorrelated with the first, has the second highest ratio, and so on. However, not all the functions may be statistically significant.

22.2 Discriminant Analysis Model

The discriminant analysis model involves linear combinations of the following form:

$$D = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k$$

where

D = discriminant score

b 's = discriminant coefficient or weight
 X 's = predictor or independent variable

The coefficients or weights (b), are estimated so that the groups differ as much as possible on the values of the discriminant function. This occurs when the ratio of between-group sum of squares to within-group sum of squares for the discriminant scores is at a maximum.

Assumptions of the Model

The discriminant model has the following assumptions:

- The predictors are not highly correlated with each other
- The mean and variance of a given predictor are not correlated
- The correlation between two predictors is constant across groups
- The values of each predictor have a normal distribution.

22.3 Statistics Associated with Discriminant Analysis

- **Canonical Correlation.** Canonical correlation measures the extent of association between the discriminant scores and the groups. It is a measure of association between the single discriminant function and the set of dummy variables that define the group membership.
- **Centroid.** The centroid is the mean values for the discriminant scores for a particular group. There are as many centroids as there are groups, as there is one for each group. The means for a group on all the functions are the *group centroids*.
- **Classification matrix.** Sometimes also called *prediction matrix*, the classification matrix contains the number of correctly classified and misclassified cases.
- **Discriminant function coefficients.** The discriminant function coefficients (unstandardized) are the multipliers of variables, when the variables are in the original units of measurement.
- **Discriminant scores.** The unstandardized coefficients are multiplied by the values of the variables. These products are summed and added to the constant term to obtain the discriminant scores.
- **Eigen value.** For each discriminant function, the Eigen value is the ratio of between-group to within-group sums of squares. Large Eigen values imply superior functions.
- **F values and their significance.** These are calculated from a one-way ANOVA, with the grouping variable serving as the categorical independent variable. Each predictor, in turn, serves as the metric dependent variable in the ANOVA.
- **Group means and group standard deviations.** These are computed for each predictor for each group.
- **Pooled within-group correlation matrix.** The pooled within-group correlation matrix is computed by averaging the separate covariance matrices for all the groups.
- **Standardized discriminant function coefficients.** The standardized discriminant function coefficients are the discriminant function coefficients and are used as the multipliers when the variables have been standardized to a mean of 0 and a variance of 1.

- **Structure correlations.** Also referred to as *discriminant loadings*, the structure correlations represent the simple correlations between the predictors and the discriminant function.
- **Total correlation matrix.** If the cases are treated as if they were from a single sample and the correlations computed, a total correlation matrix is obtained.
- **Wilks' λ .** Sometimes also called the *U* statistic. Wilks' λ for each predictor is the ratio of the within-group sum of squares to the total sum of squares. Its value varies between 0 and 1. Large values of λ (near 1) indicate that group means do not seem to be different. Small values of λ (near 0) indicate that the group means seem to be different.

22.4 R-Code for Discriminant Analysis

Discriminant Analysis (Practice Example)

Example: This data called IRIS data set gives the measurements in centimeters of the variables sepal length, sepal width, petal length and petal width, respectively for 50 flowers from each of 3 species of IRIS. The three species are **setosa, versicolor, and virginica**. We shall perform Linear Discriminant Analysis on this data. Load Library MASS

```
>library(MASS)
```

First of all, read the data from iris file (on desktop) into R as follows:

```
>data1<-read.csv(("c:/desktop/iris.csv"))
```

```
>head(data1)
```

	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

#To apply Linear Discriminant Analysis (lda), we use function lda in R as follows:

```
>disc<-lda(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,
data=data1)
```

```
>disc
```

Call:

```
lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
    data = data1)
```

Prior probabilities of groups:

setosa	versicolor	virginica
0.3333333	0.3333333	0.3333333

Group means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

Coefficients of linear discriminants:

	LD1	LD2
Sepal.Length	0.8293776	0.02410215
Sepal.Width	1.5344731	2.16452123
Petal.Length	-2.2012117	-0.93192121
Petal.Width	-2.8104603	2.83918785

Since there are three Species, therefore prior probabilities are fixed equal ,that is, **0.3333**. For these three Species, two linear discriminant functions have been generated as LD1 and LD2.

If we want to predict the Species from Discriminant model, we use predict function in R as follos.

```
>disc.p<-predict(disc, newdata=data1[c(1,2,3,4)])$class
>disc.p
#Determine how well the model fits
>tab1<-table(disc.p,data=data1[,5]) # because 5th Column contains Species
>tab1
```

	data			
disc.p		setosa	versicolor	virginica
setosa		50	0	0
versicolor		0	48	1
virginica		0	2	49

The first Species **setosa** has been 100% corrected predicted by the discriminant model (50 out of 50). Species **versicolor** 96% (48 out of 50) and virginica 98% (49 out of 50). The overall accuracy can be assessed as follows:

```
>accuracy<-sum(diag(tab1)/sum(tab1))*100
>accuracy
98
```

In this example the accuracy, that is, correct classification for the model is about 98%, which is extremely good.

Chapter 23

Cluster Analysis

Dr. Suresh Kumar Sharma, Professor, Department of Statistics & Coordinator,
Centre for Systems Biology & Bioinformatics, Panjab University, Chandigarh-India.

23.1 Introduction

Cluster analysis is a class of techniques used to classify objects or cases into relatively homogeneous groups called clusters. Objects in each cluster tend to be similar to each other and dissimilar to objects in the other clusters. Both cluster analysis and discriminant analysis are concerned with classification. However, Discriminant analysis requires prior knowledge of the cluster or group membership for each object or case included, to develop the classification rule. In cluster analysis there is no a priori information about the group or cluster membership for any of the objects. Groups or clusters are suggested by the data, not defined a priori.

23.2 Statistics Associated with Cluster Analysis

- **Agglomeration schedule.** An agglomeration schedule gives information on the objects or cases being combined at each stage of a hierarchical clustering process.
- **Cluster centroid.** The cluster centroid is the mean values of the variables for all the cases or objects in a particular cluster.
- **Cluster centers.** The cluster centers are the initial starting points in nonhierarchical clustering. Clusters are built around these centers, or *seeds*.
- **Cluster membership.** Cluster membership indicates the cluster to which each object or case belongs.
- **Dendrogram.** A dendrogram, or *tree graph*, is a graphical device for displaying clustering results. Vertical lines represent clusters that are joined together. The position of the line on the scale indicates the distances at which clusters were joined. The dendrogram is read from left to right.
- **Distances between cluster centers.** These distances indicate how separated the individual pairs of clusters are. Clusters that are widely separated are distinct, and therefore desirable.
- **Icicle diagram.** An icicle diagram is a graphical display of clustering results, so called because it resembles a row of icicles hanging from the eaves of a house. The columns correspond to the objects being clustered, and the rows correspond to the number of clusters. An icicle diagram is read from bottom to top.

- **Similarity/distance coefficient matrix.** A similarity/distance coefficient matrix is a lower-triangle matrix containing pairwise distances between objects or cases.

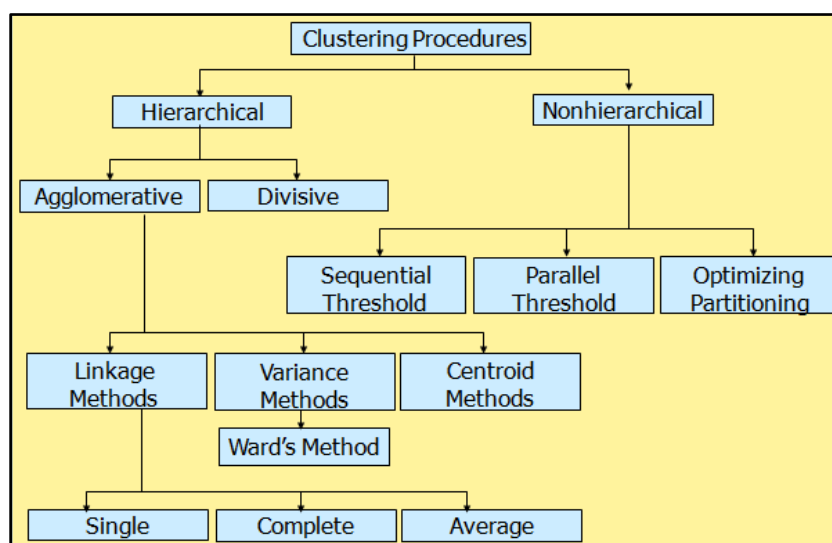
23.3 Conducting Cluster Analysis

- Perhaps the most important part of formulating the clustering problem is selecting the variables on which the clustering is based.
- Inclusion of even one or two irrelevant variables may distort an otherwise useful clustering solution.
- Basically, the set of variables selected should describe the similarity between objects in terms that are relevant to the marketing research problem.
- The variables should be selected based on past research, theory, or a consideration of the hypotheses being tested. In exploratory research, the researcher should exercise judgment and intuition.

23.4 Select a Distance or Similarity Measure

- The most commonly used measure of similarity is the Euclidean distance or its square. The **Euclidean distance** is the square root of the sum of the squared differences in values for each variable.
- If the variables are measured in vastly different units, the clustering solution will be influenced by the units of measurement. In these cases, before clustering respondents, we must standardize the data by rescaling each variable to have a mean of zero and a standard deviation of unity. It is also desirable to eliminate outliers (cases with a typical values).
- Use of different distance measures may lead to different clustering results. Hence, it is advisable to use different measures and compare the results.

23.5 A Classification of Clustering Procedures



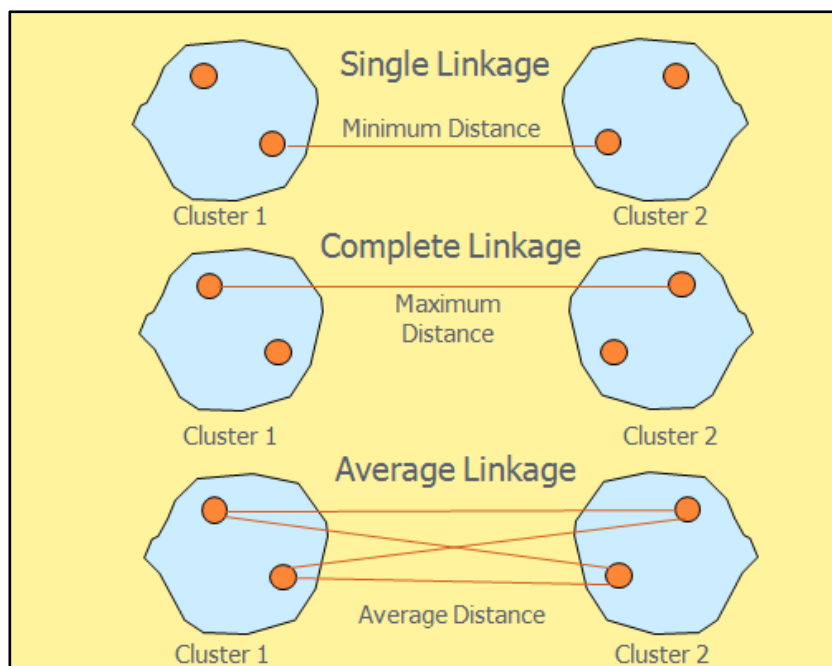
23.5.1 Select a Clustering Procedure – Hierarchical

- **Hierarchical clustering** is characterized by the development of a hierarchy or tree-like structure. Hierarchical methods can be agglomerative or divisive.
- **Agglomerative clustering** starts with each object in a separate cluster. Clusters are formed by grouping objects into bigger and bigger clusters. This process is continued until all objects are members of a single cluster.
- **Divisive clustering** starts with all the objects grouped in a single cluster. Clusters are divided or split until each object is in a separate cluster.
- **Agglomerative methods** are commonly used in marketing research. They consist of linkage methods, error sums of squares or variance methods, and centroid methods.

23.5.2 Select a Clustering Procedure – Linkage Method

- ❑ The **single linkage** method is based on minimum distance, or the nearest neighbor rule. At every stage, the distance between two clusters is the distance between their two closest points
- ❑ The **complete linkage** method is similar to single linkage, except that it is based on the maximum distance or the furthest neighbor approach. In complete linkage, the distance between two clusters is calculated as the distance between their two furthest points
- ❑ The **average linkage** method works similarly. However, in this method, the distance between two clusters is defined as the average of the distances between all pairs of objects, where one member of the pair is from each of the clusters

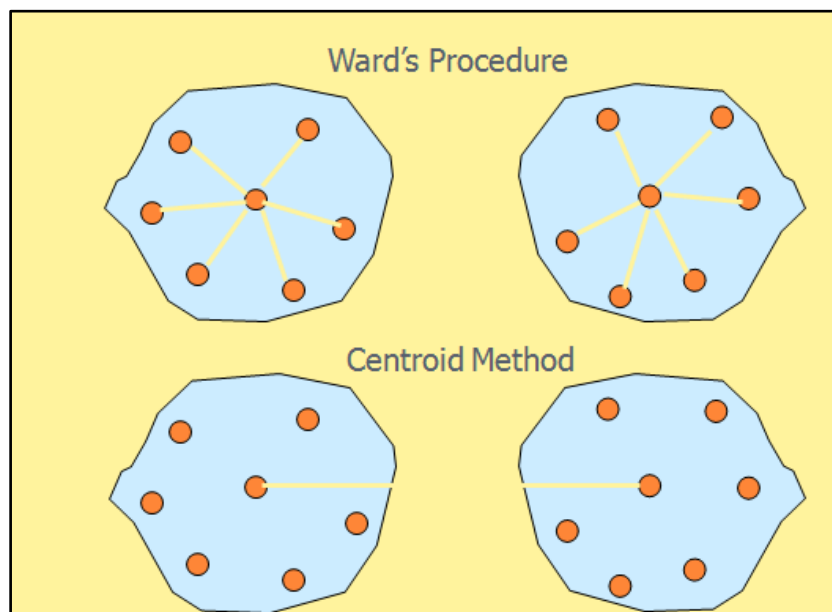
23.5.3 Linkage Methods of Clustering



23.5.4 Select a Clustering Procedure – Variance Method

- The **variance methods** attempt to generate clusters to minimize the within-cluster variance
- A commonly used variance method is the **Ward's procedure**. For each cluster, the means for all the variables are computed. Then, for each object, the squared Euclidean distance to the cluster means is calculated. These distances are summed for all the objects. At each stage, the two clusters with the smallest increase in the overall sum of squares within cluster distances are combined
- In the **centroid methods**, the distance between two clusters is the distance between their centroids (means for all the variables), as shown in Figure. Every time objects are grouped, a new centroid is computed.
- Of the **hierarchical methods**, **average linkage** and **Ward's methods** have been shown to perform better than the other procedures.

23.5.5 Other Agglomerative Clustering Methods



Select a Clustering Procedure – Non-hierarchical

- The nonhierarchical clustering methods are frequently referred to as *k*-means clustering. These methods include sequential threshold, parallel threshold, and optimizing partitioning
- In the sequential threshold method, a cluster center is selected and all objects within a pre-specified threshold value from the center are grouped together. Then a new cluster center or seed is selected, and the process is repeated for the unclustered points. Once an object is clustered with a seed, it is no longer considered for clustering with subsequent seeds

- The parallel threshold method operates similarly, except that several cluster centers are selected simultaneously and objects within the threshold level are grouped with the nearest center
- The optimizing partitioning method differs from the two threshold procedures in that objects can later be reassigned to clusters to optimize an overall criterion, such as average within cluster distance for a given number of clusters.
- It has been suggested that the hierarchical and nonhierarchical methods be used in tandem. First, an initial clustering solution is obtained using a hierarchical procedure, such as average linkage or Ward's. The number of clusters and cluster centroids so obtained are used as inputs to the optimizing partitioning method
- Choice of a clustering method and choice of a distance measure are interrelated. For example, squared Euclidean distances should be used with the Ward's and centroid methods. Several nonhierarchical procedures also use squared Euclidean distances.

23.6 Decide on the Number of Clusters

- Theoretical, conceptual, or practical considerations may suggest a certain number of clusters
- In hierarchical clustering, the distances at which clusters are combined can be used as criteria. This information can be obtained from the agglomeration schedule or from the dendrogram
- In nonhierarchical clustering, the ratio of total within-group variance to between-group variance can be plotted against the number of clusters. The point at which an elbow or a sharp bend occurs indicates an appropriate number of clusters
- The relative sizes of the clusters should be meaningful.

23.7 Interpreting and Profiling the Clusters

- Interpreting and profiling clusters involves examining the cluster centroids. The centroids enable us to describe each cluster by assigning it a name or label
- It is often helpful to profile the clusters in terms of variables that were not used for clustering. These may include demographic, psychographic, product usage, media usage, or other variables.

23.8 Assess Reliability and Validity

- Perform cluster analysis on the same data using different distance measures. Compare the results across measures to determine the stability of the solutions.
- Use different methods of clustering and compare the results.
- Split the data randomly into halves. Perform clustering separately on each half. Compare cluster centroids across the two subsamples.

- Delete variables randomly. Perform clustering based on the reduced set of variables. Compare the results with those obtained by clustering based on the entire set of variables.
- In nonhierarchical clustering, the solution may depend on the order of cases in the data set. Make multiple runs using different order of cases until the solution stabilizes.

23.9 R-Code for Cluster Analysis

Hierarchical Cluster Analysis (Practice Example)

Example: The data related to 32 different Cars (Make) with information related to 11 variables like: mileage, cylinder, displacement, horse power etc. and the data is recorded in Excel file names as: **mtcars**

```
mpg (v1)
cyl  (v2)
disp (v3)
hp   (v4)
drat (v5)
wt   (v6)
qsec (v7)
vs   (v8)
am   (v9)
gear (v10)
carb (v11)
```

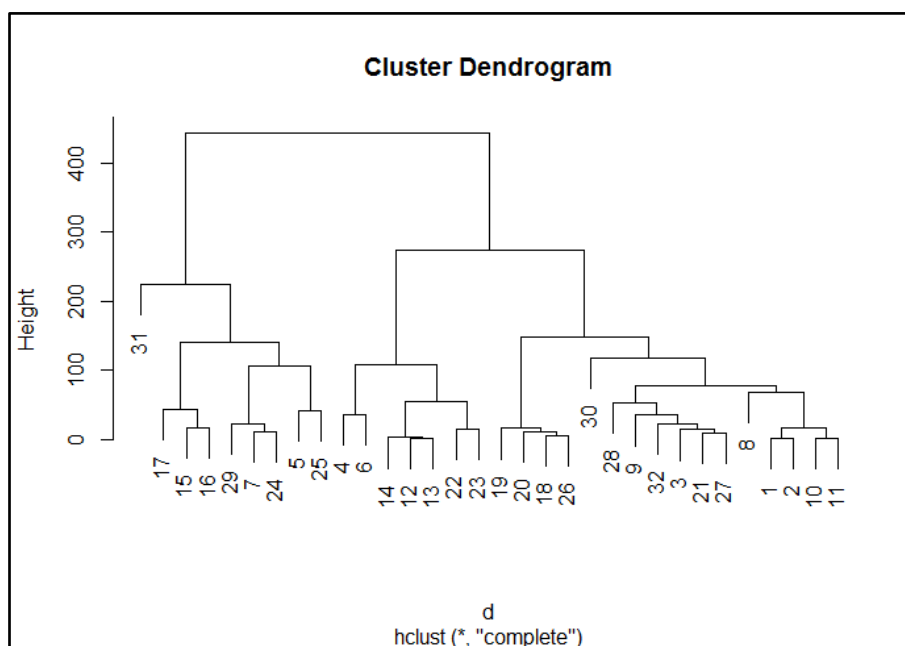
Our first aim is to import the file, say on desktop, into R as follows:

```
> data<-read.csv("c:/desktop/mtcars.csv")
> head (data)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1

Next, we compute distance matrix, apply hierarchal clustering and plot dendrogram as follows:

```
> d <- dist(as.matrix(data))      # find distance matrix (distance matrix
will be displayed)
> hc <- hclust(d)                  # apply hierarchical clustering
> plot(hc)                        # plot the dendrogram
```



Careful inspection of the dendrogram will help us to identify approximate number of Clusters. Then we can perform Non-hierarchical clustering by specifying approximate numbers of Clusters. (obtained from hierarchical clustering).

In general, there are many choices of cluster analysis methodology. The `hclust` function in R uses the **complete linkage method for hierarchical clustering by default**. This particular clustering method defines the cluster distance between two clusters to be the maximum distance between their individual components. At every stage of the clustering process, the two nearest clusters are merged into a new cluster. **The process is repeated until the whole data set is agglomerated into one single cluster.**

Suppose dendrogram results into three clusters, and then we can form Non-hierarchical clustering (**kmeans**) by specifying approximate numbers of Clusters as follows:

```
>results<-kmeans (data, 3)
>results
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
"betweeness"
[7] "size"         "iter"         "ifault"
```

```
>results$size      # will display the size of each cluster
>results$centres   # will display the cluster centres
```

Similarly, other information for above components can be displayed.



Kishinchand Chellaram College

Vidyasagar Principal K.M. Kundnani Chowk,
124, Dinshaw Wachha Road,
Churchgate, Mumbai 400020.

Tel: +91-22-2285 5726; +91-22-6698 1000;

Fax: +91-22-2202 9092;

Email: office@kcccollege.edu.in

Website: <http://www.kcccollege.edu.in/>



Shailja Prakashan

57-P, Kunj Vihar-II,
Yashoda Nagar, Kanpur-11

Ph.: 0512-2633004

Email: shailjaparakashan@gmail.com

ISBN 978-93-80788-71-5



ISBN 978-93-80788-71-5