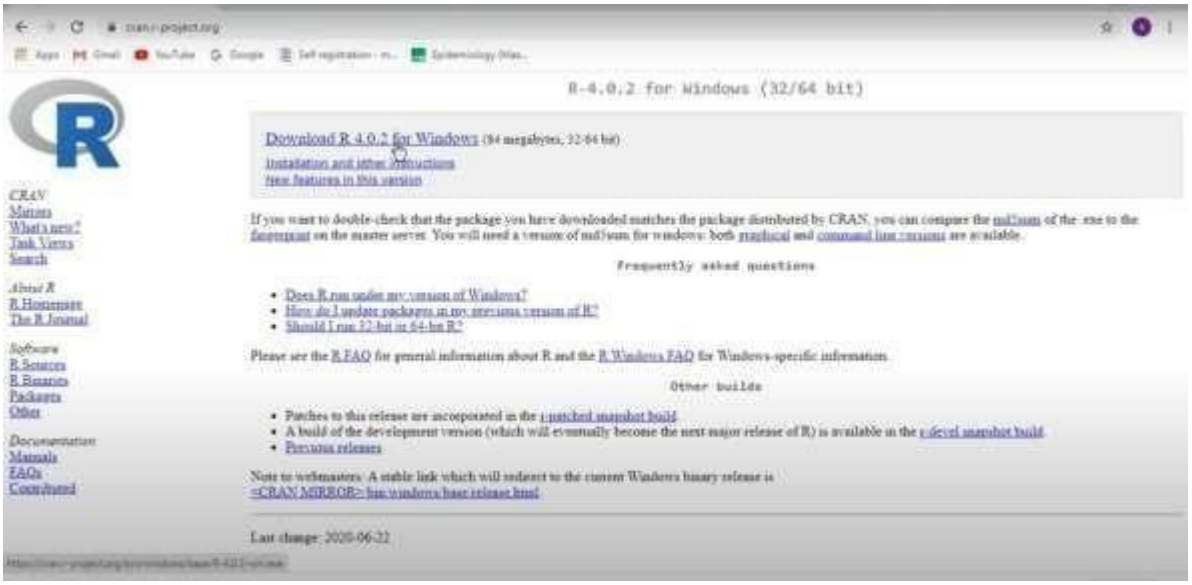
INDEX

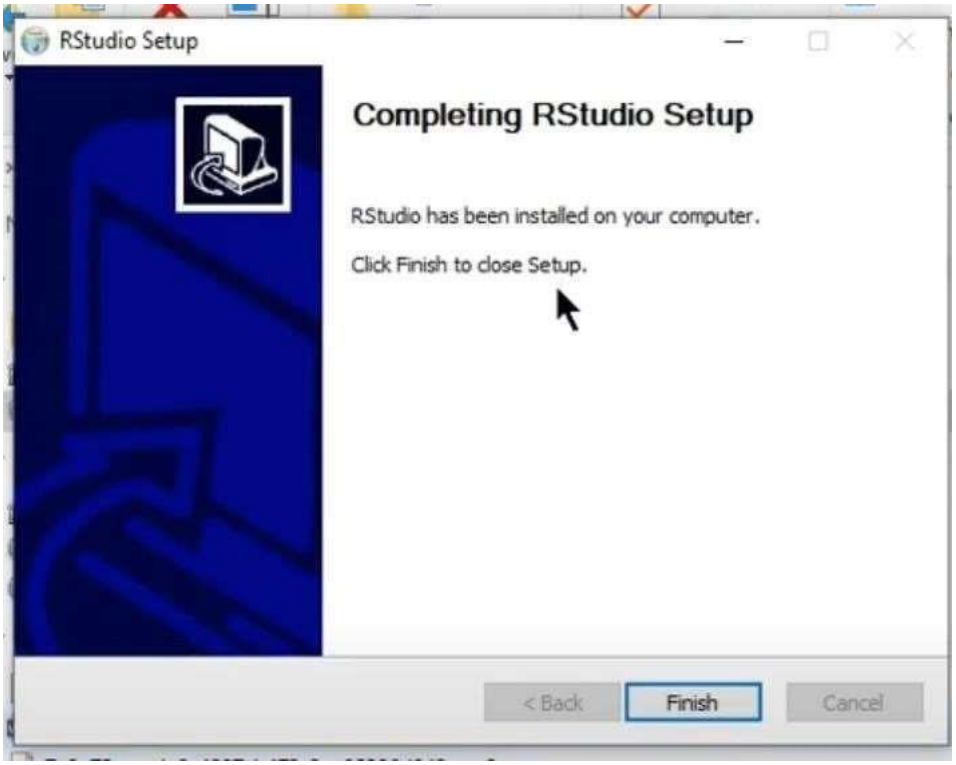
8		Implement a method to treat missing value for gender and missing value for marks.	93		
9		Implement linear regression to predict the 5th semester result of student.			
10		Implement logistic regression and decision tree to classify the student as average or clever.	102		

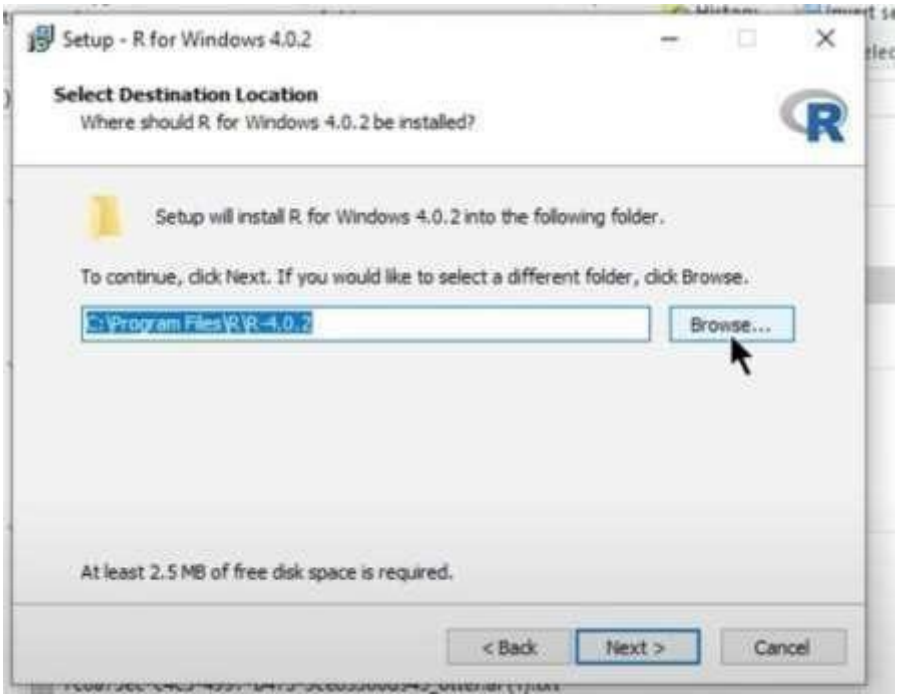
Beyond Syllabus					
11		Stock market prediction project with R.			
Assignments					
11		Assignment - 1			
12		Assignment - 2			

Practical 1

Aim- Study and Installation of R tool.




RStudio Setup
Completing RStudio Setup
RStudio has been installed on your computer.
Click Finish to close Setup.
< Back
Finish
Cancel

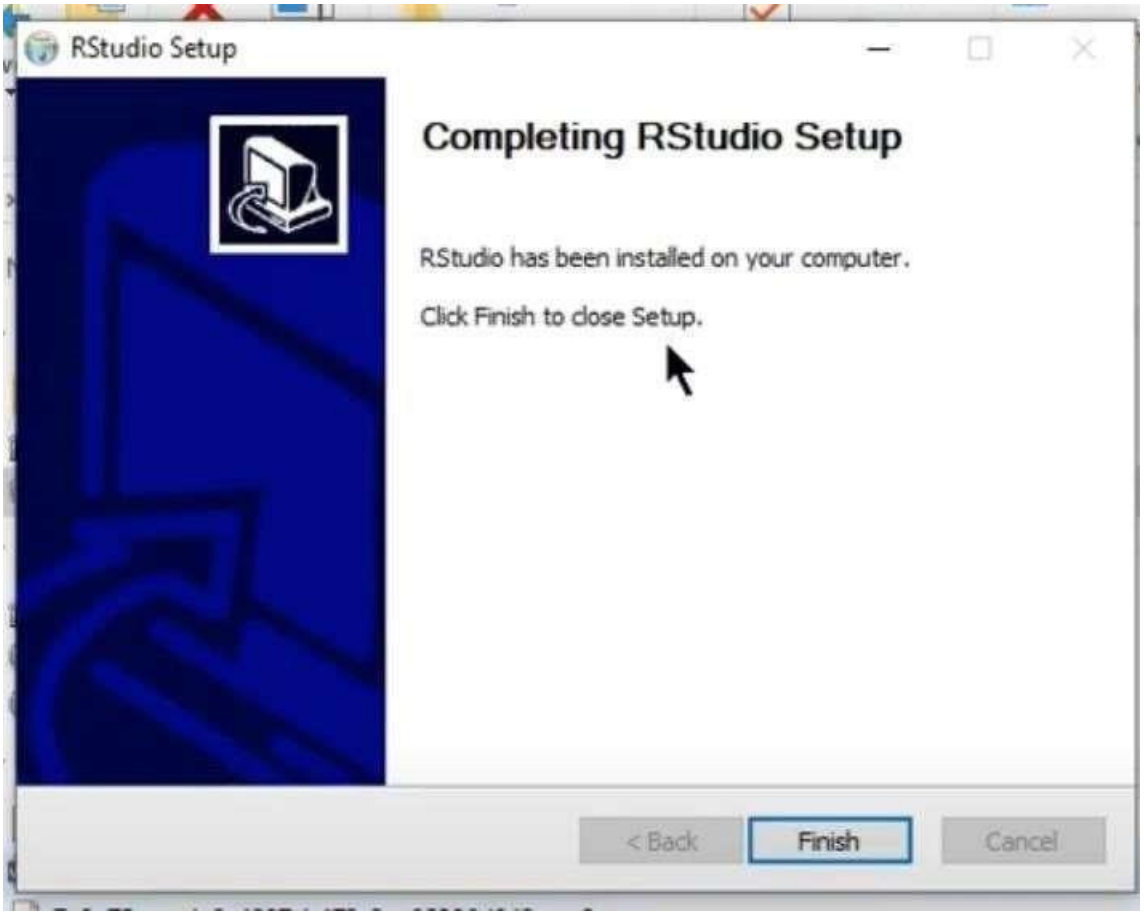

Setup - R for Windows 4.0.2
Select Destination Location
Where should R for Windows 4.0.2 be installed?
Setup will install R for Windows 4.0.2 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\R\R-4.0.2
Browse...
At least 2.5 MB of free disk space is required.
< Back
Next >
Cancel

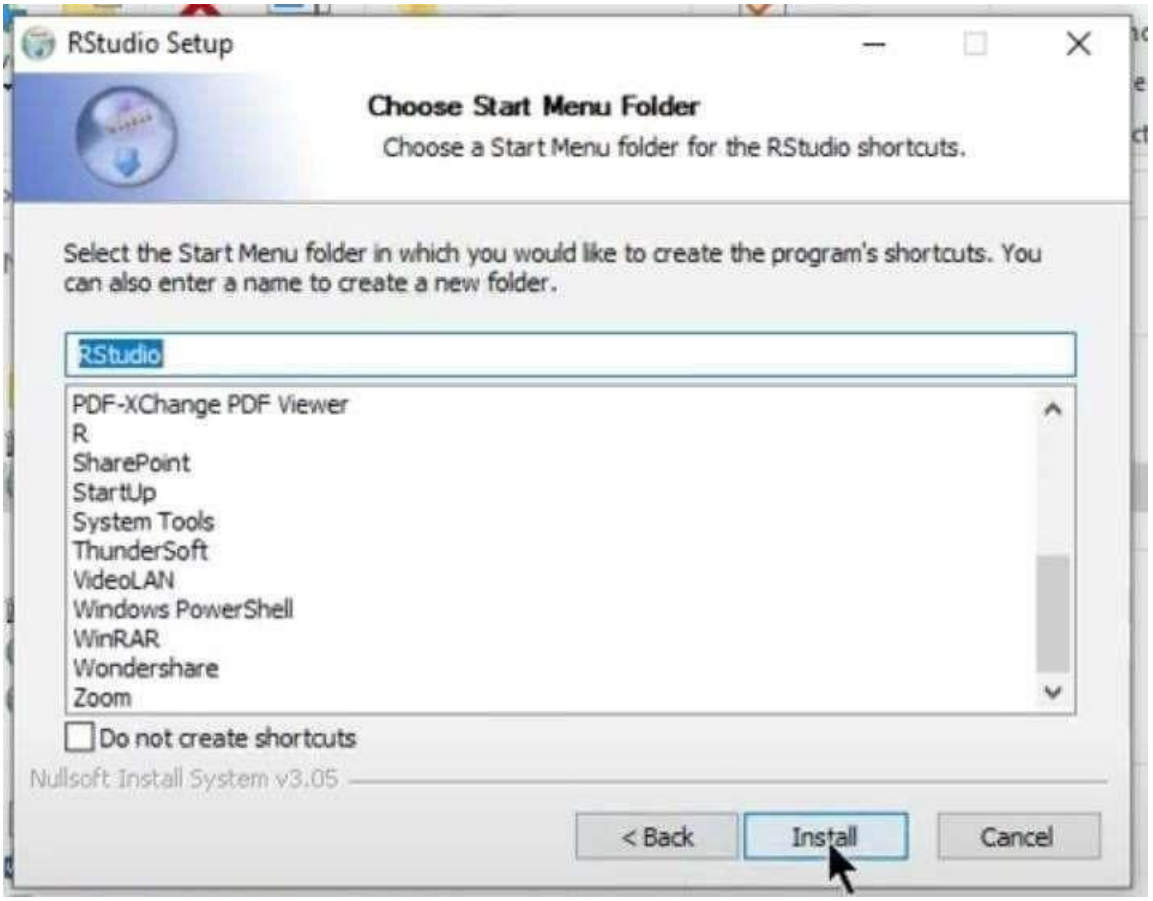

Setup - R for Windows 4.0.2
Installing
Please wait while Setup installs R for Windows 4.0.2 on your computer.
Extracting files...
C:\Program Files\R\R-4.0.2\bin\i386\Rgraphapp.dll
Cancel

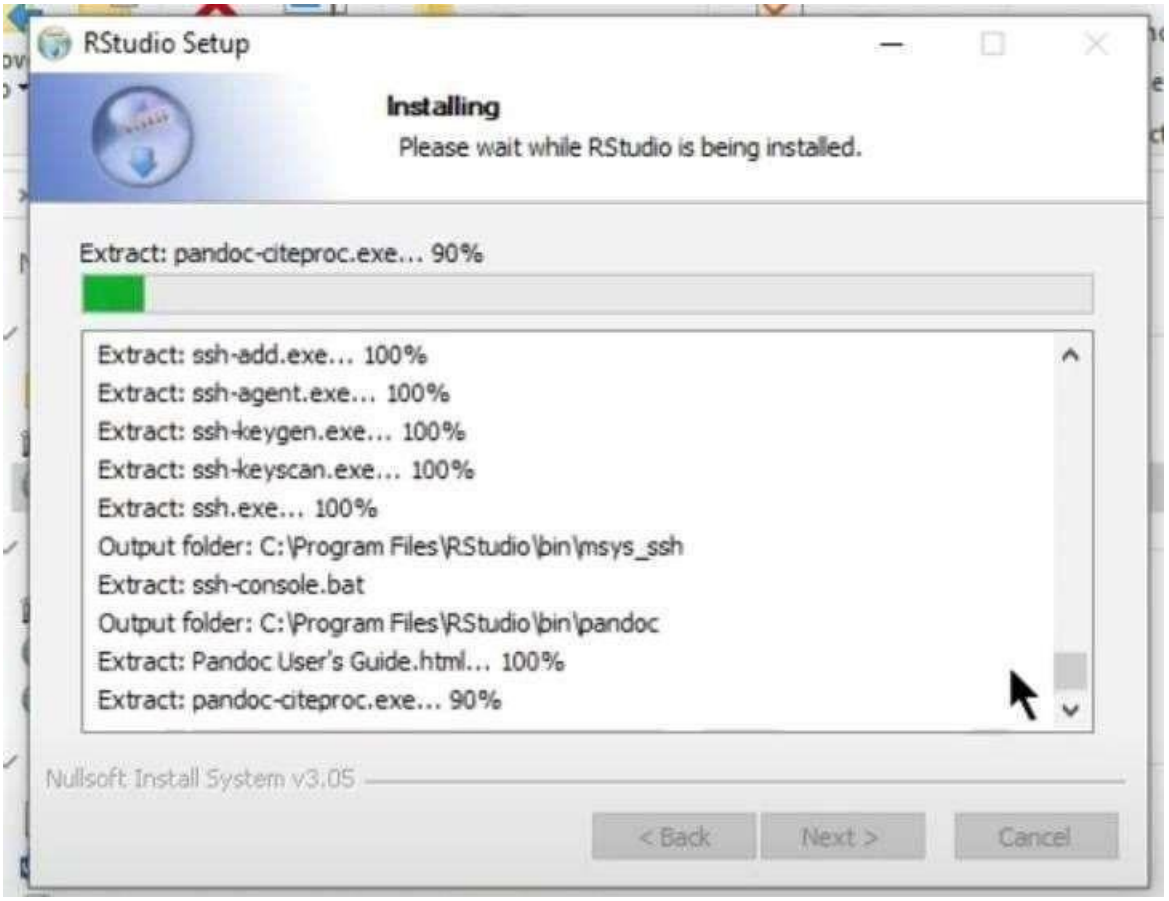

Setup - R for Windows 4.0.2
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
User installation
Core Files 87.1 MB
32-bit Files 50.6 MB
64-bit Files 57.4 MB
Message translations 7.3 MB
Current selection requires at least 204.6 MB of disk space.
< Back
Next >
Cancel


Setup - R for Windows 4.0.2
Completing the R for Windows 4.0.2 Setup Wizard
Setup has finished installing R for Windows 4.0.2 on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Finish


Studio
Products
Solutions
Customers
Resources
About
Pricing
Download the RStudio IDE
Choose Your Version
The RStudio IDE is a set of integrated tools designed to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.
Studio Team


RStudio Setup
Completing RStudio Setup
RStudio has been installed on your computer.
Click Finish to close Setup.
< Back
Finish
Cancel


RStudio Setup
Choose Start Menu Folder
Choose a Start Menu folder for the RStudio shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
RStudio
PDF-XChange PDF Viewer
R
SharePoint
StartUp
System Tools
ThunderSoft
VideoLAN
Windows PowerShell
WinRAR
Wondershare
Zoom
Do not create shortcuts
Nullsoft Install System v3.05
< Back
Install
Cancel


RStudio Setup
Installing
Please wait while RStudio is being installed.
Extract: pandoc-citeproc.exe... 90%
Extract: ssh-add.exe... 100%
Extract: ssh-agent.exe... 100%
Extract: ssh-keygen.exe... 100%
Extract: ssh-keyscan.exe... 100%
Extract: ssh.exe... 100%
Output folder: C:\Program Files\RStudio\bin\msys_ssh
Extract: ssh-console.bat
Output folder: C:\Program Files\RStudio\bin\pandoc
Extract: Pandoc User's Guide.html... 100%
Extract: pandoc-citeproc.exe... 90%
Nullsoft Install System v3.05
< Back
Next >
Cancel


RStudio Setup
Welcome to RStudio Setup
Setup will guide you through the installation of RStudio.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

Practical 2

Aim- Performing Commands in R: Data Types, Variables, Operators, Decision Making, Loops, Functions, String, Vector, Matrices, Lists, Arrays, Data Frames, Factors, File Reading.

- Data Types
  - Logical

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

  - Numeric

```
> v <- 96.6
> print(class(v))
[1] "numeric"
> v <- .058
> print(class(v))
[1] "numeric"
>
```

  - Integer

```
> v <- 02L
> print(class(v))
[1] "integer"
> v <- 10L
> print(class(v))
[1] "integer"
>
```

  - Complex

```
> v <- 5+8i
> print(class(v))
[1] "complex"
> v <- 5+10i
> print(class(v))
[1] "complex"
>
```

  - Character

```
> v <- "apple"
> print(class(v))
[1] "character"
> v <- "music"
> print(class(v))
[1] "character"
> |
```

- Raw

```
> v <- charToRaw("Hello")
> print(class(v))
[1] "raw"
> v <- charToRaw("apple")
> print(class(v))
[1] "raw"
> |
```

- Variables

- Variable assignment

```
> # Assignment using equal operator.
> var.1 = c(0,1,2,3)
> # Assignment using leftward operator.
> var.2 <- c("learn","R")
> # Assignment using rightward operator.
> c(TRUE,1) -> var.3
> print(var.1)
[1] 0 1 2 3
> cat ("var.1 is ", var.1 ,"\n")
var.1 is  0 1 2 3
> cat ("var.2 is ", var.2 ,"\n")
var.2 is  learn R
> cat ("var.3 is ", var.3 ,"\n")
var.3 is  1 1
> |
```

- Data type of variable

```
> var_x <- "Hello"
> cat("The class of var_x is ",class(var_x),"\n")
The class of var_x is  character
> var_x <- 34.5
> cat(" Now the class of var_x is ",class(var_x),"\n")
 Now the class of var_x is  numeric
> var_x <- 27L
> cat(" Next the class of var_x becomes ",class(var_x),"\n")
 Next the class of var_x becomes  integer
>
```

- Finding variable

```
> print(ls())
[1] "a"       "apple"   "iris"    "st_data" "var.1"   "var.2"   "var.3"
[8] "var_x"
> # List the variables starting with the pattern "var".
> print(ls(pattern="var"))
[1] "var.1" "var.2" "var.3" "var_x"
> print(ls(all.name=TRUE))
[1] "a"       "apple"   "iris"    "st_data" "var.1"   "var.2"   "var.3"
[8] "var_x"
>
```

- Deleting variable

```
> rm(var.3)
> print(var.3)
Error in print(var.3) : object 'var.3' not found
> rm(list=ls())
> print(ls())
character(0)
>
```

- Operators
  - Arithmetic Operators

```
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v+t)
[1] 10.0  8.5 10.0
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v-t)
[1] -6.0  2.5  2.0
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v*t)
[1] 16.0 16.5 24.0
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v/t)
[1] 0.250000 1.833333 1.500000
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v%%t)
[1] 2.0 2.5 2.0
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v%/%t)
[1] 0 1 1
> v <- c( 2,5.5,6)
> t <- c(8, 3, 4)
> print(v^t)
[1]  256.000  166.375 1296.000
>
```

- Relational Operators

```
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v>t)
[1] FALSE  TRUE FALSE FALSE
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v < t)
[1]  TRUE FALSE  TRUE FALSE
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v==t)
[1] FALSE FALSE FALSE  TRUE
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v<=t)
[1]  TRUE FALSE  TRUE  TRUE
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v>=t)
[1] FALSE  TRUE FALSE  TRUE
> v <- c(2,5.5,6,9)
> t <- c(8,2.5,14,9)
> print(v!=t)
[1]  TRUE  TRUE  TRUE FALSE
> |
```

- Logical Operators

```
> v <- c(3,1,TRUE,2+3i)
> t <- c(4,1,FALSE,2+3i)
> print(v&t)
[1]  TRUE  TRUE FALSE  TRUE
> v <- c(3,0,TRUE,2+2i)
> t <- c(4,0,FALSE,2+3i)
> print(v|t)
[1]  TRUE FALSE  TRUE  TRUE
> v <- c(3,0,TRUE,2+2i)
> print(!v)
[1] FALSE  TRUE FALSE FALSE
> v <- c(3,0,TRUE,2+2i)
> t <- c(1,3,TRUE,2+3i)
> print(v&&t)
[1] TRUE
> v <- c(0,0,TRUE,2+2i)
> t <- c(0,3,TRUE,2+3i)
> print(v||t)
[1] FALSE
> |
```

- Assignment Operators

```
> v1 <- c(3,1,TRUE,2+3i)
> v2 <<- c(3,1,TRUE,2+3i)
> v3 = c(3,1,TRUE,2+3i)
> print(v1)
[1] 3+0i 1+0i 1+0i 2+3i
> print(v2)
[1] 3+0i 1+0i 1+0i 2+3i
> print(v3)
[1] 3+0i 1+0i 1+0i 2+3i
> c(3,1,TRUE,2+3i) -> v1
> c(3,1,TRUE,2+3i) ->> v2
> print(v1)
[1] 3+0i 1+0i 1+0i 2+3i
> print(v2)
[1] 3+0i 1+0i 1+0i 2+3i
>
```

- Miscellaneous Operators

```
> v <- 2:8
> print(v)
[1] 2 3 4 5 6 7 8
> v1 <- 8
> v2 <- 12
> t <- 1:10
> print(v1 %in% t)
[1] TRUE
> print(v2 %in% t)
[1] FALSE
> M = matrix( c(2,6,5,1,10,4), nrow=2,ncol=3,byrow =
+ TRUE)
> t = M %*% t(M)
> print(t)
     [,1] [,2]
[1,]   65   82
[2,]   82  117
>
```

- Decision Making
  - If statement

```
> x <- 30L
> if(is.integer(x)){
+ print("X is an Integer")
+ }
[1] "X is an Integer"
```

  - If else statement

```
> x <- c("what","is","truth")
> if("Truth" %in% x){
+ print("Truth is found")
+ } else {
+ print("Truth is not found")
+ }
[1] "Truth is not found"
```

  - If else if else if else statement

```
> x <- c("what","is","truth")
> if("Truth" %in% x){
+ print("Truth is found the first time")
+ } else if ("truth" %in% x) {
+ print("truth is found the second time")
+ } else {
+ print("No truth found")
+ }
[1] "truth is found the second time"
```

  - Switch statement

```
> x <- switch(
+ 3,
+ "first",
+ "second",
+ "third",
+ "fourth"
+ )
> print(x)
[1] "third"
> |
```

- Loops
  - Repeat

```
> v <- c("Hello","loop")
> cnt <- 2
> repeat{
+ print(v)
+ cnt <- cnt+1
+ if(cnt > 5){
+ break
+ }
+ }
[1] "Hello" "loop"
[1] "Hello" "loop"
[1] "Hello" "loop"
[1] "Hello" "loop"
>
```

  - While

```
> v <- c("Hello","while loop")
> cnt <- 2
> while (cnt < 7){
+ print(v)
+ cnt = cnt + 1
+ }
[1] "Hello"      "while loop"
[1] "Hello"      "while loop"
[1] "Hello"      "while loop"
[1] "Hello"      "while loop"
[1] "Hello"      "while loop"
>
```

  - For

```
> v <- LETTERS[1:4]
> for ( i in v) {
+ print(i)
+ }
[1] "A"
[1] "B"
[1] "C"
[1] "D"
>
```

- Functions

- Built in function

```
> # Create a sequence of numbers from 32 to 44.
> print(seq(32,44))
 [1] 32 33 34 35 36 37 38 39 40 41 42 43 44
>
> # Find mean of numbers from 25 to 82.
> print(mean(25:82))
[1] 53.5
>
> # Find sum of numbers frm 41 to 68.
> print(sum(41:68))
[1] 1526
```

- User defined function

```
> new.function <- function(a) {
+ for(i in 1:a) {
+ b <- i^2
+ print(b)
+ }
+ }
> new.function(6)
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
```

- Calling a function

```
> new.function <- function() {
+ for(i in 1:5) {
+ print(i^2)
+ }
+ }
> new.function()
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

- String

```
> myString <- "Hello, World!"
> print ( myString)
[1] "Hello, World!"
```

- Vector

```
> apple <- c('red','green',"yellow")
> print(apple)
[1] "red"    "green"  "yellow"
> print(class(apple))
[1] "character"
>
```

- Matrices

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

- Lists

```
> list1 <- list(c(2,5,3),21.3,sin)
> print(list1)
[[1]]
[1] 2 5 3

[[2]]
[1] 21.3

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

- Arrays

```
> a <- array(c('green','yellow'),dim=c(3,3,2))
> print(a)
, , 1

     [,1]     [,2]     [,3]
[1,] "green"  "yellow" "green"
[2,] "yellow" "green"  "yellow"
[3,] "green"  "yellow" "green"

, , 2

     [,1]     [,2]     [,3]
[1,] "yellow" "green"  "yellow"
[2,] "green"  "yellow" "green"
[3,] "yellow" "green"  "yellow"
```

- Data Frames

```
> BMI <- data.frame(
+
+ gender = c("Male", "Male","Female"),height = c(152, 171.5, 165),
+ weight = c(81,93, 78),
+ Age =c(42,38,26)
+ )
> print(BMI)
  gender height weight Age
1   Male  152.0     81  42
2   Male  171.5     93  38
3 Female  165.0     78  26
>
```

- Factors

```
> apple_colors <- c('green','green','yellow','red','red','red','green')
> factor_apple <- factor(apple_colors)
> print(factor_apple)
[1] green  green  yellow red    red    red    green
Levels: green red yellow
>
```

Practical 3

Aim - Consider dataset, for rest of the Practical, with student name, gender, Enrollmentno, 4 semester result with marks of each subject, his mobile number, city.
Perform descriptive analysis and identify the data type.

R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"

Platform: x86_64-w64-mingw32/x64 (64-bit)

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

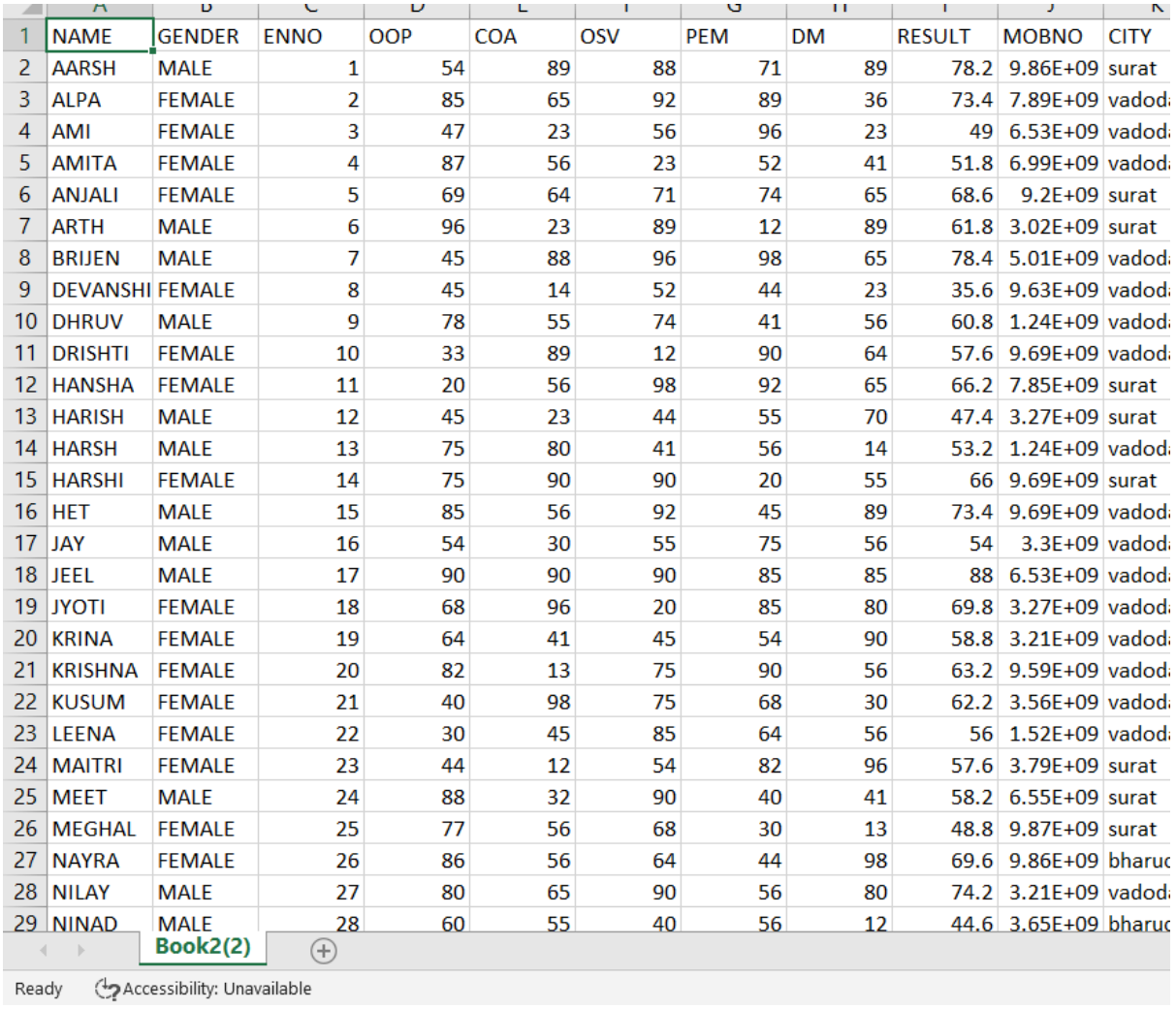
'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

```
> print("data")


[1] "data"

> getwd()
```

"C:/Users/Harshil/Documents" data<-read.csv("SYIT.csv")

> print(data)



	NAME	GENDER	ENNO	OOP	COA	OSV	PEM	DM	RESULT	MOBNO	CITY
2	AARSH	MALE	1	54	89	88	71	89	78.2	9.86E+09	surat
3	ALPA	FEMALE	2	85	65	92	89	36	73.4	7.89E+09	vadod
4	AMI	FEMALE	3	47	23	56	96	23	49	6.53E+09	vadod
5	AMITA	FEMALE	4	87	56	23	52	41	51.8	6.99E+09	vadod
6	ANJALI	FEMALE	5	69	64	71	74	65	68.6	9.2E+09	surat
7	ARTH	MALE	6	96	23	89	12	89	61.8	3.02E+09	surat
8	BRIJEN	MALE	7	45	88	96	98	65	78.4	5.01E+09	vadod
9	DEVANSHI	FEMALE	8	45	14	52	44	23	35.6	9.63E+09	vadod
10	DHRUV	MALE	9	78	55	74	41	56	60.8	1.24E+09	vadod
11	DRISHTI	FEMALE	10	33	89	12	90	64	57.6	9.69E+09	vadod
12	HANSHA	FEMALE	11	20	56	98	92	65	66.2	7.85E+09	surat
13	HARISH	MALE	12	45	23	44	55	70	47.4	3.27E+09	surat
14	HARSH	MALE	13	75	80	41	56	14	53.2	1.24E+09	vadod
15	HARSHI	FEMALE	14	75	90	90	20	55	66	9.69E+09	surat
16	HET	MALE	15	85	56	92	45	89	73.4	9.69E+09	vadod
17	JAY	MALE	16	54	30	55	75	56	54	3.3E+09	vadod
18	JEEL	MALE	17	90	90	90	85	85	88	6.53E+09	vadod
19	JYOTI	FEMALE	18	68	96	20	85	80	69.8	3.27E+09	vadod
20	KRINA	FEMALE	19	64	41	45	54	90	58.8	3.21E+09	vadod
21	KRISHNA	FEMALE	20	82	13	75	90	56	63.2	9.59E+09	vadod
22	KUSUM	FEMALE	21	40	98	75	68	30	62.2	3.56E+09	vadod
23	LEENA	FEMALE	22	30	45	85	64	56	56	1.52E+09	vadod
24	MAITRI	FEMALE	23	44	12	54	82	96	57.6	3.79E+09	surat
25	MEET	MALE	24	88	32	90	40	41	58.2	6.55E+09	surat
26	MEGHAL	FEMALE	25	77	56	68	30	13	48.8	9.87E+09	surat
27	NAYRA	FEMALE	26	86	56	64	44	98	69.6	9.86E+09	bharuc
28	NILAY	MALE	27	80	65	90	56	80	74.2	3.21E+09	vadod
29	NINAD	MALE	28	60	55	40	56	12	44.6	3.65E+09	bharuc

Book2(2)

Ready Accessibility: Unavailable

> data<-read.csv("Book2.csv")

> print(data)

	NAME	GENDER	ENNO	OOP	COA	OSV	PEM	DM	RESULT	MOBNO	CITY
1	AARSH	MALE	1	54	89	88	71	89	78.2	9856321456	surat
2	ALPA	FEMALE	2	85	65	92	89	36	73.4	7894561230	vadodara
3	AMI	FEMALE	3	47	23	56	96	23	49.0	6532147890	vadodara
4	AMITA	FEMALE	4	87	56	23	52	41	51.8	6987451230	vadodara
5	ANJALI	FEMALE	5	69	64	71	74	65	68.6	9201456320	surat

6 ARTH MALE 6 96 23 89 12 89 61.8 3021045632 surat
7 BRIJEN MALE 7 45 88 96 98 65 78.4 5010203045 vadodara
8 DEVANSHI FEMALE 8 45 14 52 44 23 35.6 9632560120 vadodara
9 DHRUV MALE 9 78 55 74 41 56 60.8 1236050489 vadodara
10 DRISHTI FEMALE 10 33 89 12 90 64 57.6 9685741230 vadodara
11 HANSHA FEMALE 11 20 56 98 92 65 66.2 7845123690 surat
12 HARISH MALE 12 45 23 44 55 70 47.4 3265987410 surat
13 HARSH MALE 13 75 80 41 56 14 53.2 1236547890 vadodara
14 HARSHI FEMALE 14 75 90 90 20 55 66.0 9685741230 surat
15 HET MALE 15 85 56 92 45 89 73.4 9685742130 vadodara
16 JAY MALE 16 54 30 55 75 56 54.0 3298745610 vadodara
17 JEEL MALE 17 90 90 90 85 85 88.0 6532149870 vadodara
18 JYOTI FEMALE 18 68 96 20 85 80 69.8 3265987410 vadodara
19 KRINA FEMALE 19 64 41 45 54 90 58.8 3214569870 vadodara
20 KRISHNA FEMALE 20 82 13 75 90 56 63.2 9586741230 vadodara
21 KUSUM FEMALE 21 40 98 75 68 30 62.2 3562148970 vadodara
22 LEENA FEMALE 22 30 45 85 64 56 56.0 1524369870 vadodara
23 MAITRI FEMALE 23 44 12 54 82 96 57.6 3791826450 surat
24 MEET MALE 24 88 32 90 40 41 58.2 6548237910 surat
25 MEGHAL FEMALE 25 77 56 68 30 13 48.8 9873214560 surat
26 NAYRA FEMALE 26 86 56 64 44 98 69.6 9856321470 bharuch
27 NILAY MALE 27 80 65 90 56 80 74.2 3210654789 vadodara
28 NINAD MALE 28 60 55 40 56 12 44.6 3652147890 bharuch
29 NISHTHA FEMALE 29 74 56 30 65 32 51.4 1166655778 bharuch
30 OM MALE 30 57 89 44 55 56 60.2 2365419870 bharuch
31 PALAK FEMALE 31 54 45 88 56 56 59.8 6985471230 vadodara
32 PALLAVI FEMALE 32 30 67 77 89 65 65.6 1236540902 vadodara
33 PANKTI FEMALE 33 80 78 86 45 55 68.8 3021090403 vadodara
34 PARTH MALE 34 80 46 90 67 56 67.8 2698403040 vadodara
35 PRINCE MALE 35 42 55 92 78 89 71.2 3698521478 vadodara
36 PRITI FEMALE 36 44 64 55 46 64 54.6 2354556478 vadodara

```
37    PRIYAL FEMALE  37  50  56  56  55 82   59.8 4563214555       vadodara
38     RAMESH  MALE   38  66  32  12  64 40   42.8 1388965321      vadodara
39     RITUL   MALE   39  77  98  85  56 30  69.2 4268897720       vadodara
40    SAHDEV  MALE   40  66  32  41  32 44   43.0 1532010403       bharuch
41     SHIKHA FEMALE  41  77  33  95  98 88   78.2 4653214563     ahemdabad
42    SHRUSTI FEMALE  42  45  65  95  32 77   62.8 1236504109     ahemdabad
43     SOHAM  MALE   43  90  32  90  33 86   66.2 1532669988       vadodara
44     TUSAR   MALE   44  41  32  55  65 90  56.6 9865774423       vadodara
45      URVI FEMALE  45  44  31  51  32 92   50.0 9874563201       vadodara
46    VAIDEHI FEMALE  46  78  31  11  32 55   41.4 6853120321      vadodara
47 VAISHNAVI FEMALE   47  45  98  59  31 56   57.8 5897856466      vadodara
48    VEDANT   MALE   48  22  74  60  31 12   39.8 5689745620      vadodara
49    VISHWA FEMALE   49  78  54  70  98 85   77.0 1546328940      vadodara
50     VRAJ   MALE   50  52 100  40  74 41   61.4 1236547890      vadodara
> dim(data)
[1] 50 11
> class(data)
[1] "data.frame"
> head(data)
   NAME GENDER ENNO OOP COA OSV PEM DM RESULT      MOBNO     CITY
1  AARSH  MALE     1 54  89  88  71 89    78.2 9856321456    surat
2  ALPA FEMALE     2 85  65  92  89 36    73.4 7894561230 vadodara
3    AMI FEMALE    3 47  23  56  96 23    49.0 6532147890 vadodara
4  AMITA FEMALE    4  87  56  23  52 41  51.8 6987451230 vadodar a
5 ANJALI FEMALE    5  69  64  71  74 65   68.6 9201456320    surat
6   ARTH  MALE     6  96  23  89  12 89  61.8 3021045632     surat
> ncol(data)
[1] 11
> nrow(data)
[1] 50
> colnames(data)
```

```
 [1] "NAME"   "GENDER" "ENNO"   "OOP"    "COA"    "OSV"    "PEM"    "DM"     "RESULT"
"MOBNO"  "CITY"
> print(data$NAME)
 [1] "AARSH"    "ALPA"     "AMI"      "AMITA"    "ANJALI"   "ARTH"     "BRIJEN"
"DEVANSHI"  "DHRUV"    "DRISHTI"  "HANSHA"   "HARISH"   "HARSH"    "HARSHI"
"HET"
[16] "JAY"      "JEEL"     "JYOTI"    "KRINA"    "KRISHNA"  "KUSUM"    "LEENA"
"MAITRI"   "MEET"     "MEGHAL"   "NAYRA"    "NILAY"    "NINAD"    "NISHTHA"  "OM"
[31] "PALAK"    "PALLAVI"  "PANKTI"   "PARTH"    "PRINCE"   "PRITI"    "PRIYAL"
"RAMESH"   "RITUL"    "SAHDEV"   "SHIKHA"   "SHRUSTI"  "SOHAM"    "TUSAR"
"URVI"
[46] "VAIDEHI"   "VAISHNAVI" "VEDANT"   "VISHWA"   "VRAJ"
> print(data$RESULT)
 [1] 78.2 73.4 49.0 51.8 68.6 61.8 78.4 35.6 60.8 57.6 66.2 47.4 53.2 66.0 73.4 54.0 88.0
69.8 58.8 63.2 62.2 56.0 57.6 58.2 48.8 69.6 74.2 44.6 51.4 60.2 59.8 65.6 68.8 67.8
71.2 54.6
[37] 59.8 42.8 69.2 43.0 78.2 62.8 66.2 56.6 50.0 41.4 57.8 39.8 77.0 61.4
>  print(class(data$NAME))
[1] "character"
> print(class(data$RESULT))
[1] "numeric"
>  print(class(data$OOP))
[1] "integer"
> print(class(data$COA))
[1] "integer"
> print(class(data$OSV))
[1] "integer"
>  print(class(data$PEM))
[1] "integer"
>  print(class(data$DM))
[1] "integer"
>  print(class(data$CITY))
```

```
[1] "character"
> print(class(data$MOBNO))
[1] "numeric"
```

Practical 4

Aim - Implement a method to find out variation in data. For example the difference between highest and lowest marks in each subject semester wise.

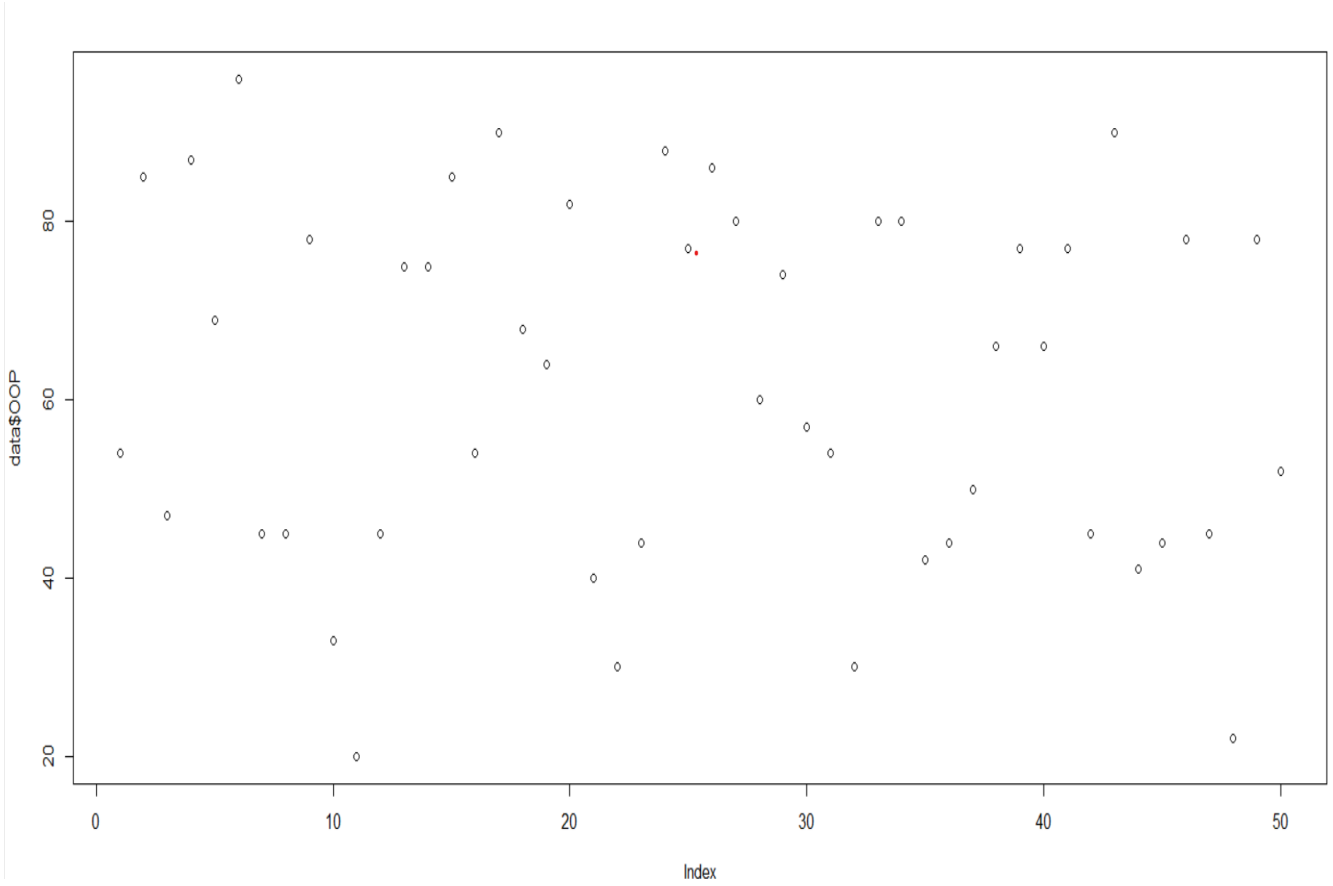
```
>  print(mean(data$OOP))
[1] 61.88
> print(mean(data$COA))
[1] 56.56
> print(mean(data$PEM))
[1] 59.96
> print(mean(data$DM))
[1] 59.76
> print(mean(data$RESULT))
[1] 60.636
>  print(median(data$OOP))
[1] 65
>  print(median(data$COA))
[1] 56
>  print(median(data$PEM))
[1] 56
>  print(median(data$DM))
[1] 56
```

```
>  print(median(data$RESULT))

[1] 60.5

> getmode<-function(v){
+ uniqv<-unique(v)
+ uniqv[which.max(tabulate(match(v,uniqv)))]}

> print(getmode(data$RESULT))
[1] 78.2
> print(getmode(data$OOP))
[1] 45
> print(getmode(data$COA))
[1] 56
> print(getmode(data$PEM))
[1] 56
> print(getmode(data$DM))
[1] 56
> print(getmode(data$OSV))
[1] 90
> print(getmode(data$CITY))
[1] "vadodara"

> quantile(data$OOP)
  0%  25% 50%  75% 100%
  20  45   65   78   96

> quantile(data$COA)
  0%  25% 50%  75% 100%
  12   32   56   77  100
> quantile(data$OSV)
   0%   25%  50%   75%  100%
11.00 46.50 69.00 89.75 98.00
```

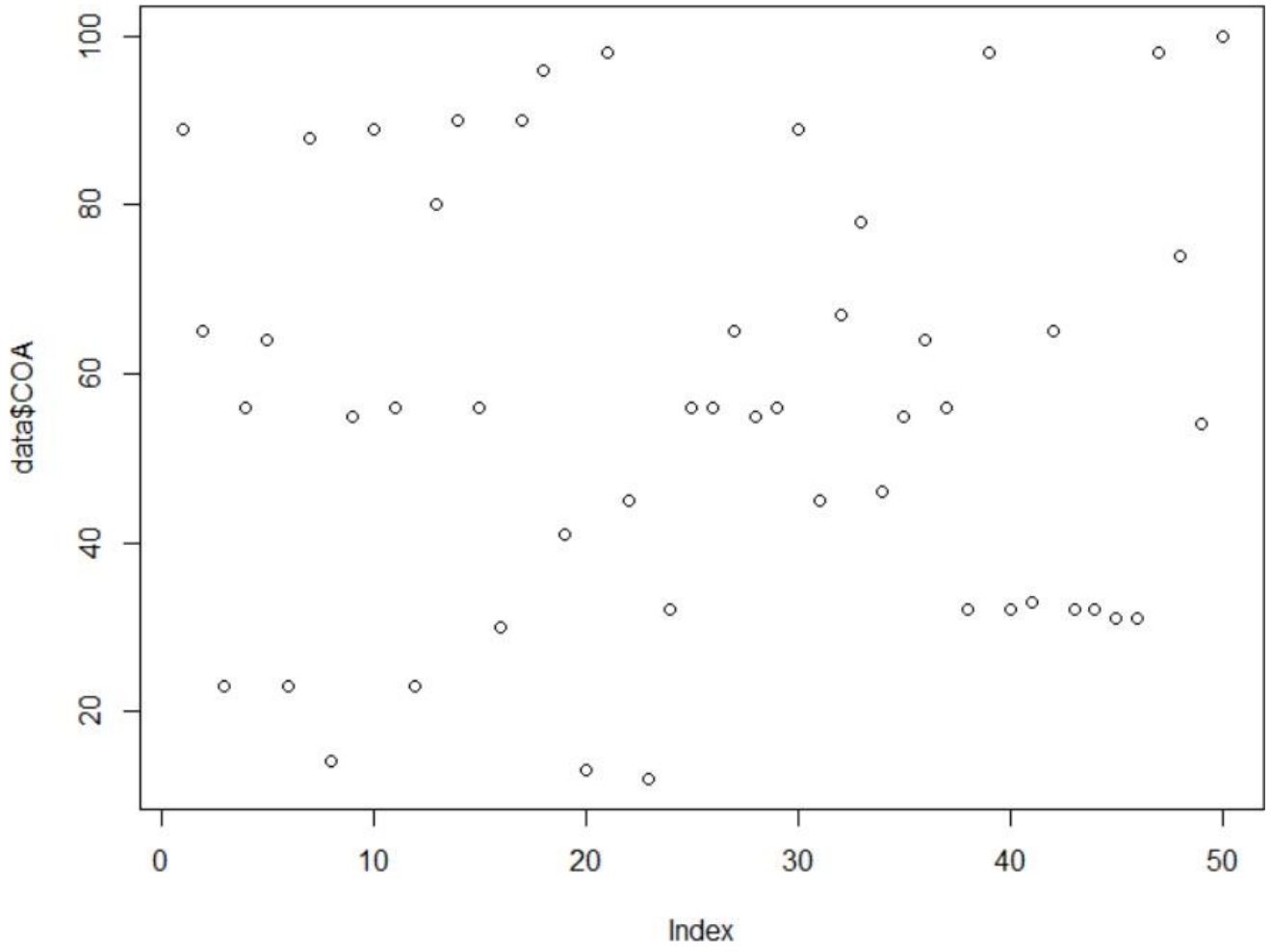
```
> quantile(data$PEM)
  0%  25%  50%  75%  100%
12.00 44.00 56.00 77.25 98.00

> quantile(data$DM)
  0%  25%  50%  75%  100%
12.00 41.00 56.00 84.25 98.00

> quantile(data$RESULT)
  0%  25%  50%  75%  100%
35.60 53.40 60.50 68.75 88.00

> quantile(data$OOP,probs=c(0.1,0.2,0.3,0.4,0.5))
 10%  20%  30%  40% 50%
39.3 44.0 45.0 54.0 65.0

> quantile(data$COA,probs=c(0.1,0.2,0.3,0.4,0.5))
 10%  20%  30%  40% 50%
23.0 32.0 38.6 54.6 56.0

> quantile(data$OSV,probs=c(0.1,0.2,0.3,0.4,0.5))
 10%  20%  30%  40% 50%
29.3 43.4 53.4 56.0 69.0

> quantile(data$PEM,probs=c(0.1,0.2,0.3,0.4,0.5))
 10%  20%  30%  40% 50%
31.9 38.6 45.0 55.0 56.0


> quantile(data$DM,probs=c(0.1,0.2,0.3,0.4,0.5))
 10%  20%  30%  40% 50%
23.0 39.2 51.7 56.0 56.0
```

```
> quantile(data$RESULT,probs=c(0.1,0.2,0.3,0.4,0.5))
  10%  20%  30%  40%  50%
44.44 51.12 55.58 58.04 60.50

> quantile(data$OOP,0.25)
25%
 45
> quantile(data$OOP,0.5)
50%
 65
> quantile(data$OOP,0.75)
75%
 78
> quantile(data$COA,0.25)
25%
 32
> quantile(data$COA,0.5)
50%
 56
> quantile(data$COA,0.75)
75%
 77
> quantile(data$OSV,0.25)
 25%
46.5
> quantile(data$OSV,0.5)
50%
 69
> quantile(data$OSV,0.75)
  75%
89.75
> quantile(data$PEM,0.25)
```

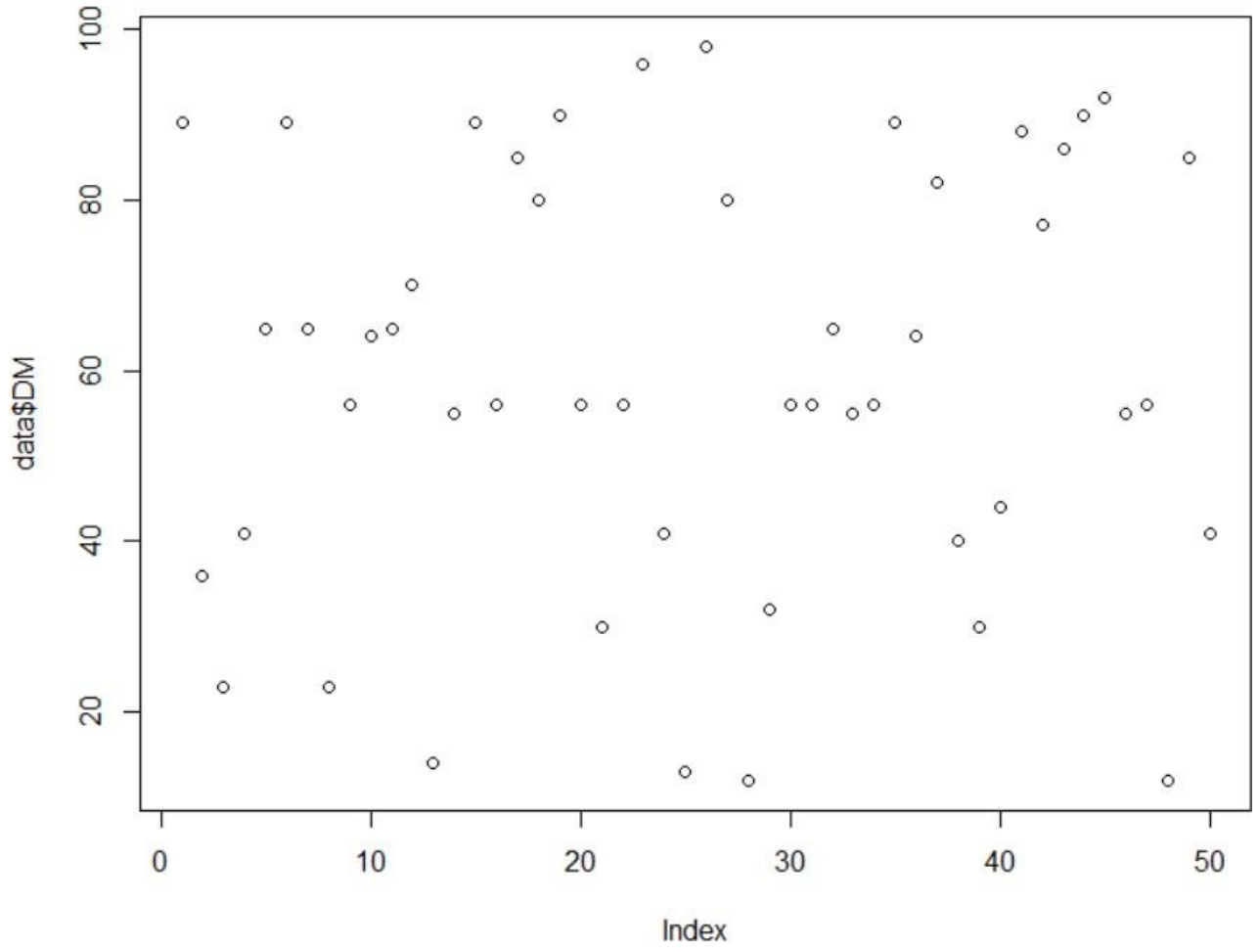
```
25%
 44
> quantile(data$PEM,0.5)
50%
 56
> quantile(data$PEM,0.75)
  75%
77.25

> quantile(data$DM,0.25)
25%
 41
> quantile(data$DM,0.5)
50%
 56
> quantile(data$DM,0.75)
  75%
84.25
> quantile(data$RESULT,0.25)
 25%
53.4
> quantile(data$RESULT,0.5)
 50%
60.5
> quantile(data$RESULT,0.75)
  75%
68.75

> plot(data$OOP)
```



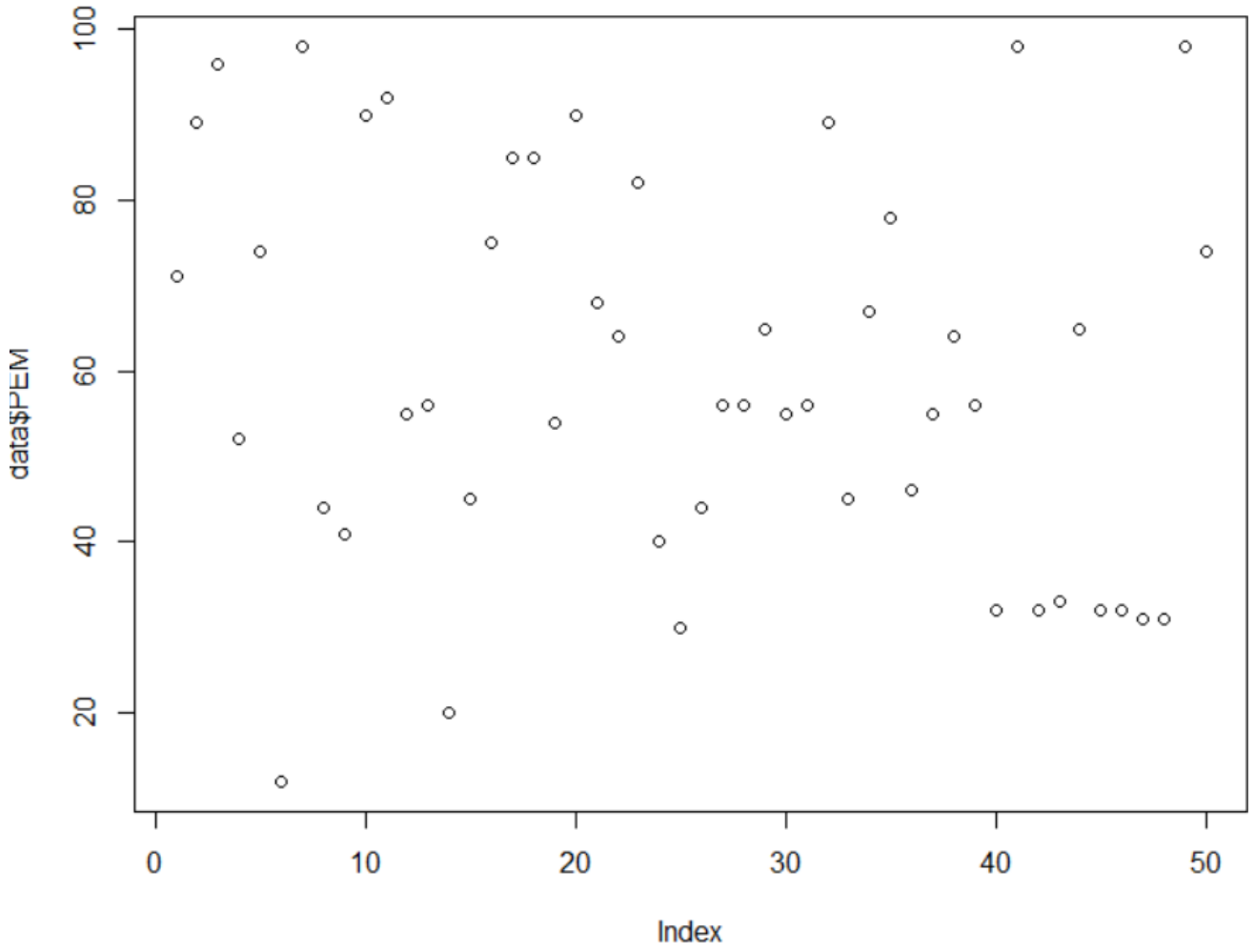
>plot(data$COA)



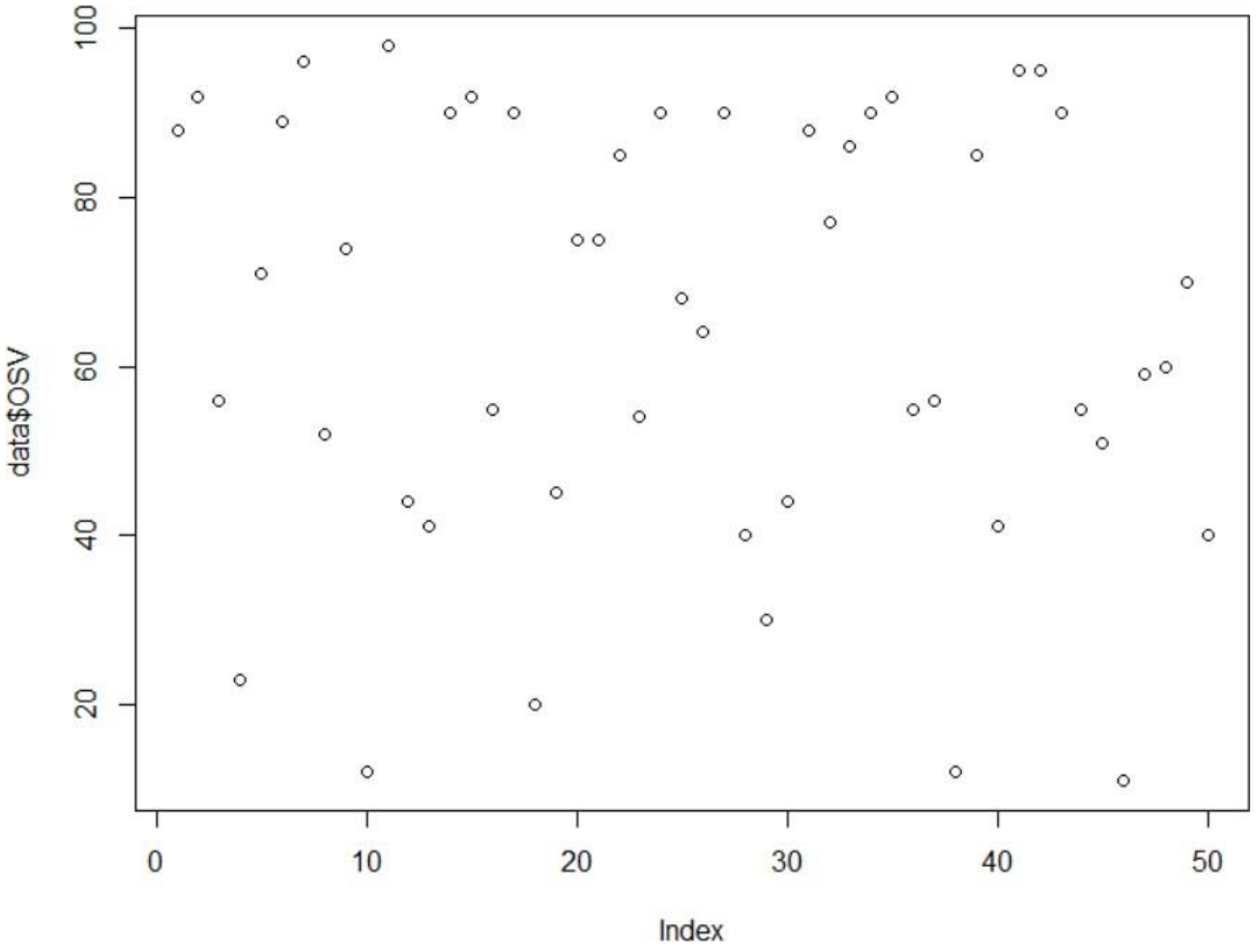
> plot(data$DM)



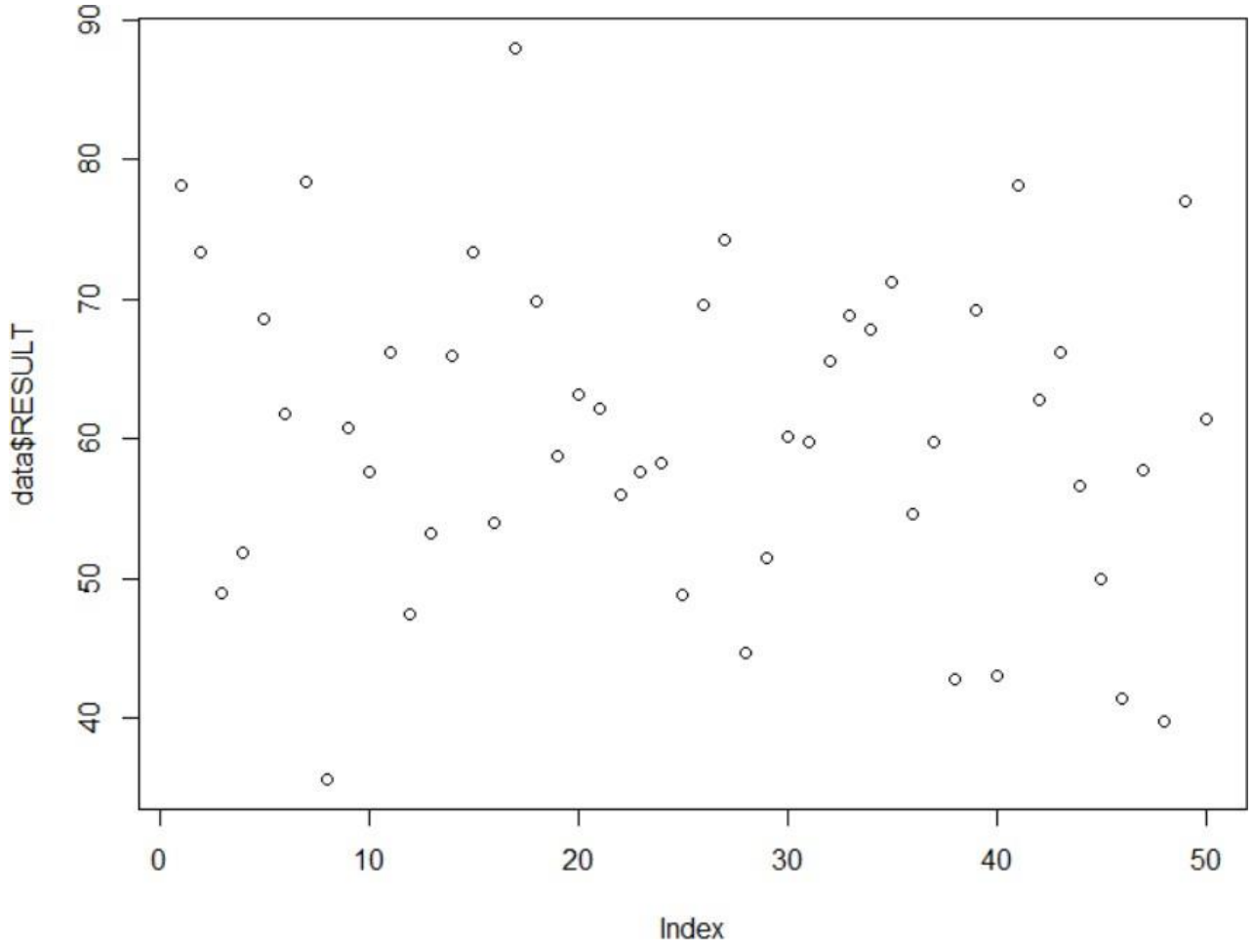
> plot(data$PEM)



>plot(data$OSV)




```
> plot(data$RESULT)
```


data$RESULT
90
80
70
60
50
40
0
10
20
30
40
50
Index

```
var(data$OOP)
[1] 403.2914
> var(data$COA)
[1] 653.3943
> var(data$DM)
[1] 616.5943
> var(data$PEM)
[1] 523.4678
> var(data$OSV)
[1] 640.7955
> var(data$RESULT)
[1] 130.1219
> sd(data$OOP)
[1] 20.08212
> sd(data$COA)
[1] 25.56158
[1]
>
```

```
> sd(data$DM)

[1] 24.83132

> sd(data$PEM)

[1] 22.87942

> sd(data$OSV)

[1] 25.31394

> sd(data$RESULT)

[1] 11.4071

> min(data$OOP)

[1] 20

> min(data$COA)

[1] 12

> min(data$PEM)

[1] 12

> min(data$DM)

[1] 12

> min(data$OSV)

[1] 11

> min(data$RESULT)

[1] 35.6
[1]

>
```

```
> max(data$OOP)
[1] 96
> max(data$COA)
[1] 100
> max(data$PEM)
[1] 98
> max(data$DM)
[1] 98
> max(data$OSV)
[1] 98
> max(data$RESULT)
[1] 88
> range(data$OOP)
[1] 20 96
> range(data$COA)
[1]  12 100
> range(data$PEM)
[1] 12 98
> range(data$DM)
[1] 12 98
```

```
> range(data$OSV)

[1] 11 98

> range(data$RESULT)

[1] 35.6 88.0

> IQR(data$OOP)

[1] 33

> IQR(data$COA)

[1] 45

> IQR(data$PEM)

[1] 33.25

> IQR(data$DM)

[1] 43.25

> IQR(data$OSV)

[1] 43.25

> IQR(data$RESULT)

[1] 15.35

> subset(data,NAME=="ALPA")
  NAME GENDER ENNO OOP COA OSV PEM DM RESULT      MOBNO     CITY
2 ALPA FEMALE    2  85  65  92  89 36  73.4 7894561230 vadodara

> subset(data,CITY=="SURAT")

[1]

>
```

	NAME	GENDER	ENNO	OOP	COA	OSV	PEM	DM	RESULT	MOBNO	CITY
1	AARSH	MALE	1	54	89	88	71	89	78.2	9856321456	surat
5	ANJALI	FEMALE	5	69	64	71	74	65	68.6	9201456320	surat
6	ARTH	MALE	6	96	23	89	12	89	61.8	3021045632	surat
11	HANSHA	FEMALE	11	20	56	98	92	65	66.2	7845123690	surat
12	HARISH	MALE	12	45	23	44	55	70	47.4	3265987410	surat
14	HARSHI	FEMALE	14	75	90	90	20	55	66.0	9685741230	surat
23	MAITRI	FEMALE	23	44	12	54	82	96	57.6	3791826450	surat
24	MEET	MALE	24	88	32	90	40	41	58.2	6548237910	surat
25	MEGHAL	FEMALE	25	77	56	68	30	13	48.8	9873214560	surat

TYIT- 2-C

```
> subset(data,GENDER=="FEMALE")
      NAME GENDER ENNO OOP COA OSV PEM DM RESULT      MOBNO      CITY
2     ALPA FEMALE    2  85  65  92  89 36  73.4 7894561230  vadodara
3      AMI FEMALE    3  47  23  56  96 23  49.0 6532147890  vadodara
4    AMITA FEMALE    4  87  56  23  52 41  51.8 6987451230  vadodara
5   ANJALI FEMALE    5  69  64  71  74 65   68.6 9201456320     surat
8  DEVANSHI FEMALE    8  45  14  52  44 23  35.6 9632560120  vadodara
10  DRISHTI FEMALE  10  33  89  12  90 64   57.6 9685741230  vadodara
11   HANSHA FEMALE   11  20  56  98  92 65   66.2 7845123690     surat
14   HARSHI FEMALE   14  75  90  90  20 55   66.0 9685741230     surat
18    JYOTI FEMALE  18  68  96  20  85 80   69.8 3265987410  vadodara
19    KRINA FEMALE  19  64  41  45  54 90   58.8 3214569870  vadodara
20  KRISHNA FEMALE  20  82  13  75  90 56   63.2 9586741230  vadodara
21    KUSUM FEMALE  21  40  98  75  68 30   62.2 3562148970  vadodara
22    LEENA FEMALE  22  30  45  85  64 56   56.0 1524369870  vadodara
23   MAITRI FEMALE   23  44  12  54  82 96   57.6 3791826450     surat
25   MEGHAL FEMALE   25  77  56  68  30 13   48.8 9873214560     surat
26    NAYRA FEMALE  26  86  56  64  44 98   69.6 9856321470   bharuch
29  NISHTHA FEMALE  29  74  56  30  65 32   51.4 1166655778   bharuch
31    PALAK FEMALE   31  54  45  88  56 56   59.8 6985471230  vadodara
32 PALLAVI FEMALE    32  30  67  77  89 65   65.6 1236540902  vadodara
33   PANKTI FEMALE   33  80  78  86  45 55   68.8 3021090403  vadodara
36    PRITI FEMALE  36  44  64  55  46 64   54.6 2354556478  vadodara
37   PRIYAL FEMALE  37  50  56  56  55 82   59.8 4563214555  vadodara
41   SHIKHA FEMALE  41  77  33  95  98 88   78.2 4653214563 ahemdabad
42  SHRUSTI FEMALE  42  45  65  95  32 77   62.8 1236504109 ahemdabad
45     URVI FEMALE  45  44  31  51  32 92   50.0 9874563201  vadodara
46  VAIDEHI FEMALE  46  78  31  11  32 55   41.4 6853120321  vadodara
47 VAISHNAVI FEMALE  47  45  98  59  31 56   57.8 5897856466  vadodara
```

```
49    VISHWA FEMALE  49  78  54  70  98 85   77.0 1546328940  vadodara

> subset(data,ENNO=="5")
   NAME GENDER ENNO OOP COA OSV PEM DM RESULT      MOBNO CITY
5 ANJALI FEMALE    5  69  64  71  74 65  68.6 9201456320 surat
> print(max(data$OOP))
[1] 96
> print(min(data$OOP))
[1] 20
> m1=min(data$OOP)
> m2=max(data$OOP)
> print(m2-m1)
[1] 76
> print(range(data$OOP))
[1] 20 96
> print(min(data$COA))
[1] 12
> print(max(data$COA))
[1] 100
> m2=max(data$COA)
> m1=min(data$COA)
```

```
> print(range(data$COA))
[1]  12 100
> print(m2-m1)
[1] 88
> print(min(data$DM))
[1] 12
> print(max(data$DM))
[1] 98
> m1=min(data$DM)
> m2=max(data$DM)
> print(m2-m1)
[1] 86
> print(range(data$DM))
[1] 12 98
> print(min(data$PEM))
[1] 12
> print(max(data$PEM))
[1] 98
> m2=max(data$PEM)
> m1=min(data$PEM)
> print(m2-m1)
[1] 86
```

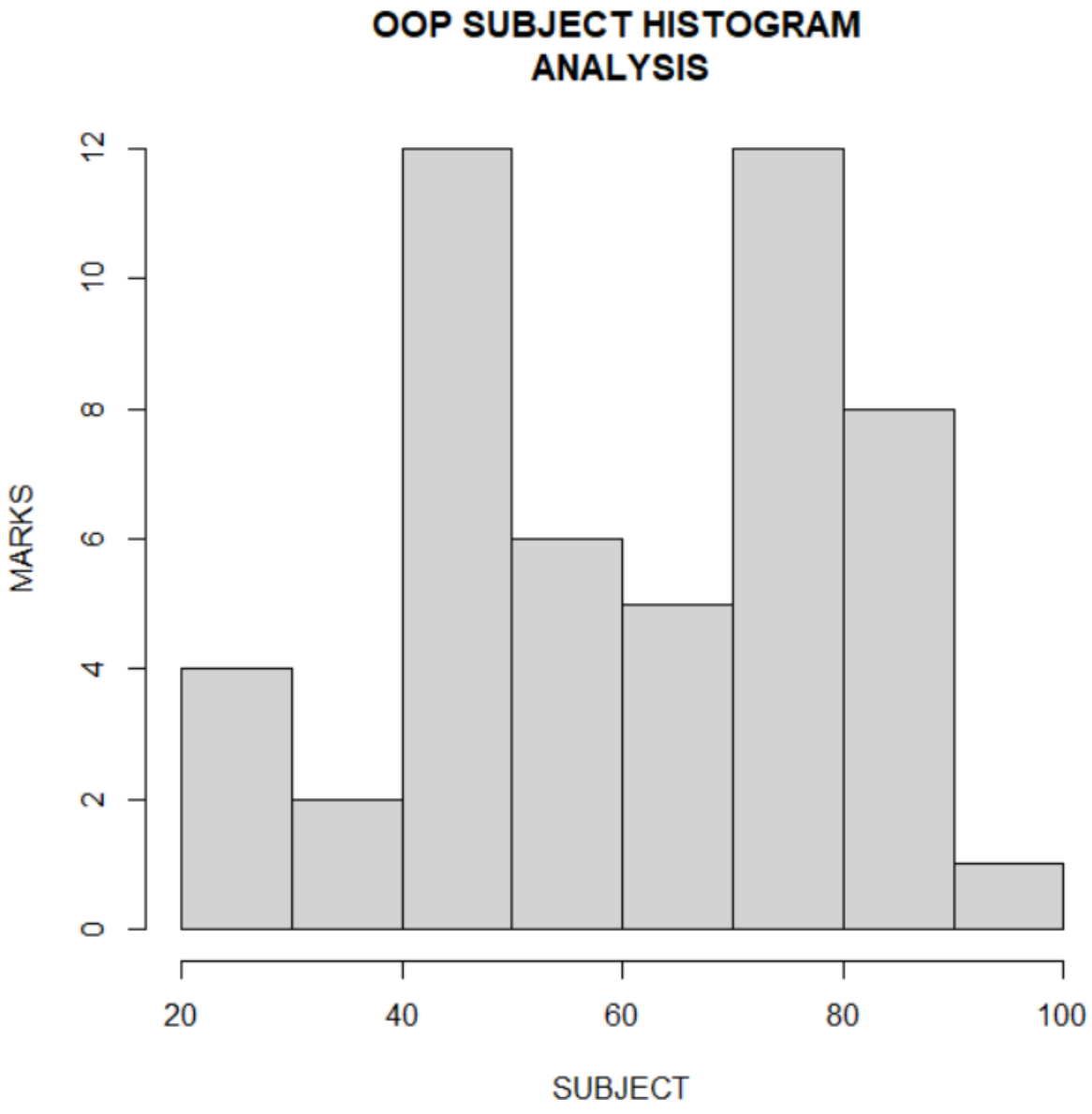
```
> print(range(data$PEM))
[1] 12 98
>  print(min(data$OSV))
[1] 11
> print(max(data$OSV))
[1] 98
> m2=max(data$OSV)
> m1=min(data$OSV)
> print(m2-m1)
[1] 87
> print(range(data$OSV))
[1] 11 98
> print(min(data$RESULT))
[1] 35.6
> print(max(data$RESULT))
[1] 88
> m2=max(data$RESULT)
> m1=min(data$RESULT)
> print(m2-m1)
[1] 52.4
```

```
> print(range(data$RESULT))

[1] 35.6 88.0
```

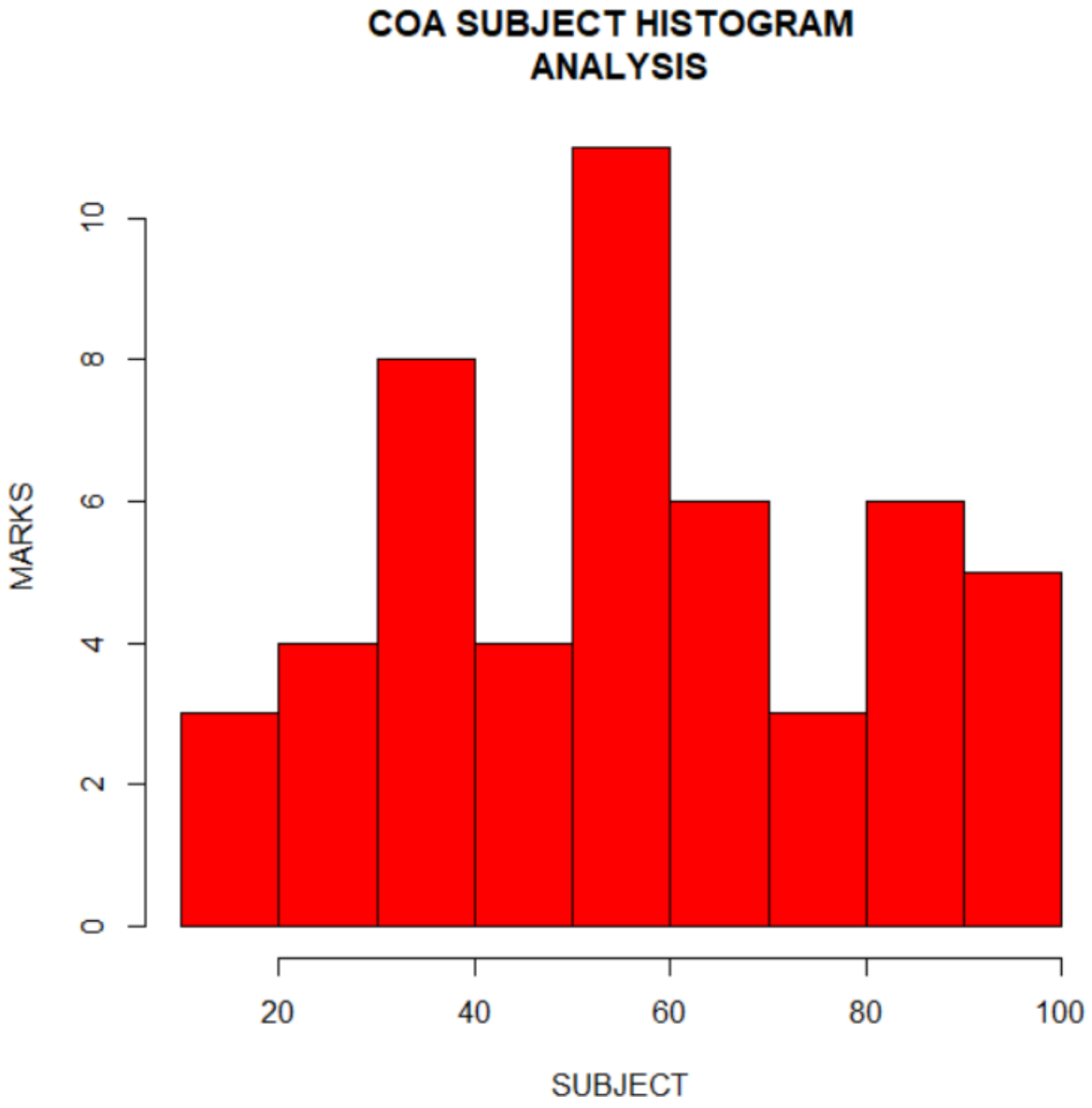
Practical 5

Aim - Plot the graph showing result of student in each semester.

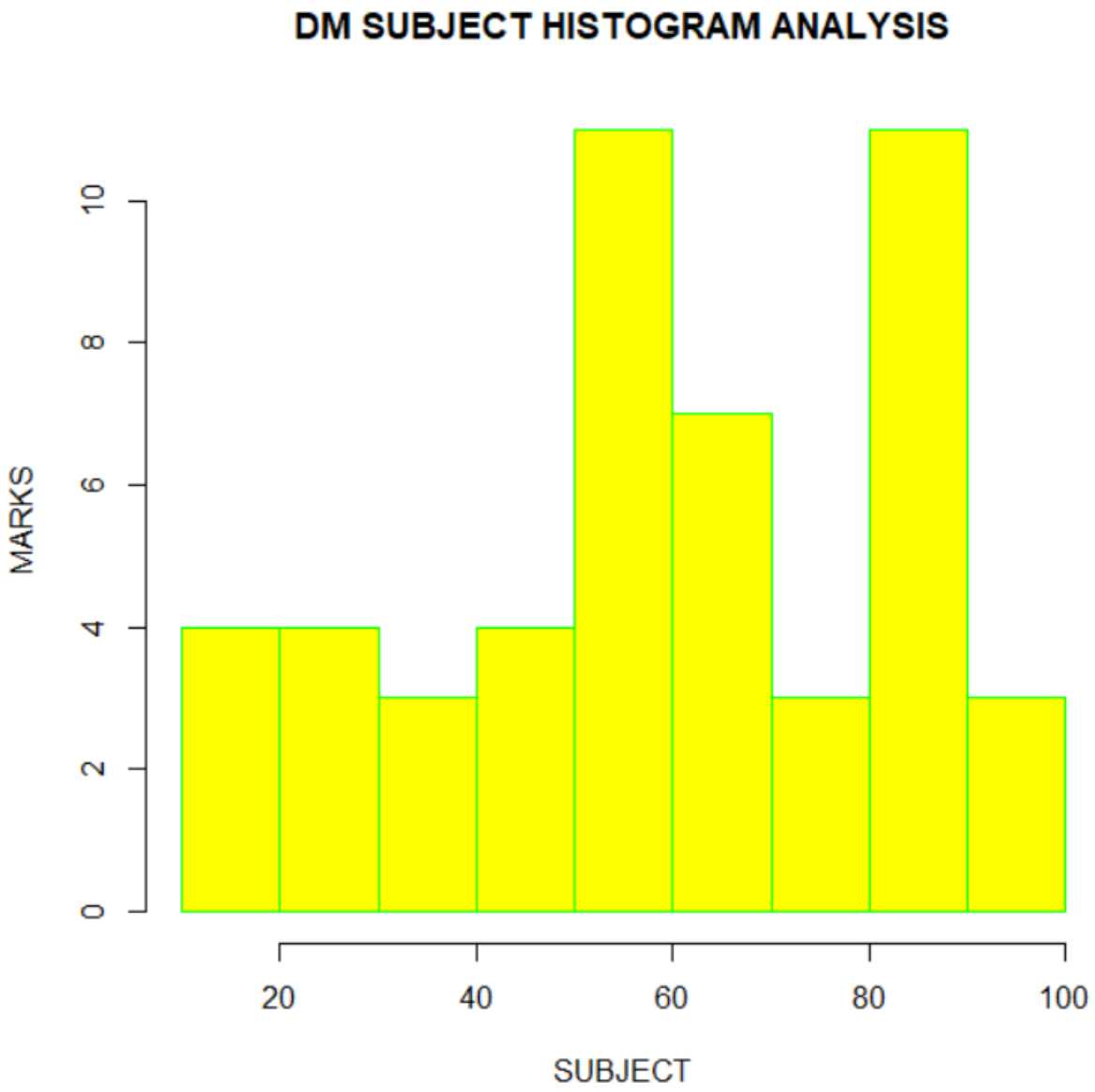
```
>hist(data$OOP,breaks=10,main="OOP SUBJECT HISTOGRAM ANALYSIS",xlab="SUBJECT",ylab="MARKS")
```



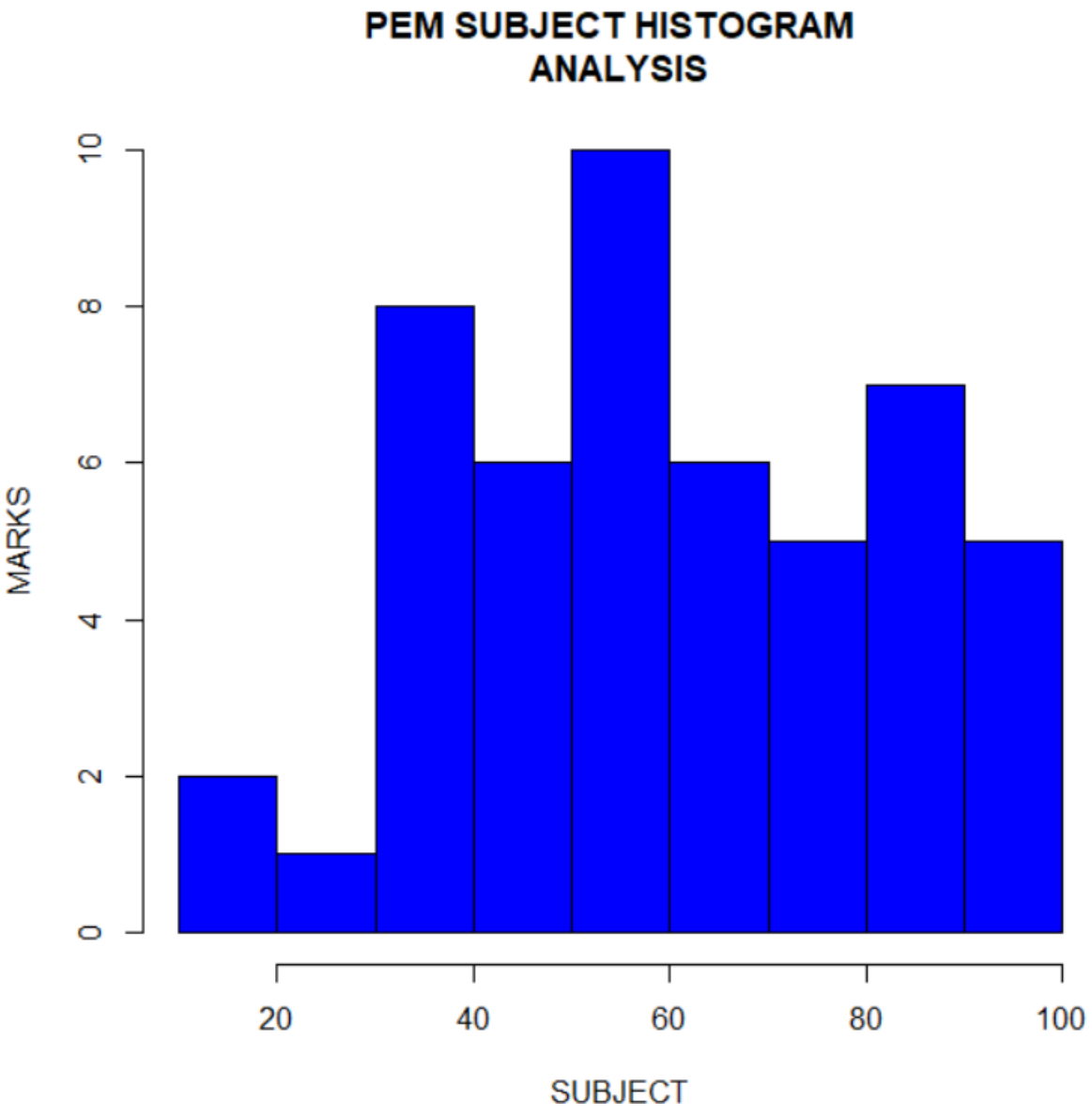
>hist(data$COA,breaks=10,main="COA SUBJECT HISTOGRAM ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="red",border="black")



>hist(data$DM,breaks=10,main="DM SUBJECT HISTOGRAM ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="yellow",border=”green”)

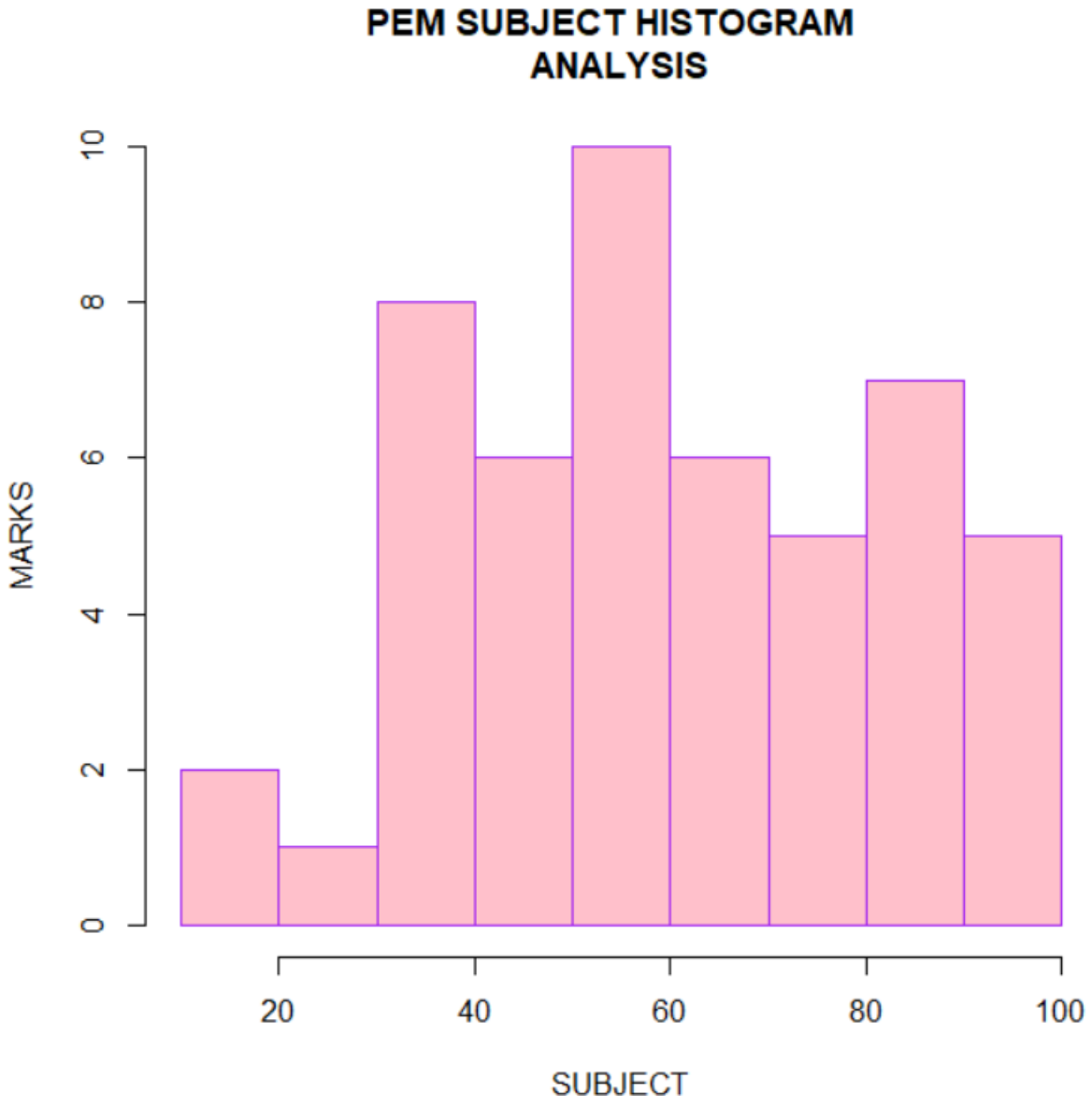



```
>hist(data$PEM,breaks=10,main="PEM SUBJECT HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="blue",border="black")
```

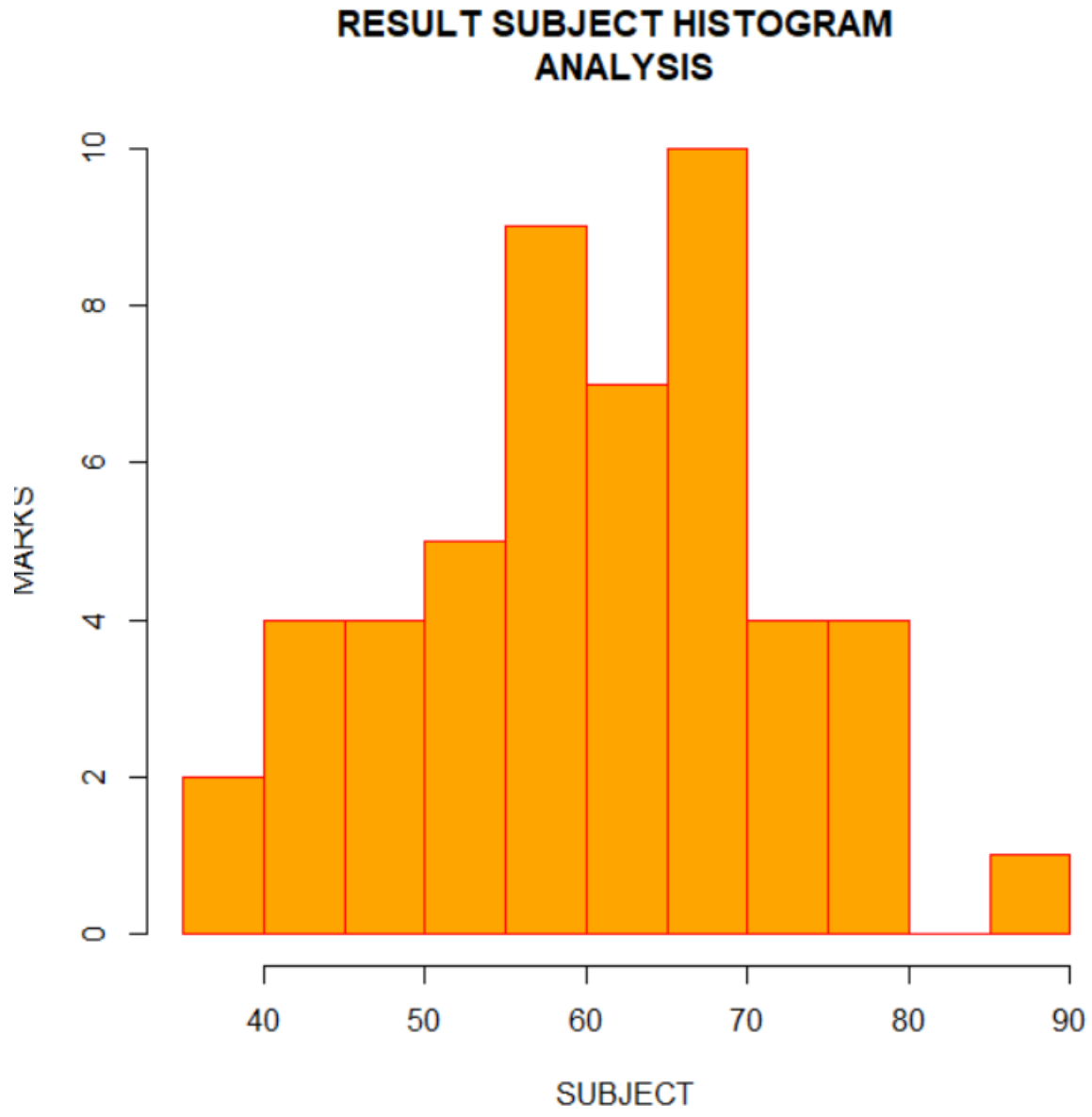


>hist(data$PEM,breaks=10,main="PEM SUBJECT HISTOGRAM

ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="pink",border="purple")

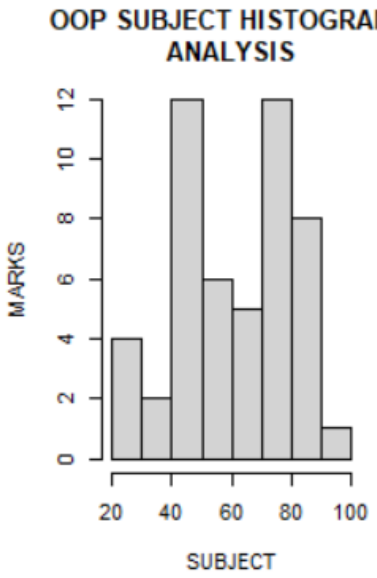


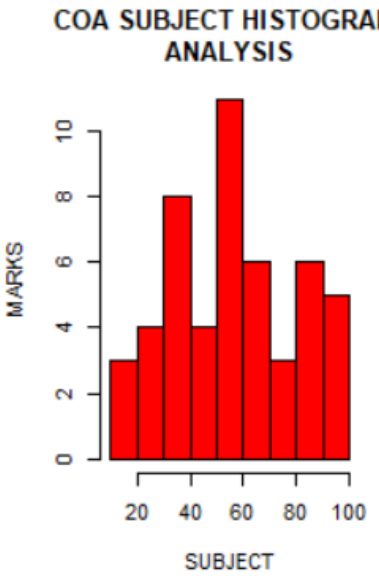
hist(data$RESULT,breaks=10,main="RESULT SUBJECT HISTOGRAM ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="orange",border="red")

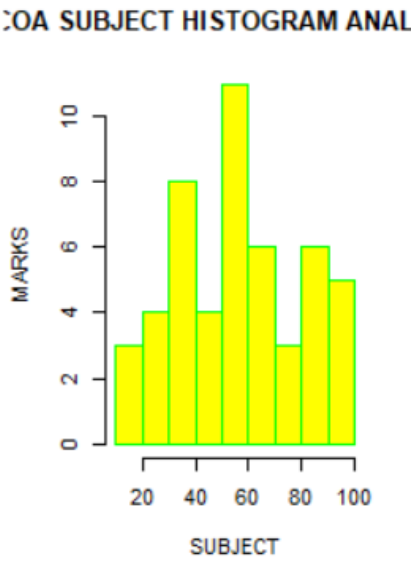


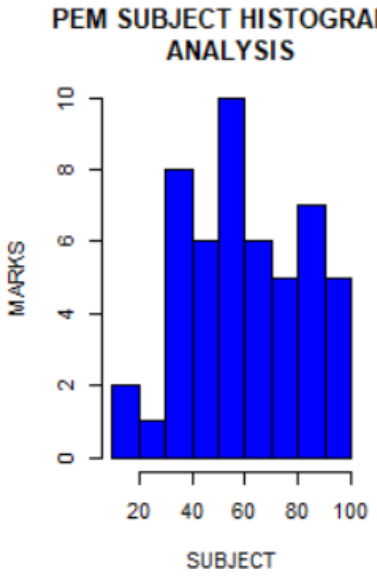
>par(mfrow=c(2,3))

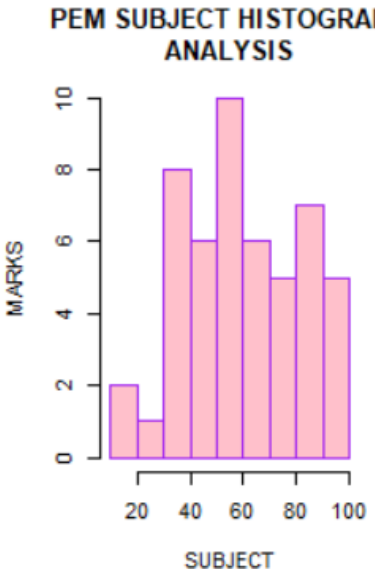
```
> hist(data$OOP,breaks=10,main="OOP SUBJECT HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS") >hist(data$COA,breaks=10,main="COA
      SUBJECT      HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="red",border="black")

hist(data$DM,breaks=10,main="DM SUBJECT HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="yellow",border=”green”)

>hist(data$PEM,breaks=10,main="PEM SUBJECT HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="blue",border="black")

>hist(data$PEM,breaks=10,main="PEM SUBJECT HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="pink",border="purple")

hist(data$RESULT,breaks=10,main="RESULT          SUBJECT          HISTOGRAM
ANALYSIS",xlab="SUBJECT",ylab="MARKS",col="orange",border="red")
```

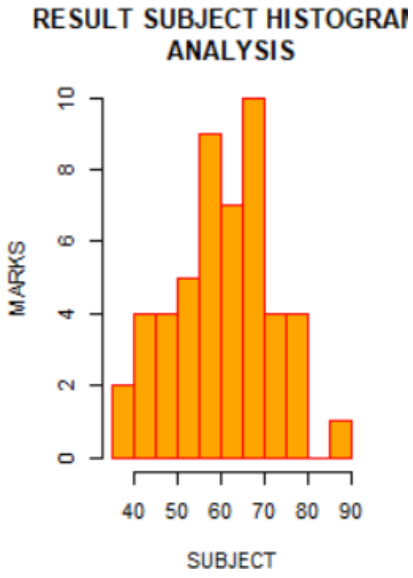

OOP SUBJECT HISTOGRAM ANALYSIS
MARKS
SUBJECT
0
2
4
6
8
10
12
20
40
60
80
100


COA SUBJECT HISTOGRAM ANALYSIS
MARKS
SUBJECT
0
2
4
6
8
10
20
40
60
80
100

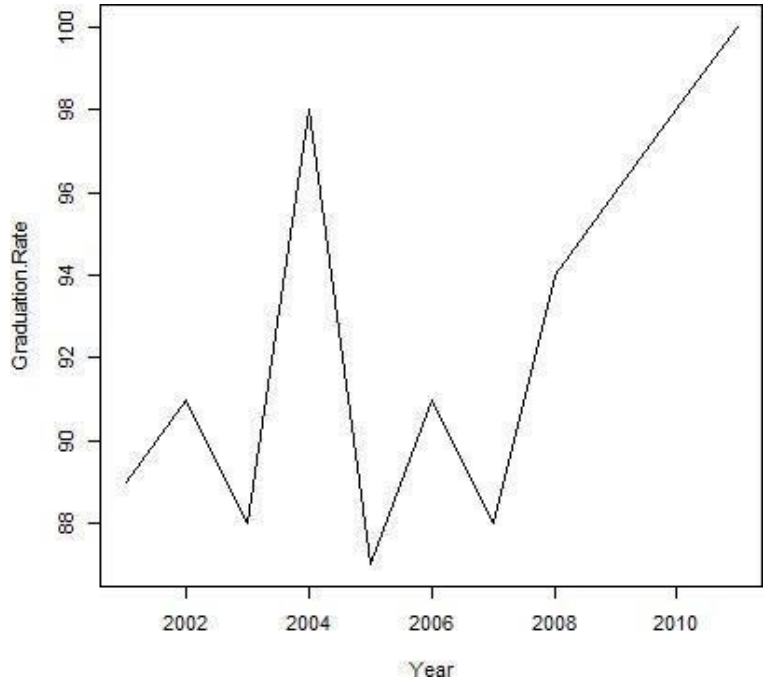

COA SUBJECT HISTOGRAM ANAL
MARKS
SUBJECT
0
2
4
6
8
10
20
40
60
80
100


PEM SUBJECT HISTOGRAM ANALYSIS
MARKS
SUBJECT
0
2
4
6
8
10
20
40
60
80
100

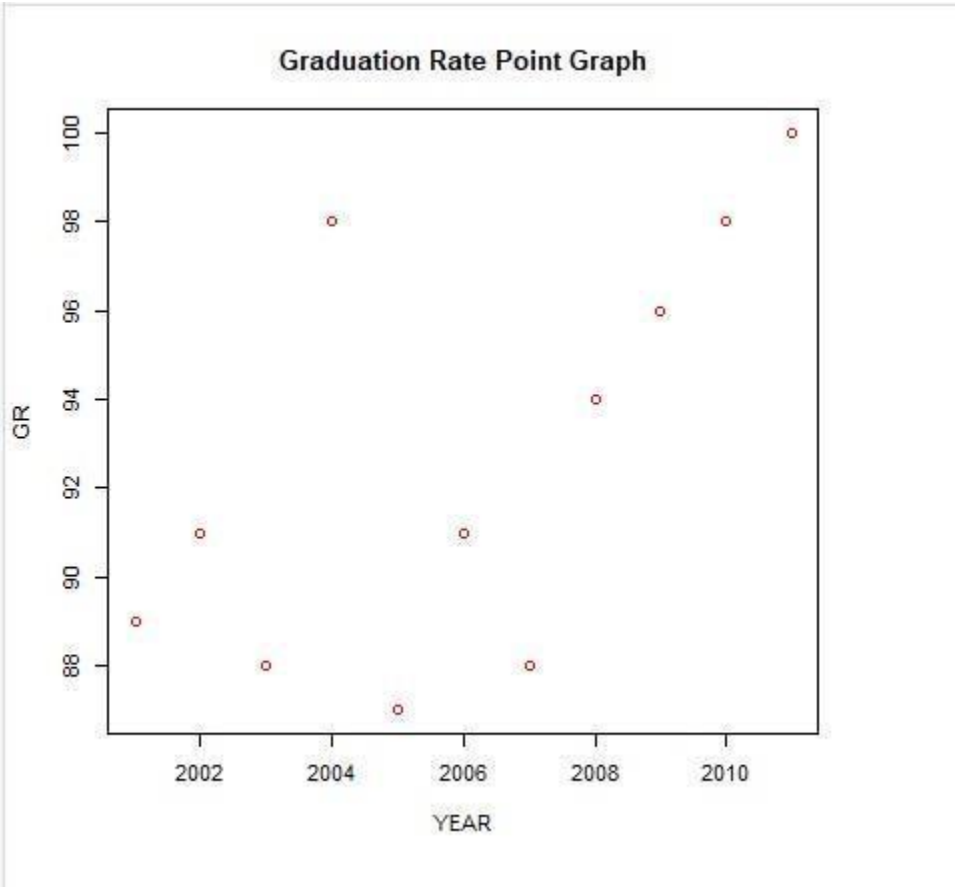

PEM SUBJECT HISTOGRAM ANALYSIS
MARKS
SUBJECT
0
2
4
6
8
10
20
40
60
80
100


RESULT SUBJECT HISTOGRAM ANALYSIS
MARKS
SUBJECT
0
2
4
6
8
10
40
50
60
70
80
90

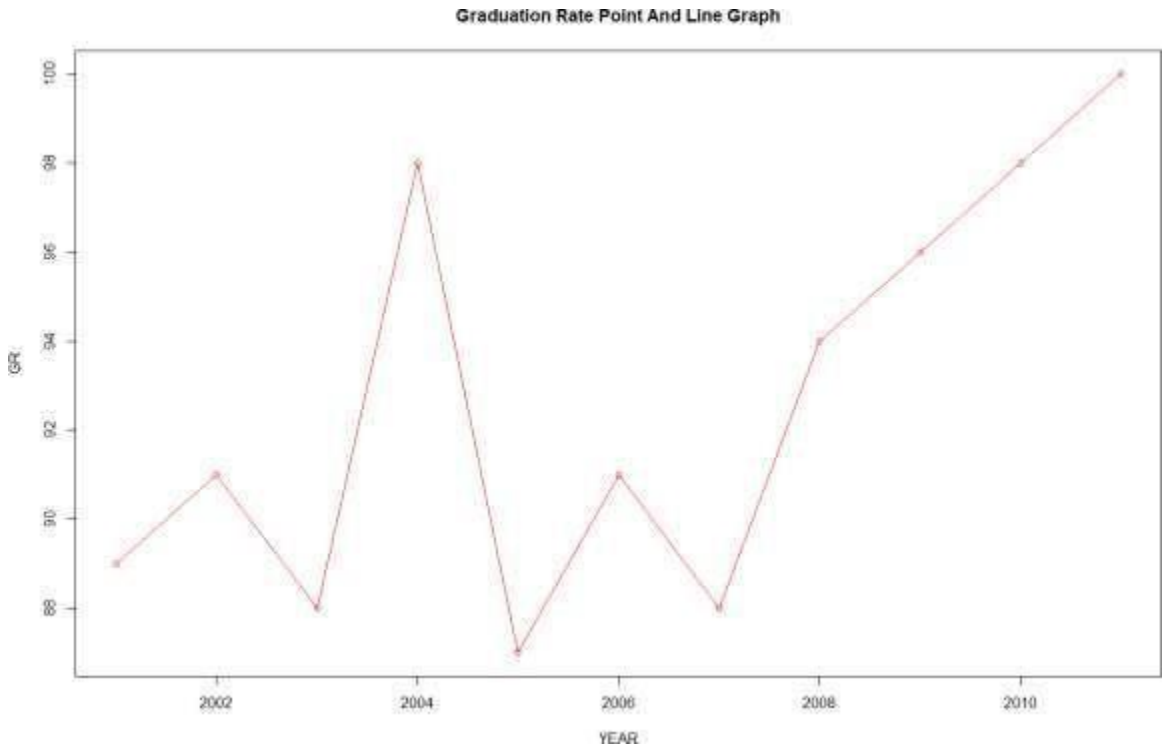
```
> data1<-read.csv("Grate.csv")
> print(data1)
   Year Graduation.Rate
7 2007              88
8 2008              94
9 2009              96
10 2010              98
11 2011             100
```



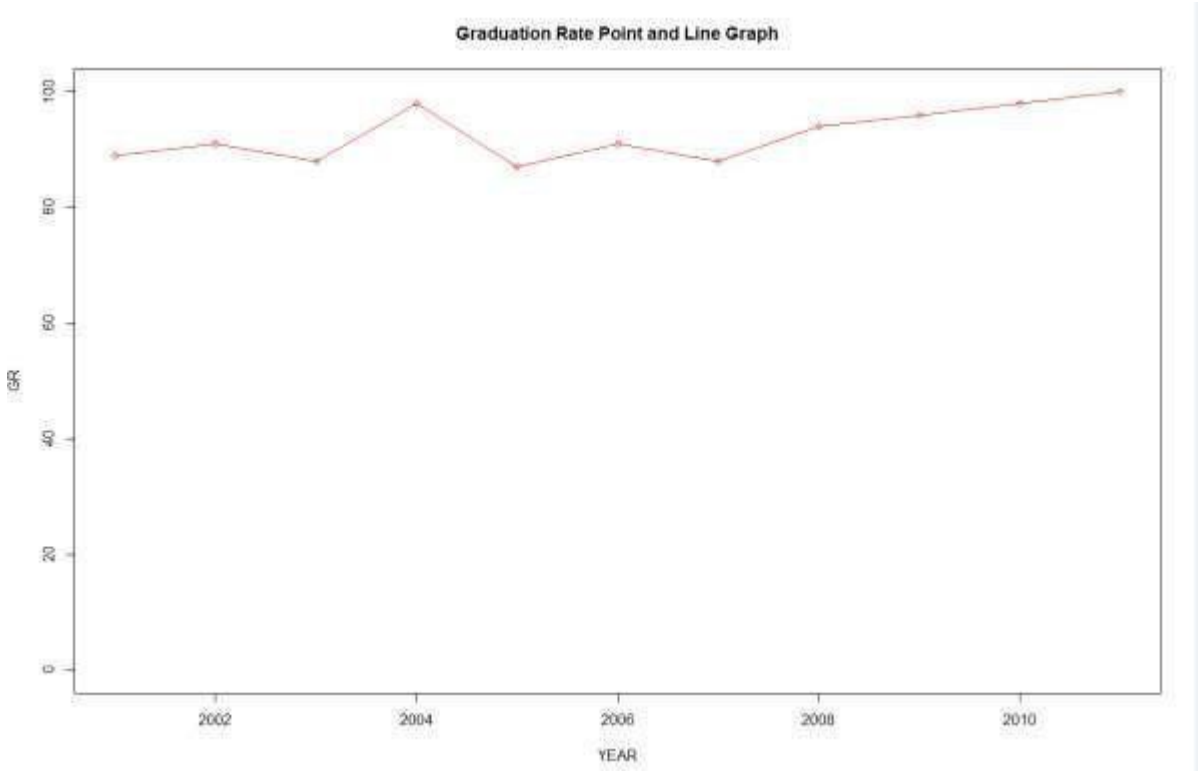

```
>plot(data1,type="p",main="Graduation Rate Point
  Graph",xlab="YEAR",ylab="GR",col="RED")
```



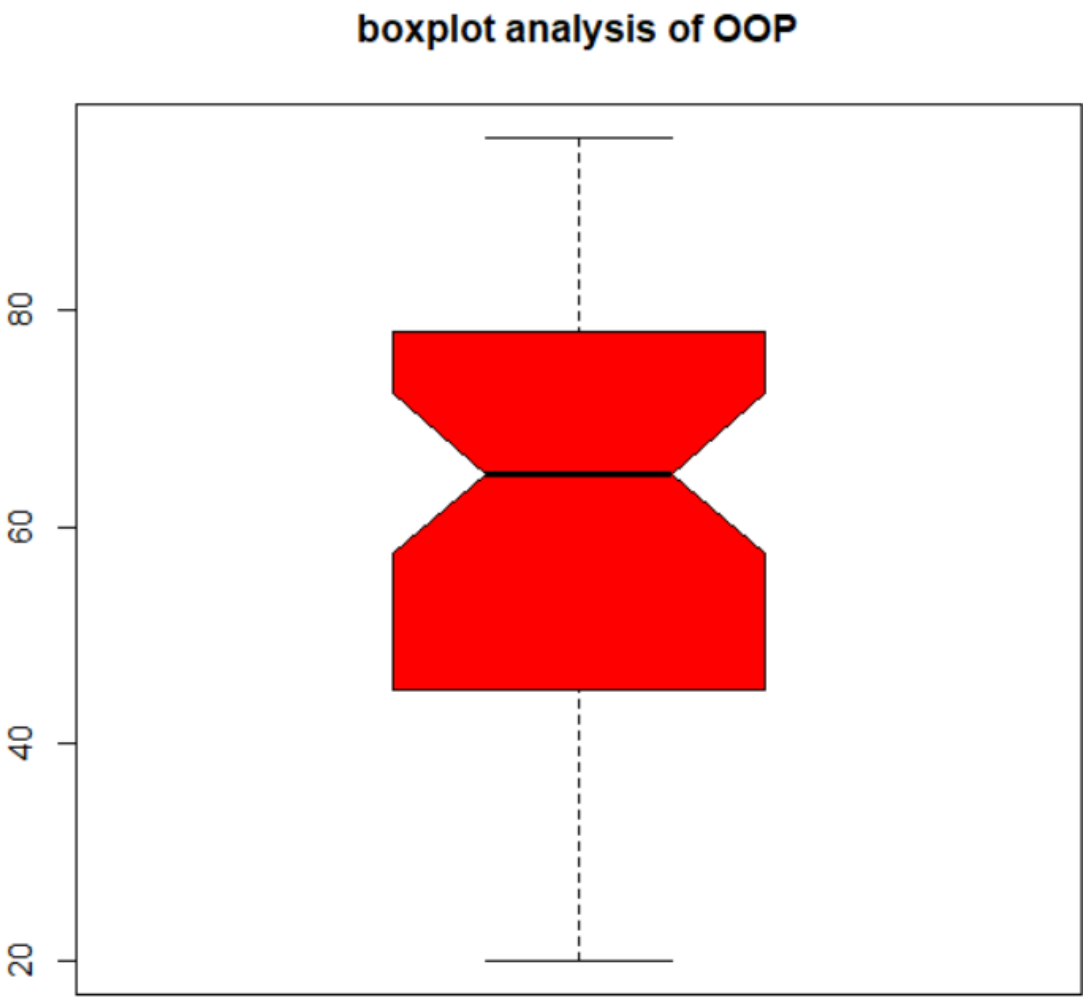

```
plot(data1,type="o",main="Graduation Rate Point And Line
  Graph",xlab="YEAR",ylab="GR",col="RED")
```



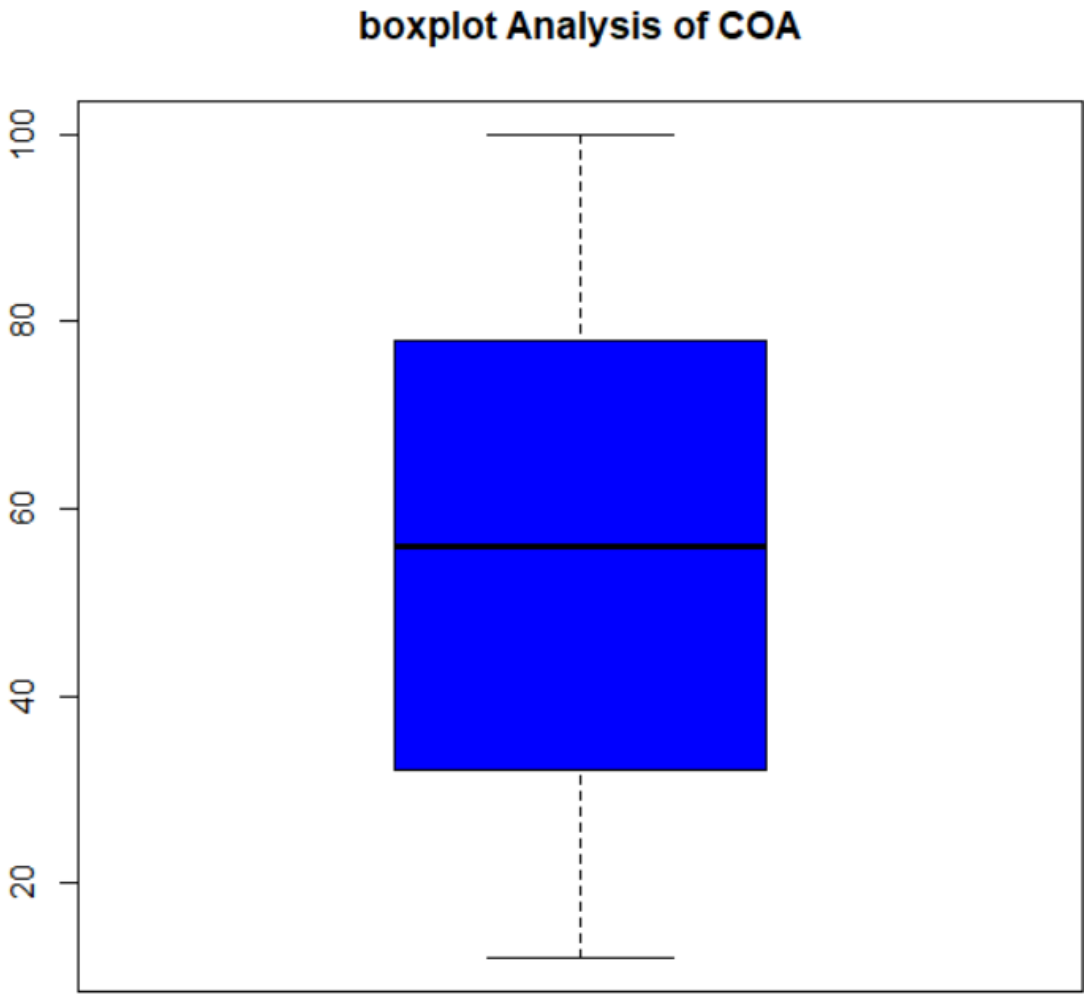

```
plot(data1,type="o",main="Graduation Rate Point and Line Graph",xlab="YEAR",ylab="GR",col="RED",xlim=c(2001,2011),ylim=c(0,100))
```



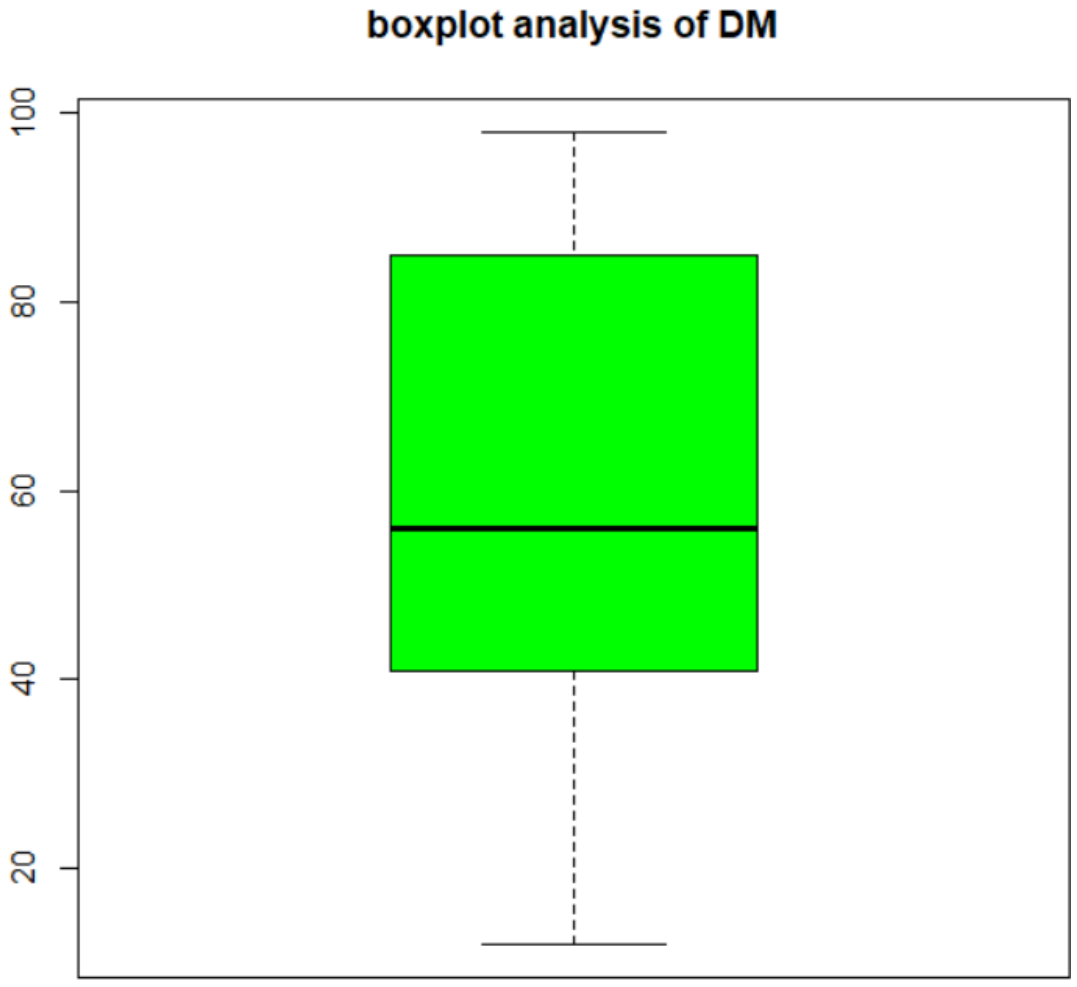

```
boxplot(data$OOP,notch=TRUE,col="red",main="boxplot analysis of OOP")
```


boxplot analysis of OOP
80
60
40
20

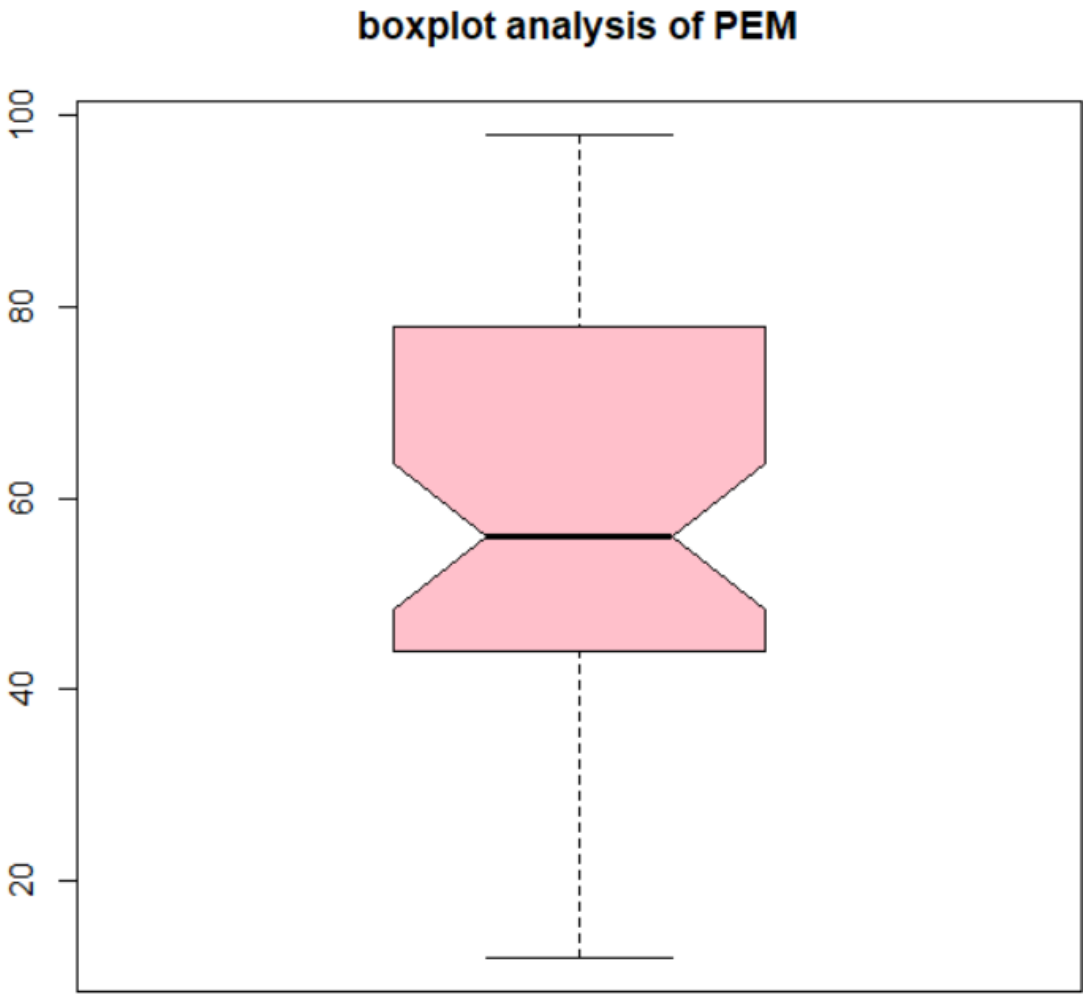
```
boxplot(data$COA,notch=FALSE,col="blue",main="boxplot Analysis of COA")
```


boxplot Analysis of COA
100
80
60
40
20

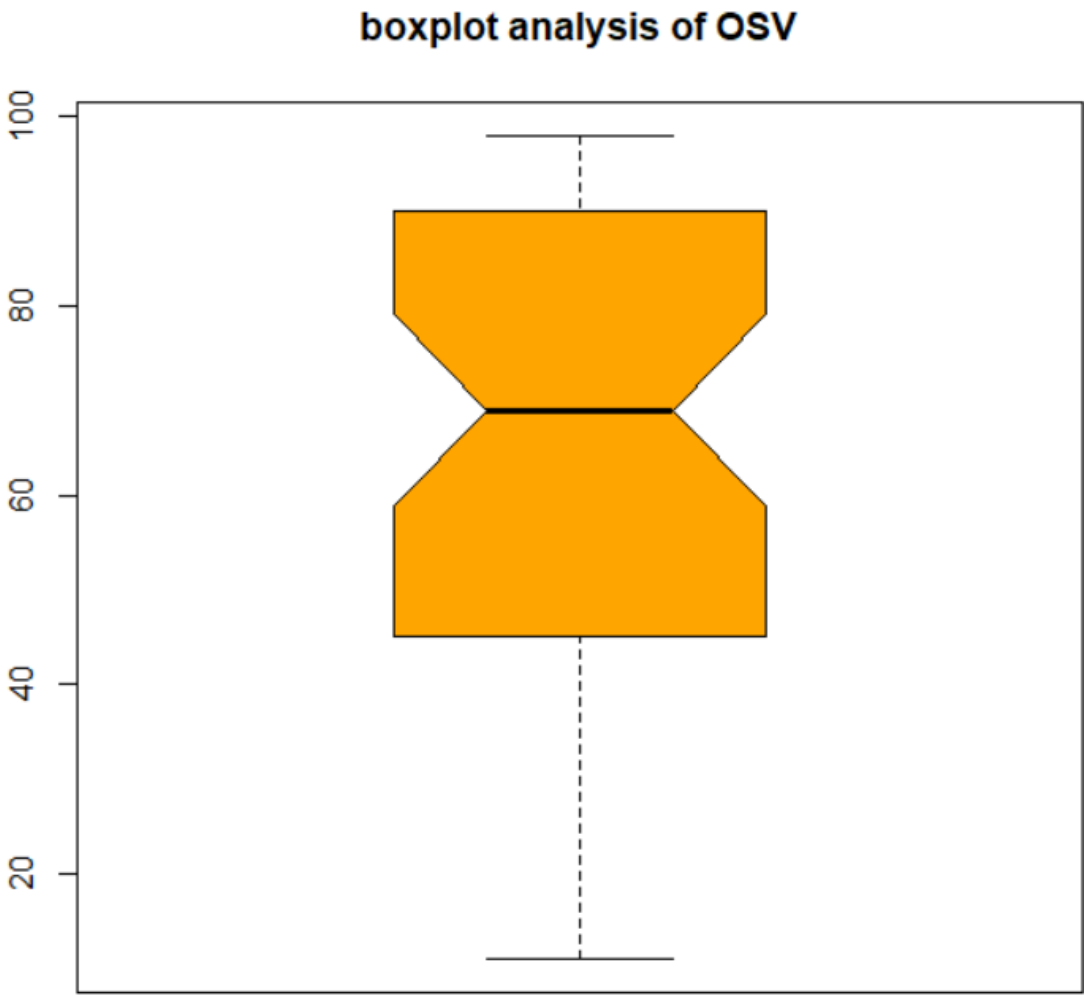
```
boxplot(data$DM,notch=FALSE,col="green",main="boxplot analysis of DM")
```


boxplot analysis of DM
100
80
60
40
20

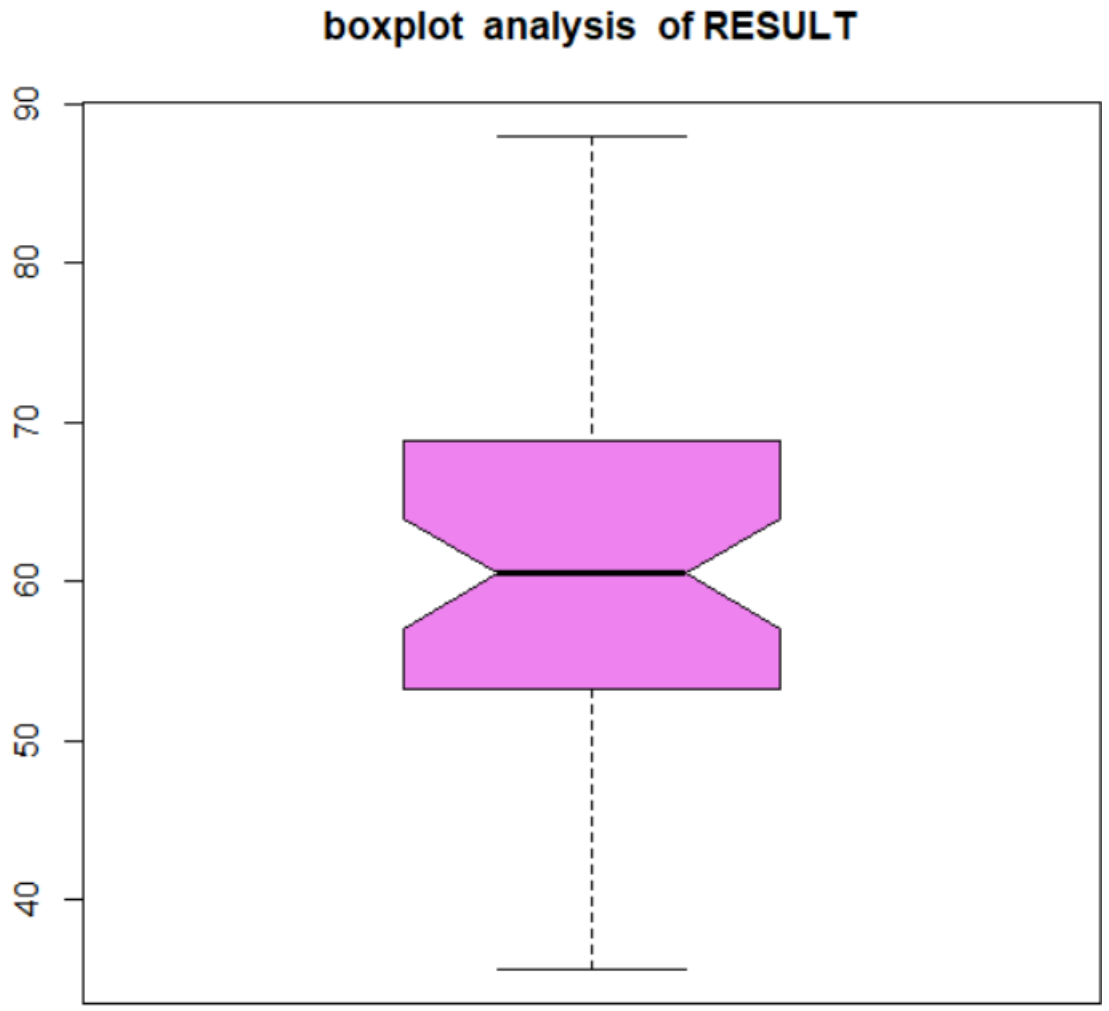
```
boxplot(data$PEM,notch=TRUE,col="pink",main="boxplot analysis of
PEM")
```


boxplot analysis of PEM
100
80
60
40
20

```
boxplot(data$OSV,notch=TRUE,col="orange",main="boxplot analysis of OSV")
```

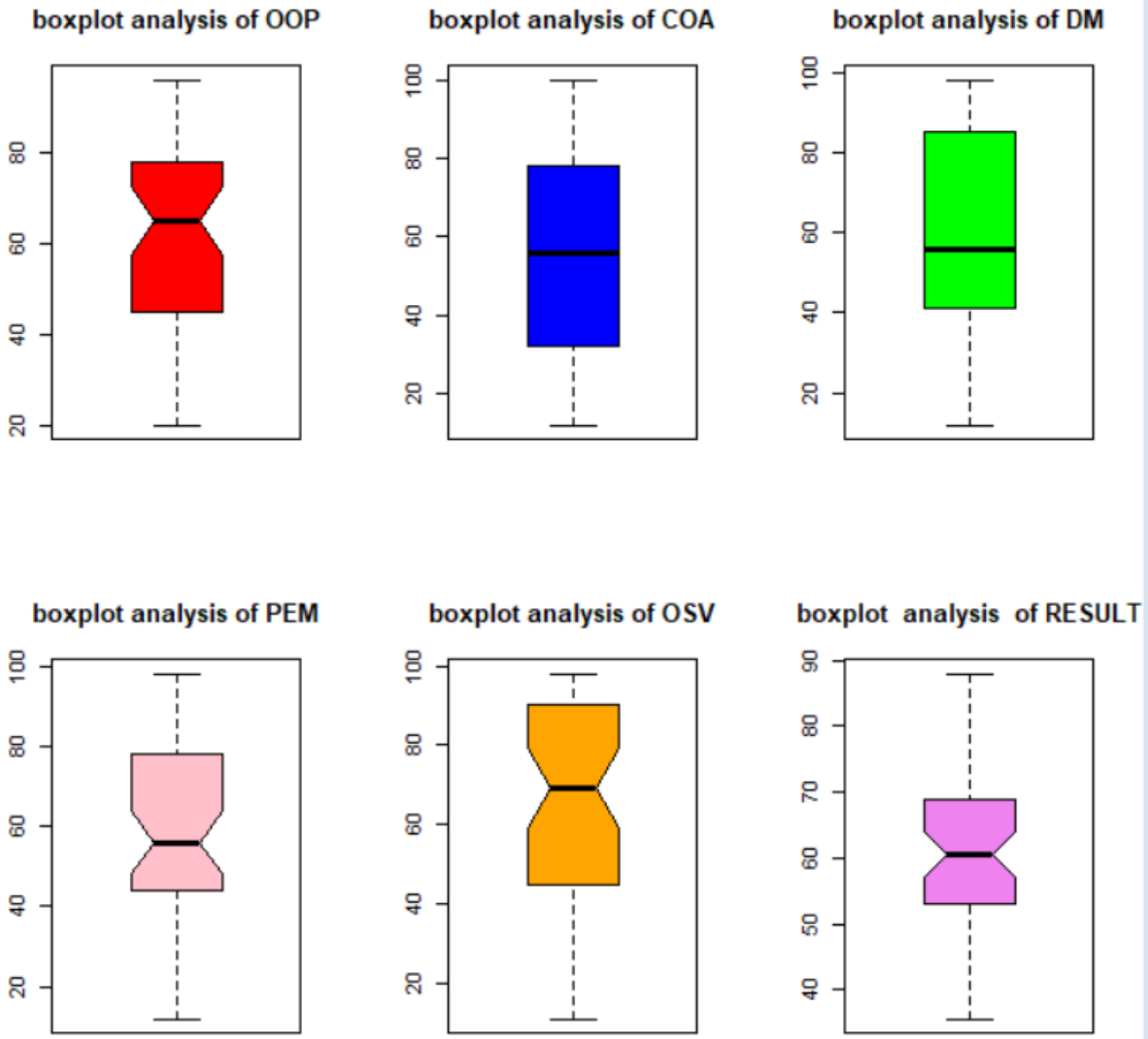


```
boxplot(data$RESULT,notch=TRUE,col="violet",main="boxplot    analysis
 of RESULT")
```

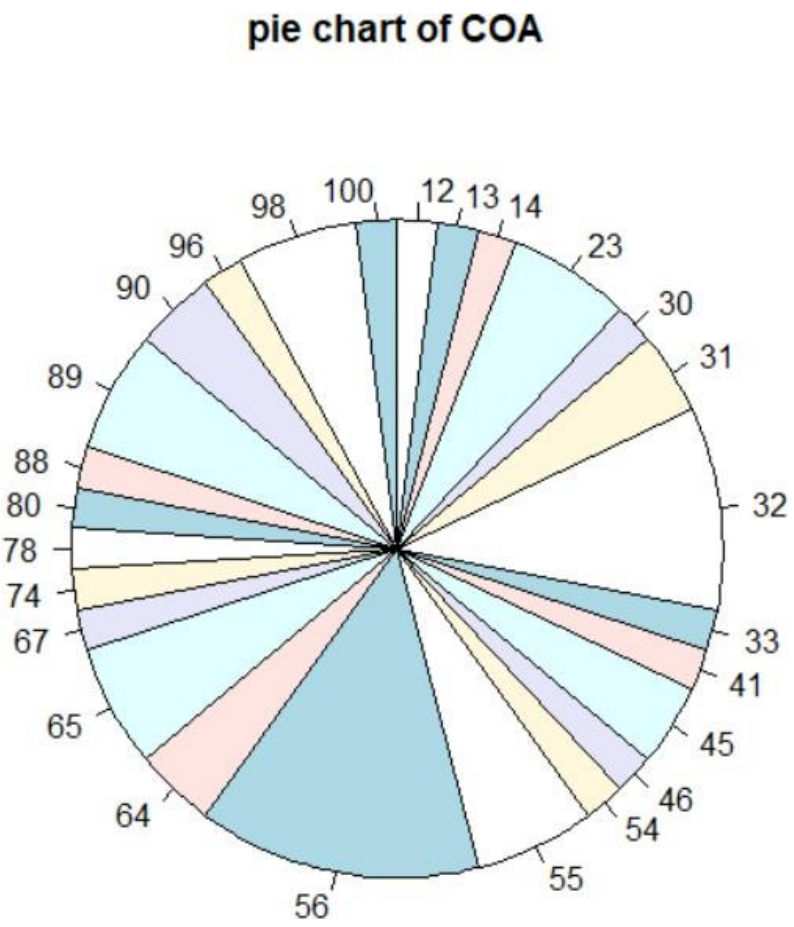



```
>par(mfrow=c(2,3))

>boxplot(data$OOP,notch=TRUE,col="red",main="boxplot analysis of OOP")
```

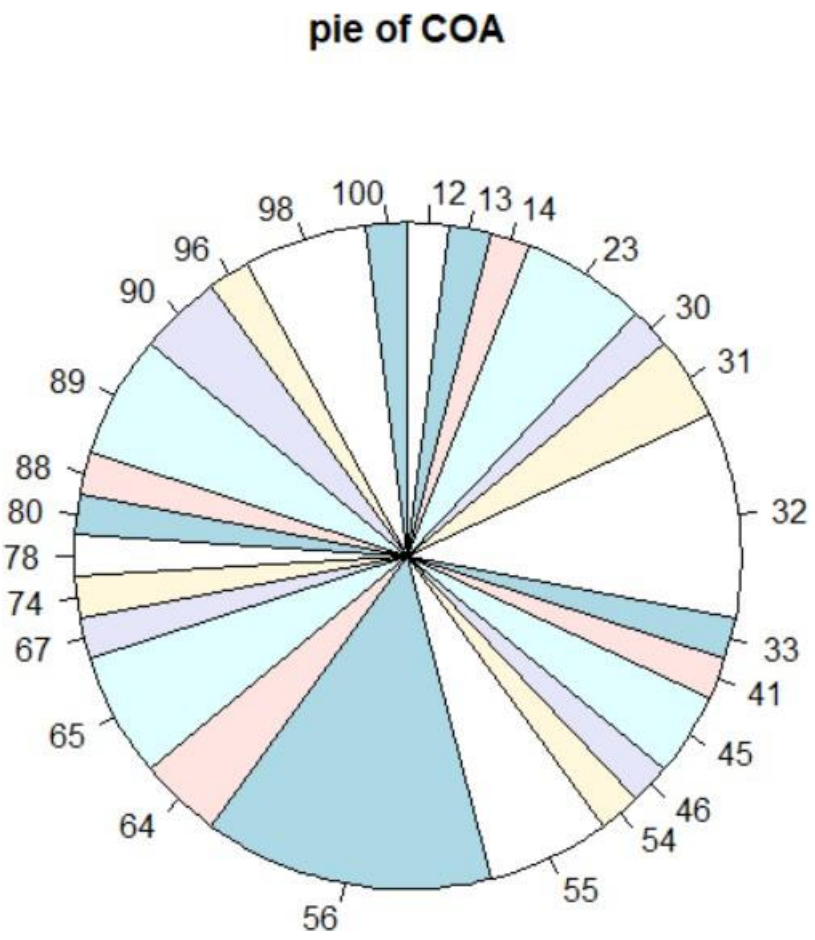
```
>boxplot(data$COA,notch=FALSE,col="blue",main="boxplot analysis of COA")

>boxplot(data$DM,notch=FALSE,col="green",main="boxplot analysis of DM")

>boxplot(data$PEM,notch=TRUE,col="pink",main="boxplot analysis of PEM")
>boxplot(data$OSV,notch=TRUE,col="orange",main="boxplot analysis of
OSV")

>boxplot(data$RESULT,notch=TRUE,col="violet",main="boxplot analysis of
RESULT")
```




```
pie(table(data$COA),main="pie chart of COA",clockwise=TRUE)
```

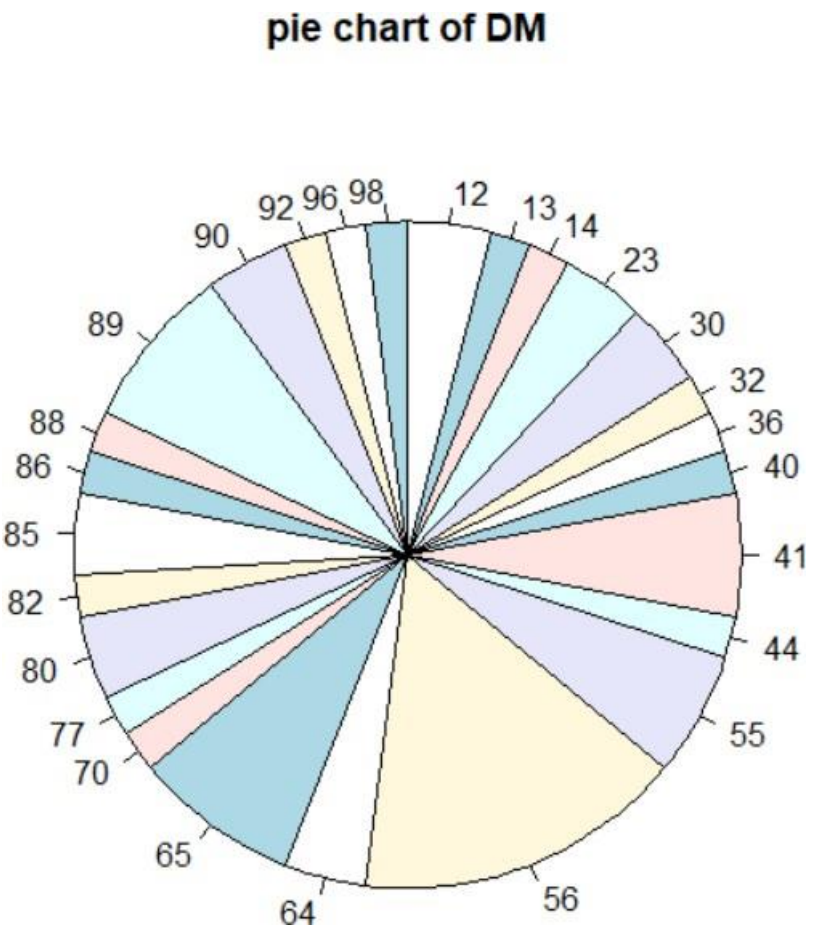

pie chart of COA
100
12
13
14
23
30
31
32
33
41
45
46
54
55
56
64
65
67
74
78
80
88
89
90
96
98

```
pie(table(data$COA),main="pie          of COA",clockwise=TRUE)
```

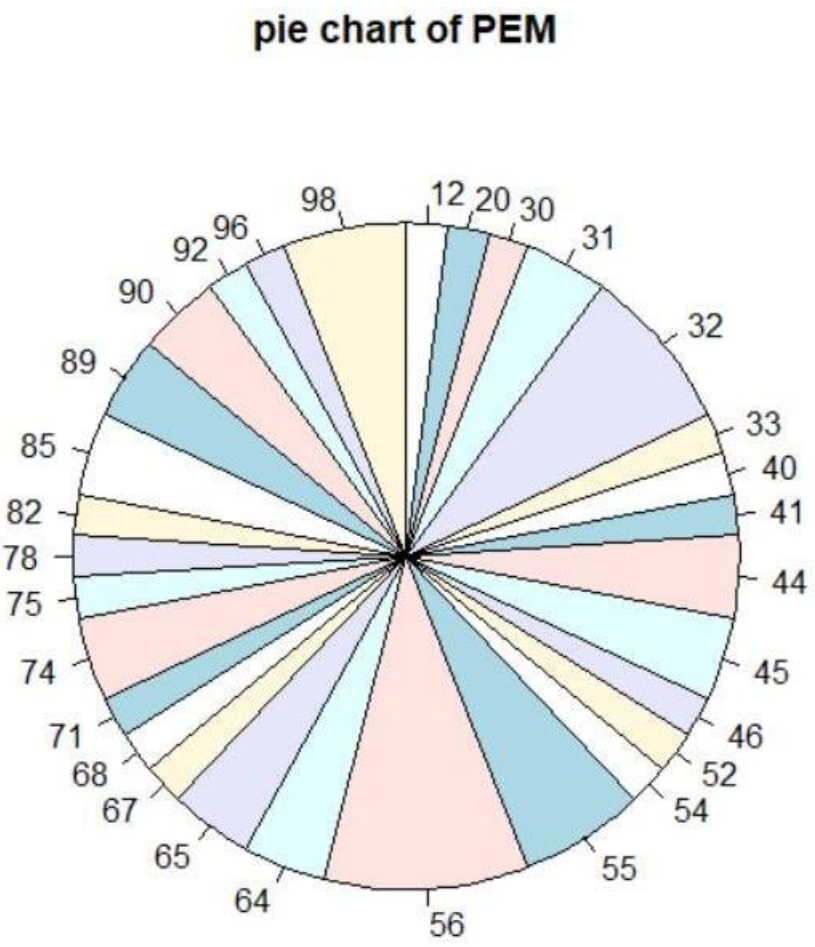



```
pie(table(data$DM),main="pie chart of DM",clockwise=TRUE)
```

TYIT-2-C **210410116108** chart

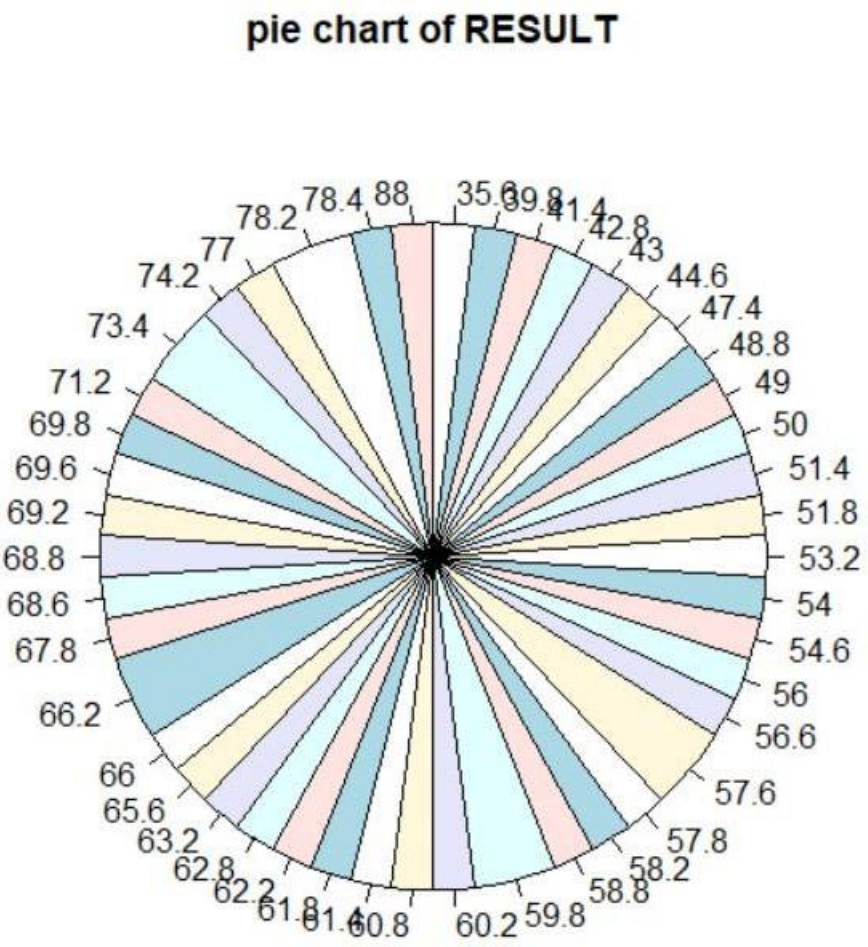


pie(table(data$PEM),main="pie chart of PEM",clockwise=TRUE)

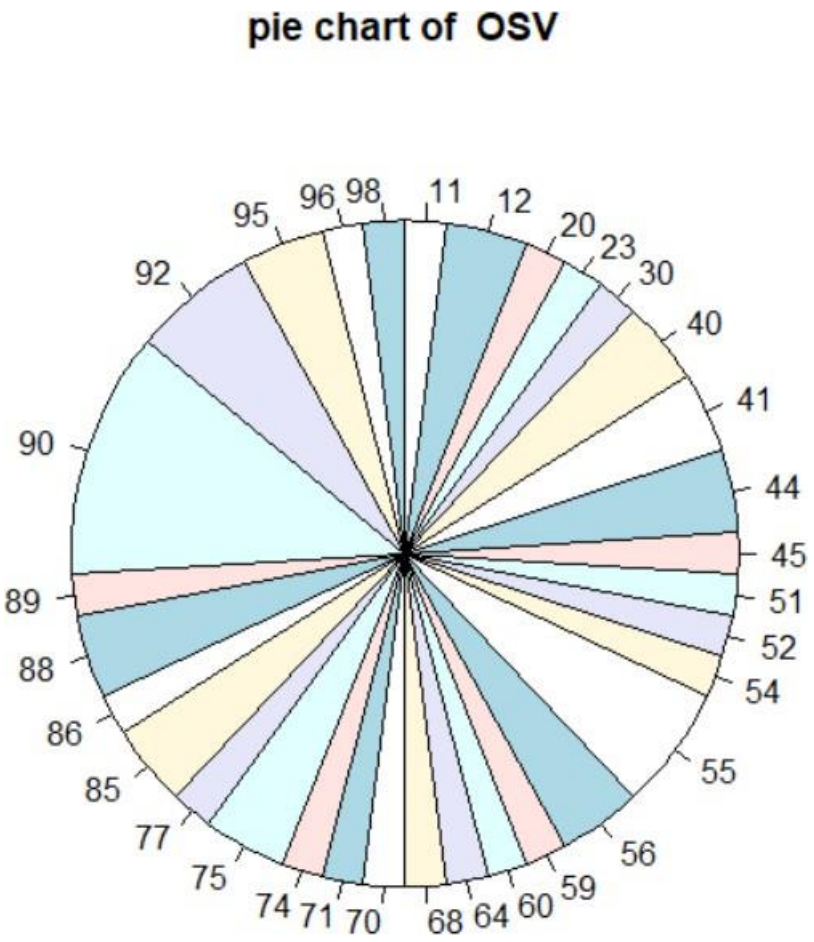



```
pie(table(data$RESULT),main="pie chart of RESULT",clockwise=TRUE)
```

TYIT-2-C **210410116108** chart



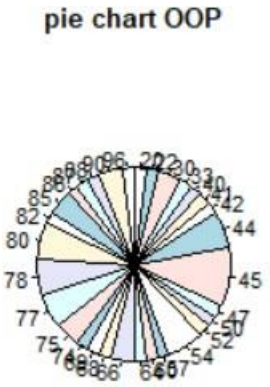
pie(table(data$OSV),main="pie chart of OSV",clockwise = TRUE)

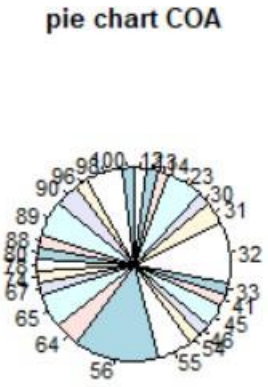


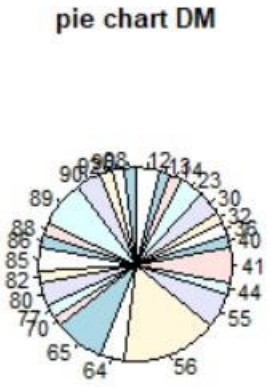

```
par(mfrow = c(2,3))


pie(table(data$OOP),main="pie chart OOP",clockwise = TRUE)

pie(table(data$COA),main="pie chart COA",clockwise = TRUE)

pie(table(data$DM),main="pie chart DM",clockwise = TRUE)

pie(table(data$PEM),main="pie chart PEM",clockwise = TRUE)
```

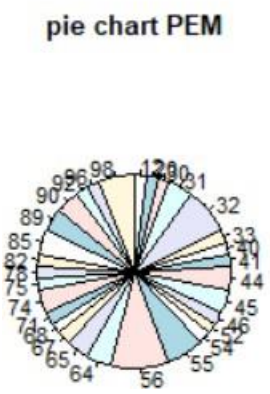
TYIT-2-C **210410116108** chart

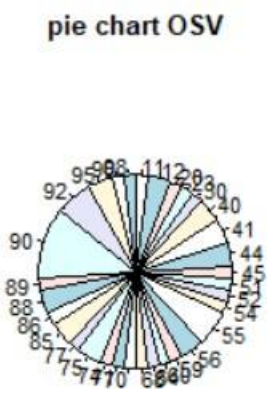
```
pie(table(data$RESULT),main="pie chart RESULT",clockwise = TRUE)

pie(table(data$OSV),main="pie chart OSV",clockwise = TRUE)
```



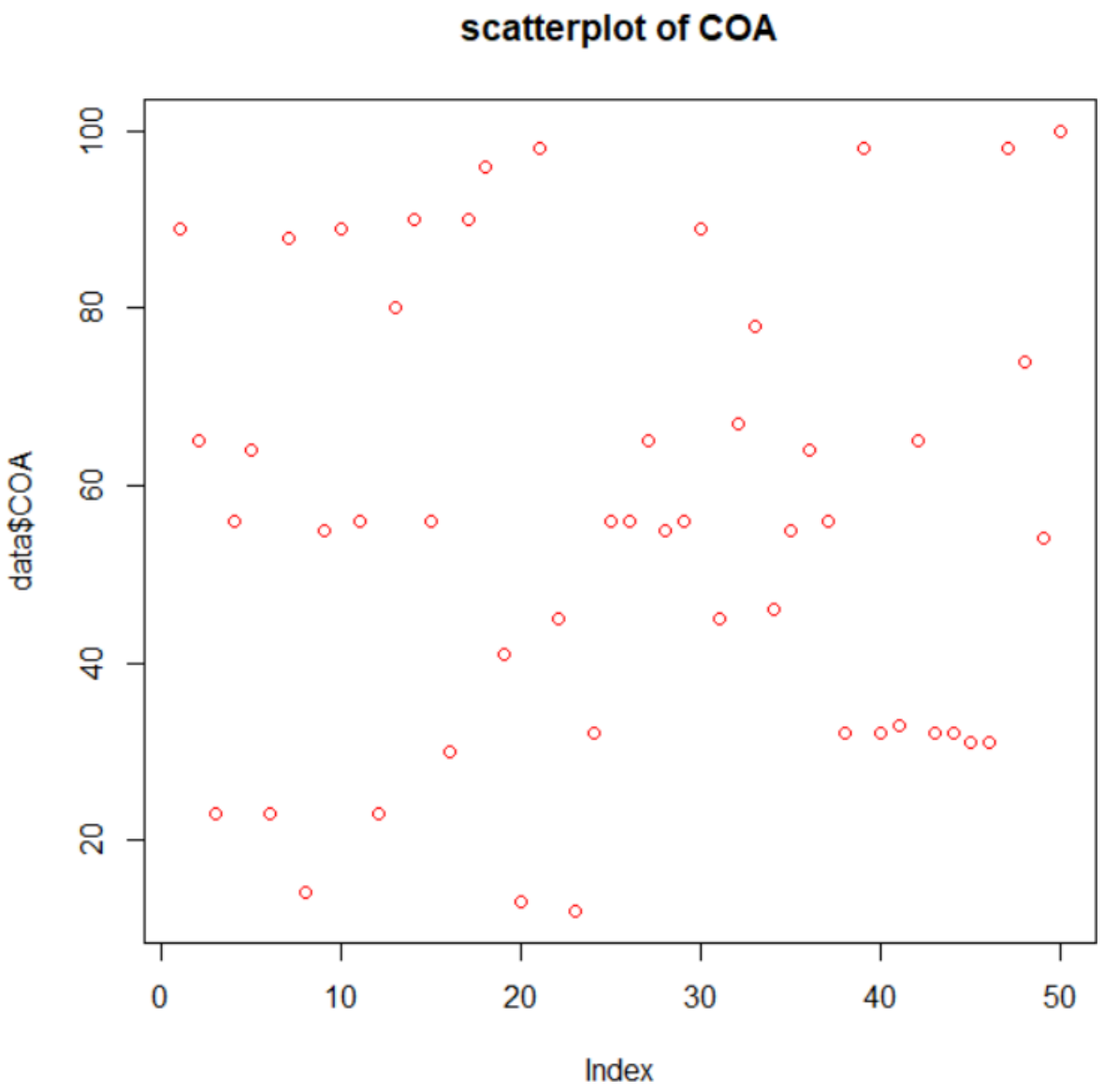




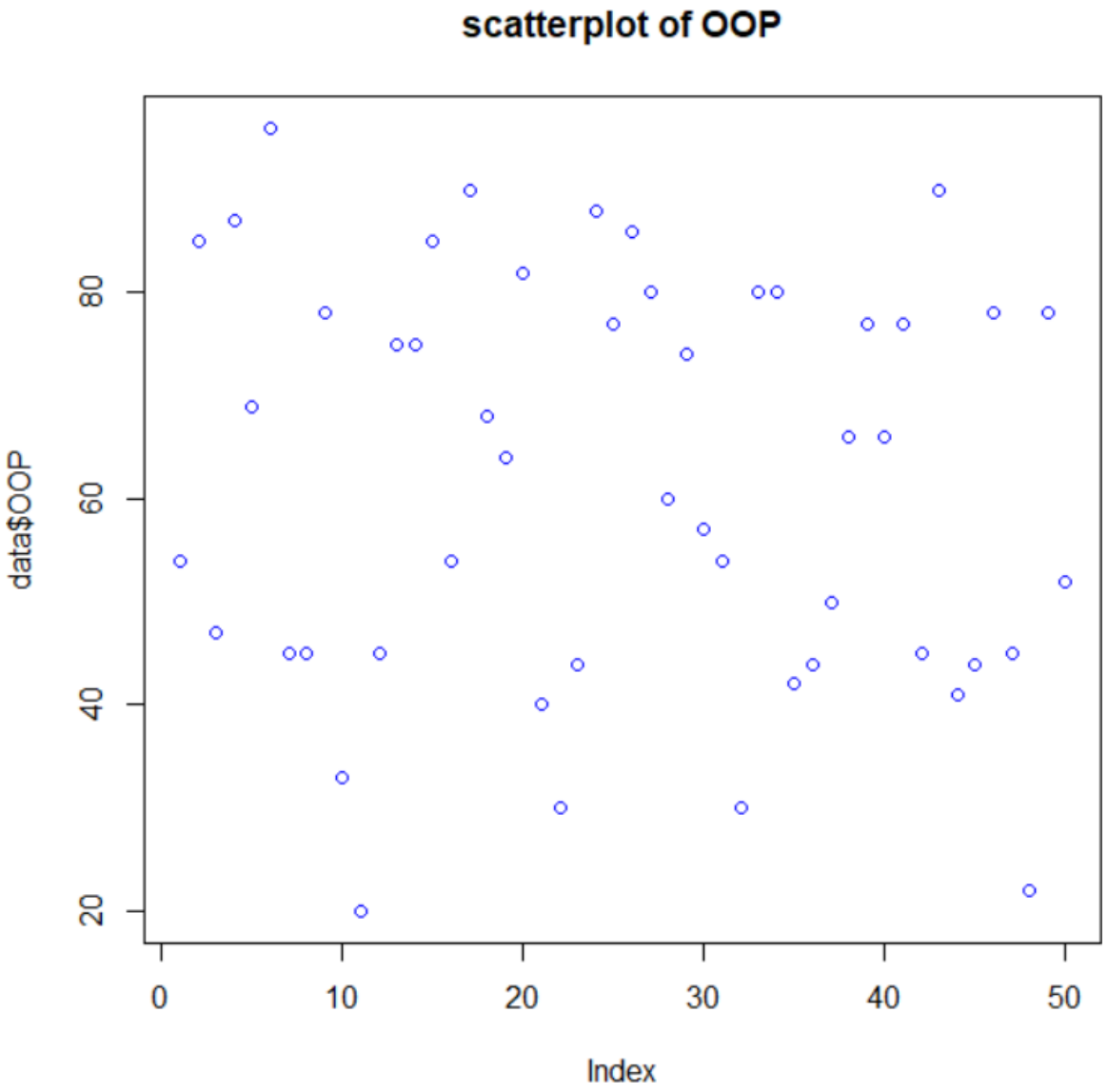




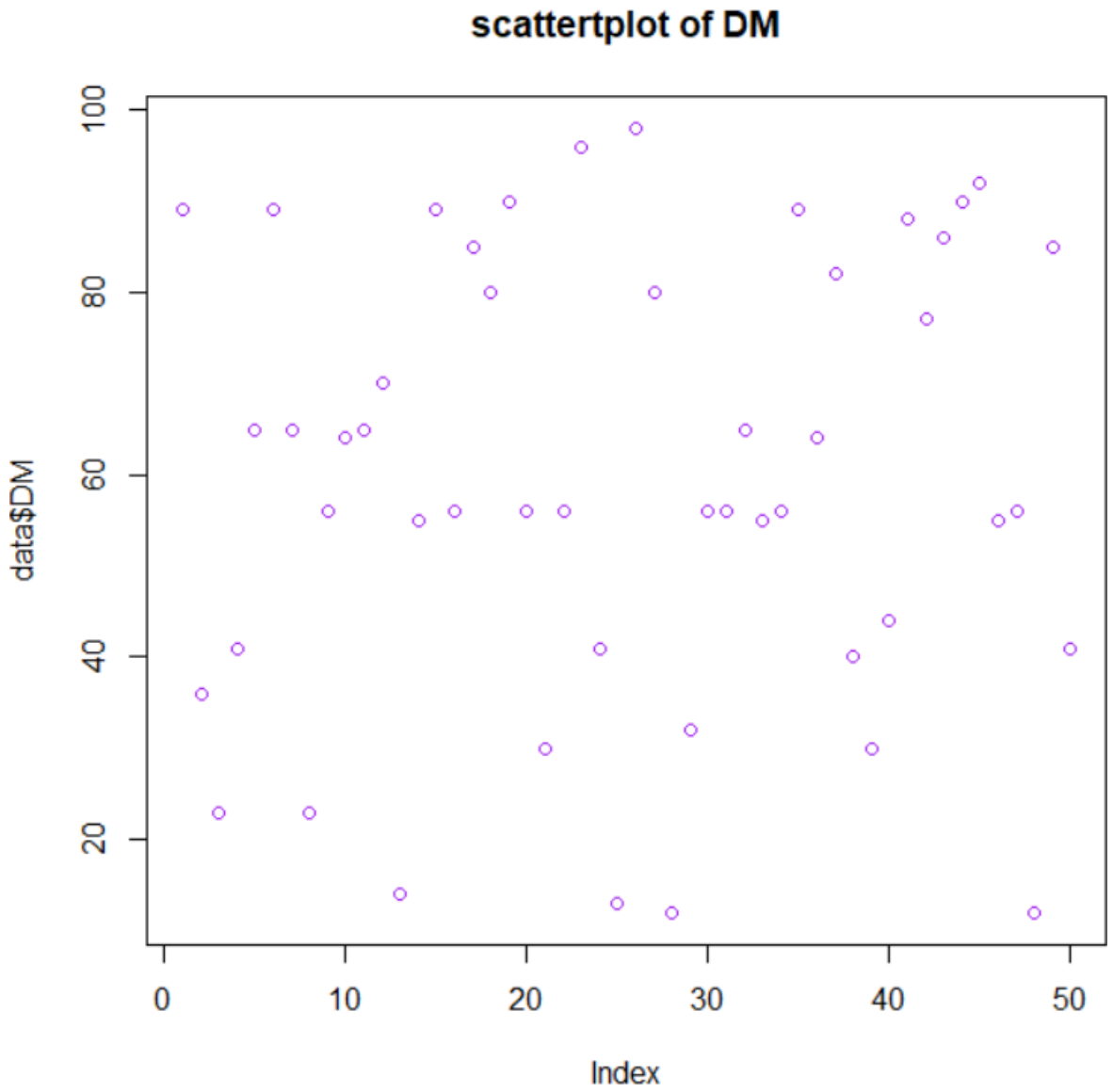
```
plot(data$COA , main="scatterplot of COA" ,col="red")
```



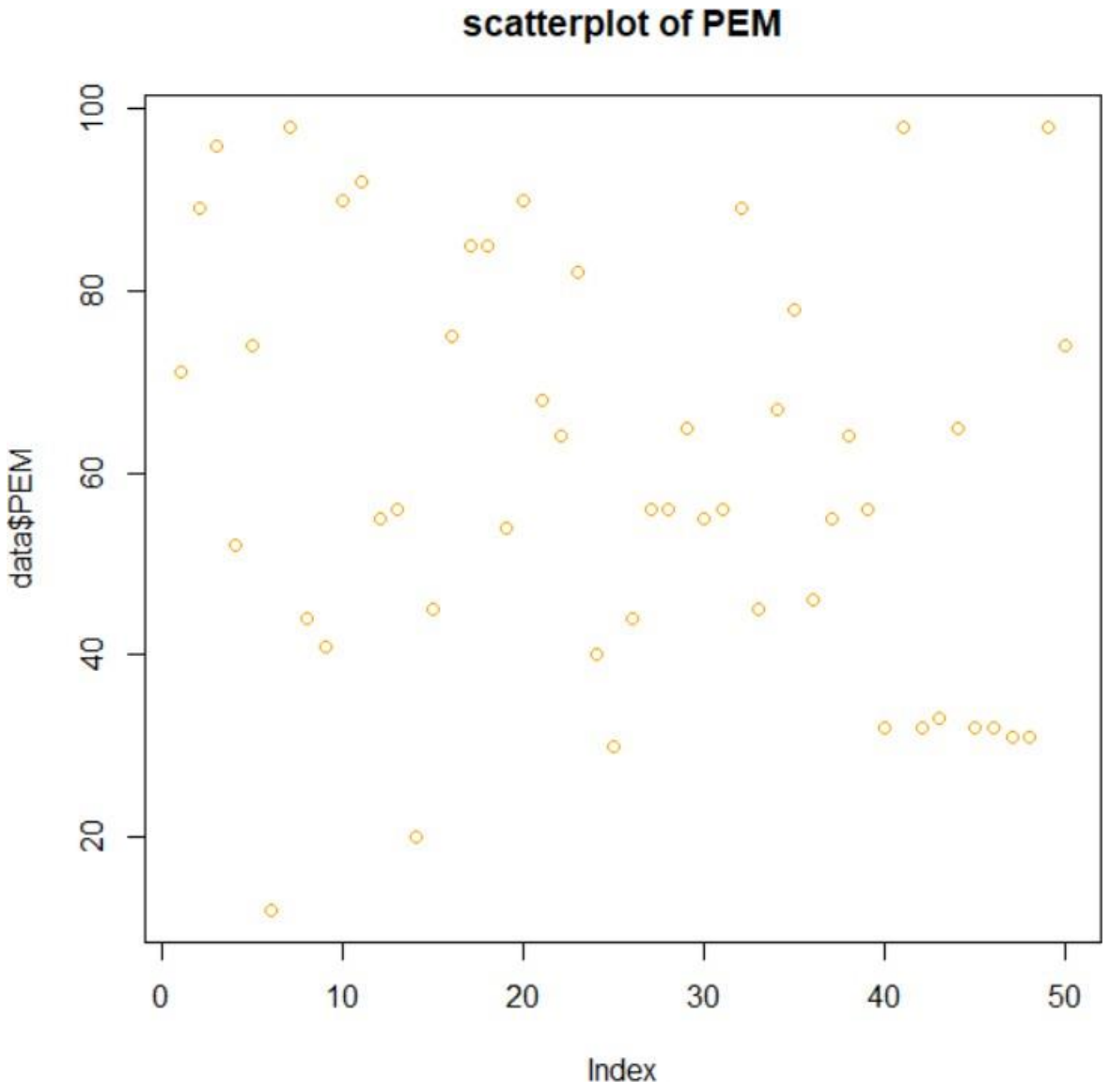

```
plot(data$OOP,main="scatterplot of OOP",col="blue")
```


scatterplot of OOP
data$OOP
Index
20
40
60
80
0
10
20
30
40
50

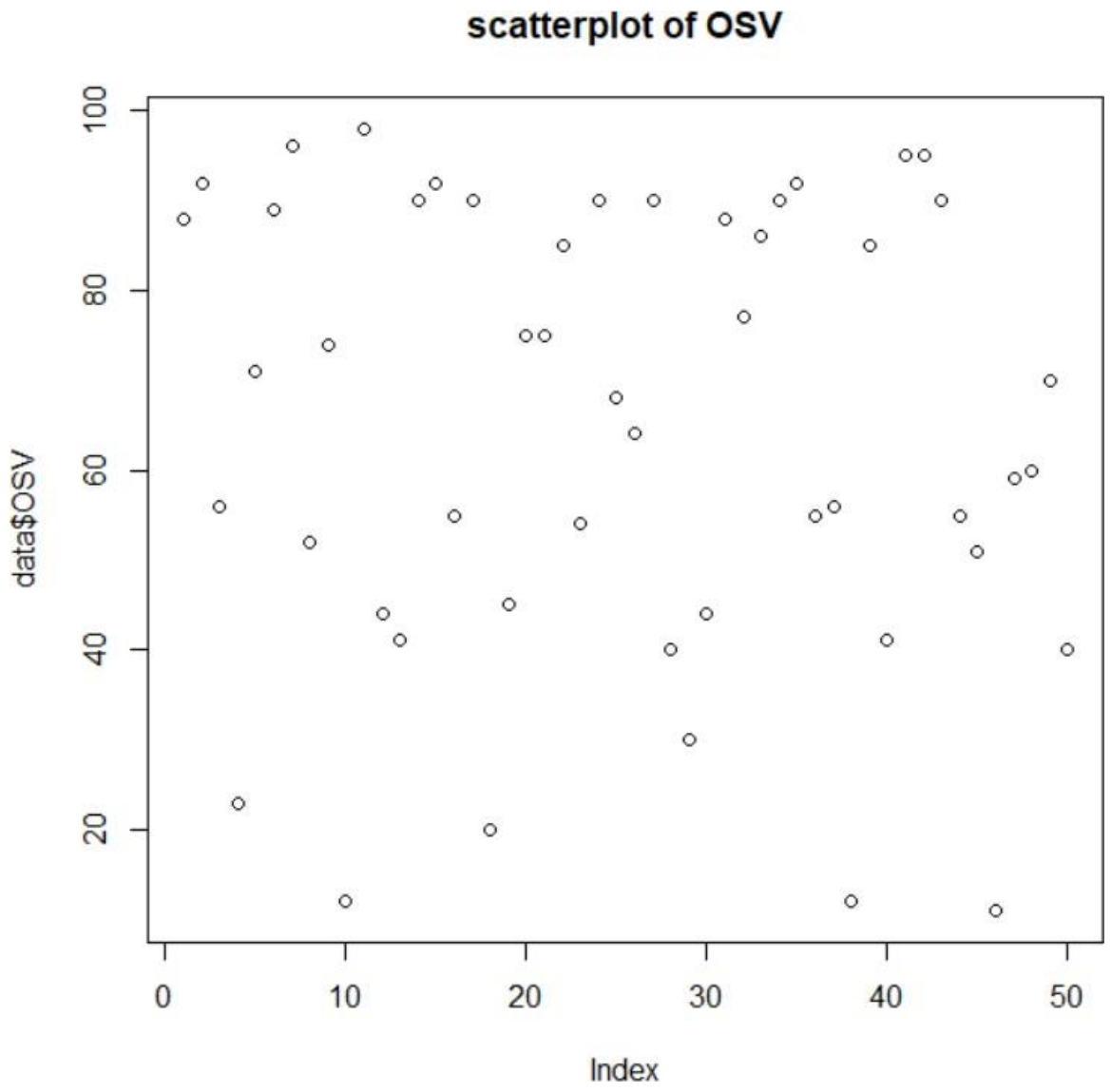
plot(data$DM,main="scatterplot of DM”,col="purple")



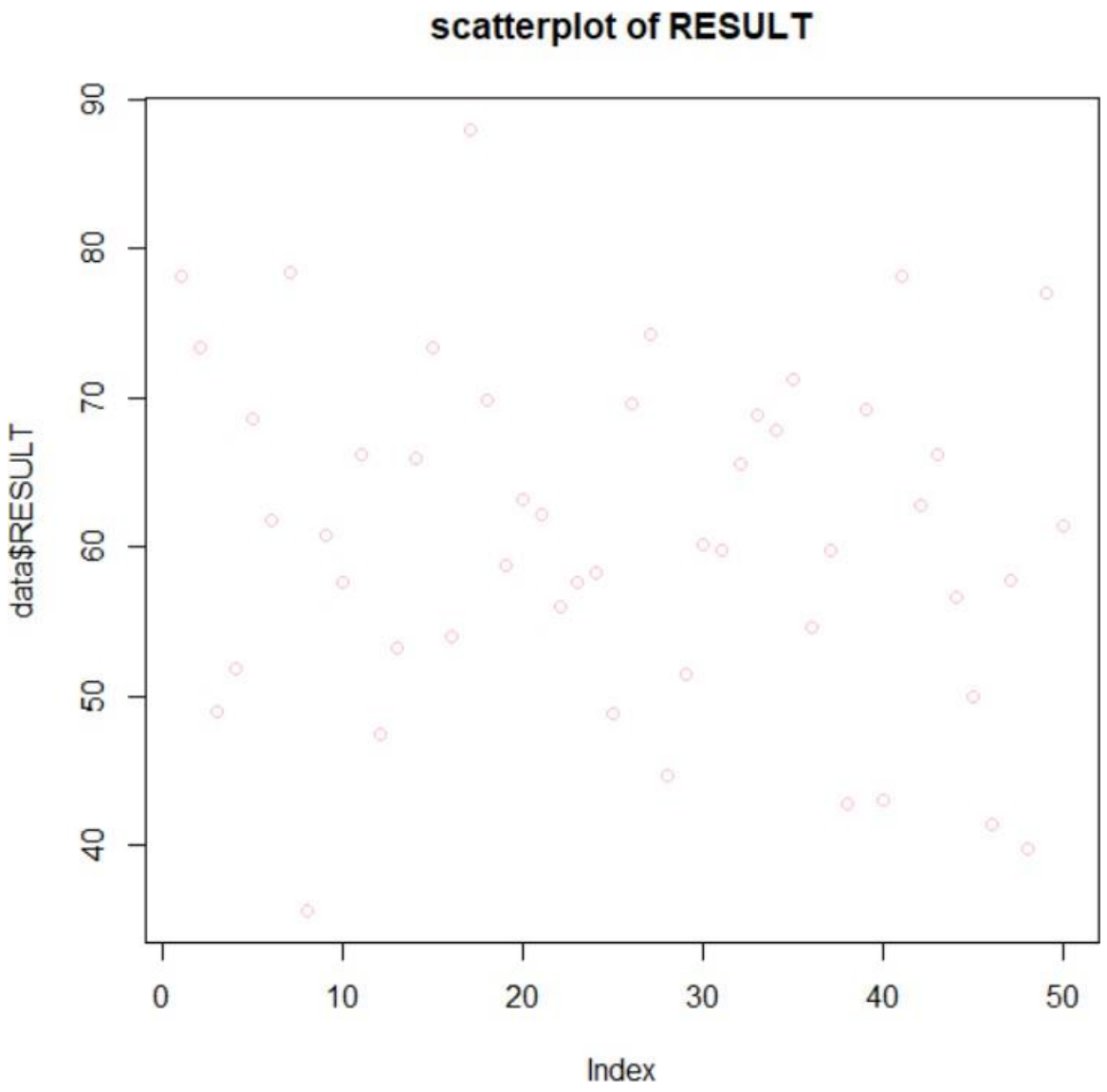
plot(data$PEM,main="scatterplot of PEM",col="orange")



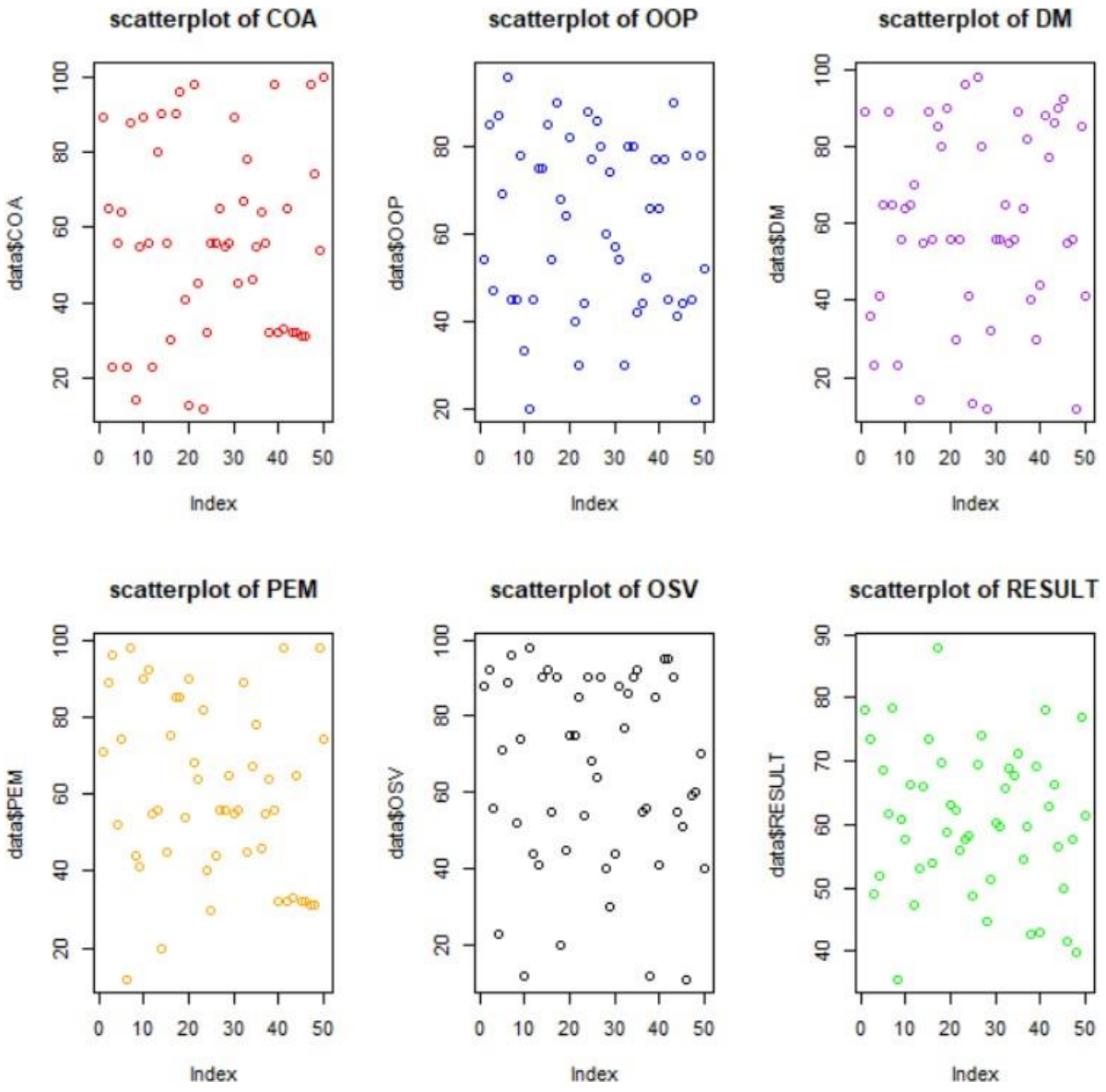
plot(data$OSV,main="scatterplot of OSV",col”black”)


scatterplot of OSV
data$OSV
Index
100
80
60
40
20
0
10
20
30
40
50

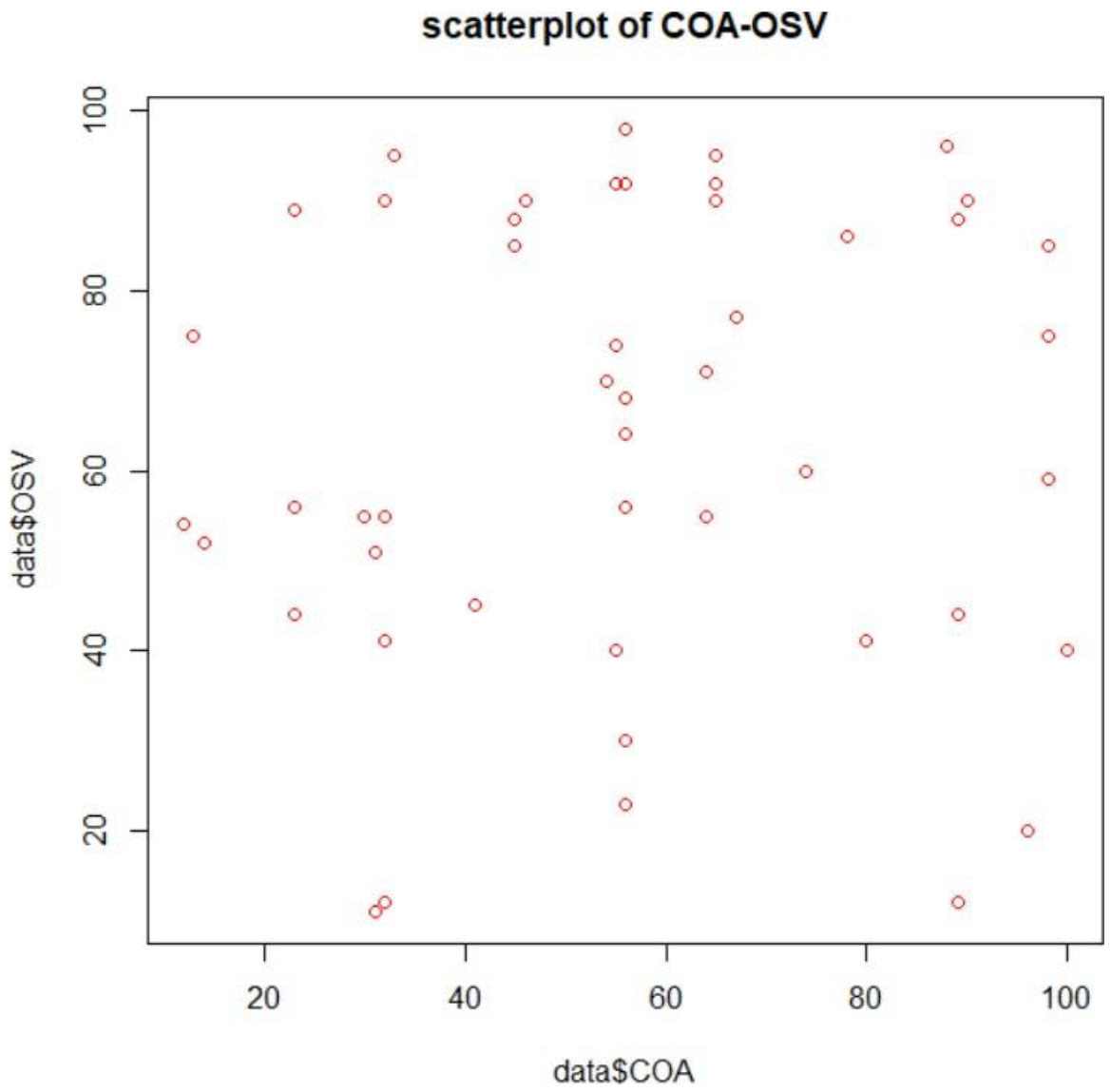
```
plot(data$RESULT,main="scatterplot of RESULT",col="pink")
```



```
par(mfrow = c(2,3))

> plot(data$COA,main="scatterplot of COA",col="red")
  > plot(data$OOP,main="scatterplot of OOP",col="blue")

  > plot(data$DM,main="scatterplot of DM",col="purple")

  > plot(data$PEM,main="scatterplot of PEM",col="orange")

  > plot(data$OSV,main="scatterplot of OSV")

  > plot(data$RESULT,main="scatterplot of RESULT",col="green")
```


scatterplot of COA
data$COA
Index
scatterplot of OOP
data$OOP
Index
scatterplot of DM
data$DM
Index
scatterplot of PEM
data$PEM
Index
scatterplot of OSV
data$OSV
Index
scatterplot of RESULT
data$RESULT
Index

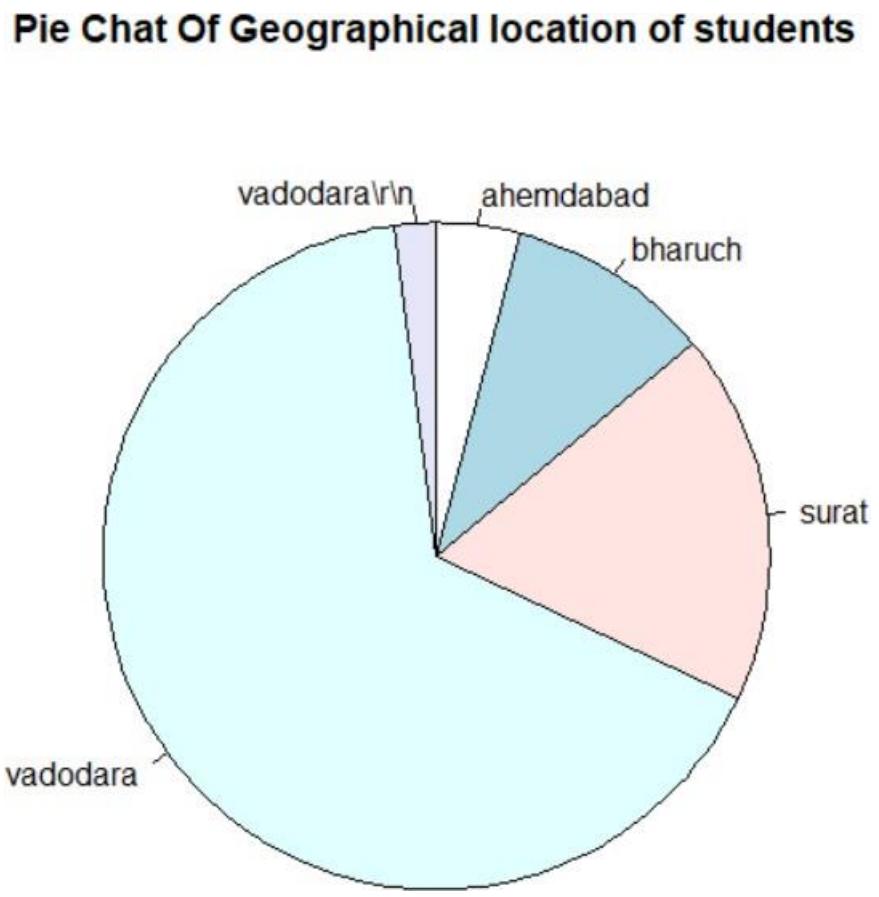
```
plot(data$COA,data$OSV,main="scatterplot of COA-OSV",col="red")
```


scatterplot of COA-OSV
data$OSV
data$COA
20
40
60
80
100
20
40
60
80
100

Practical 6

Aim - Plot the graph showing the geographical location of students

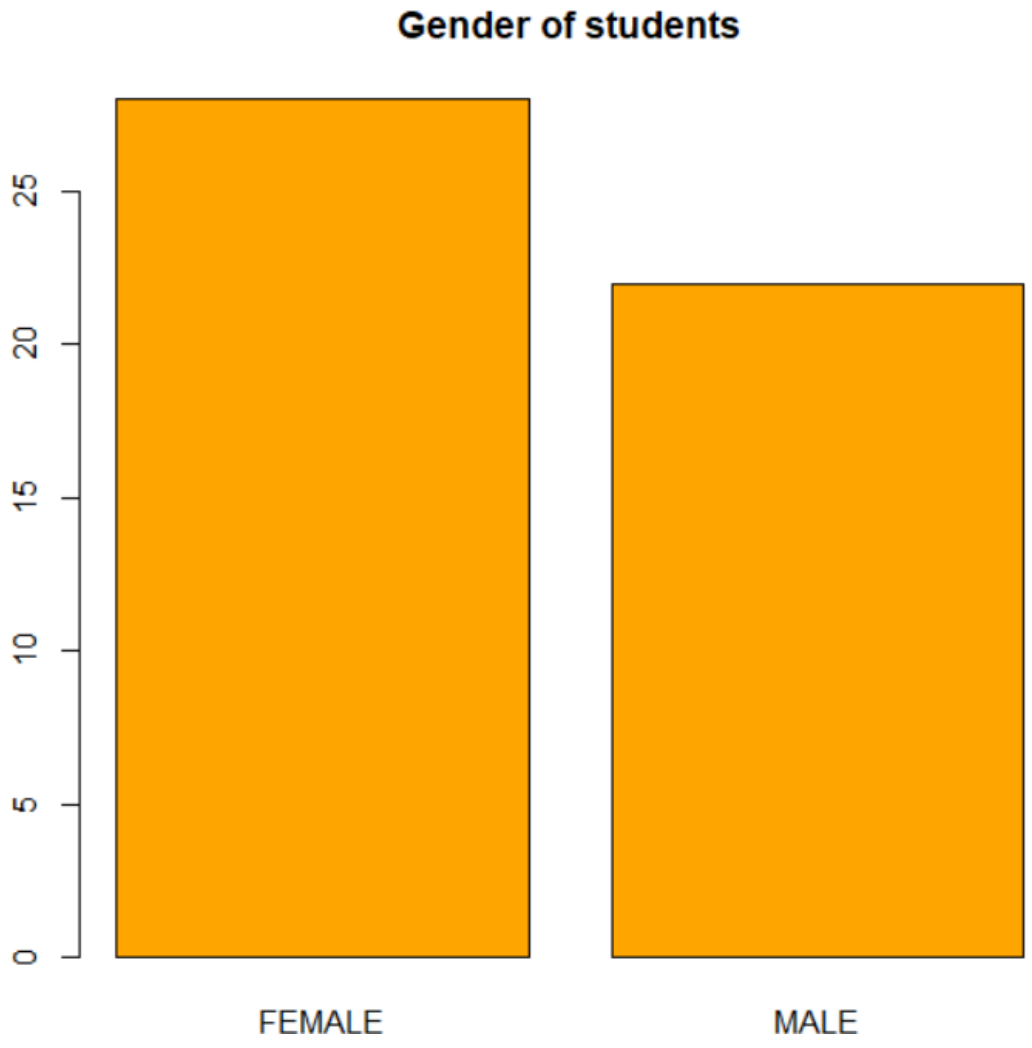
```
pie(table(data$CITY),main="Pie Chat Of Geographical location of students”,
clockwise=True)
```



PRACTICAL 7

Aim - Plot the graph showing number of male and female students.

```
barplot(table(data$GENDER),main="Gender of students",col="orange")
```



8

Practical 8

Aim: Implement a method to treat Missing value for gender and missing value for marks.

```
>       data<-read.csv("Dataset1.csv")

>       print(data)
```

```
NAME GENDER ENNO OOP COA OSV PEM DM RESULT       MOBNO       CITY
  1       AKSHAY MALE 1 88 80 78 68 45 71.8 9545565623 junagadh
  2       JIMI MALE 2 96 82 81 70 49 75.6 9656325623 navsari
  3       AARSH MALE 3 45 84 84 72 53 67.6 8632326656 vadodara
  4       ALPA FEMALE 4 12 86 87 74 57 63.2 7623656566 surat
  5       ANJALI FEMALE 5 46 88 90 76 61 72.2 5569556556 vadodara
```

6 DEVANSH MALE 6 82 90 93 78 65 81.6 9656225452 surat

7 DHRUV MALE 7 32 92 96 80 69 73.8 6852315453 surat

8 HUZEFA MALE 8 98 94 78 82 73 85.0 8656565656 vadodara

9 BHAUTIK MALE 9 56 96 80 84 77 78.6 6532656562 surat

10 SAHIL MALE 10 24 78 82 86 81 70.2 7656565656 surat

11 JAY MALE 11 56 79 84 88 85 78.4 9565656656 surat

12 DHYEY MALE 12 24 80 86 90 89 73.8 8565323623 junagadh

13 JEEL MALE 13 78 81 88 92 93 86.4 7556235635 surat

14 KRISHNA FEMALE 14 46 82 90 94 97 81.8 6992454154 surat

15 AYUSH MALE 15 90 83 92 96 78 87.8 9895565656 vadodara

16 DEVANG MALE 16 78 84 94 98 56 82.0 9681473213 vadodara

17 YASH MALE 17 76 75 84 90 76 80.2 9773170543 navsari

```
18      MEET MALE 18 56 77 45 65 56 59.8 7612655432 surat

19      DHWANI FEMALE 19 99 98 97 96 95 97.0 9134544134 vadodara

20      ARTH MALE 20 46 56 45 35 56 47.6 8535484545 vadodara

21      HARISH MALE 21 12 58 48 38 58 42.8 7234415454 surat

22      BHARGAV <NA> 22 56 60 51 41 60 53.6 7876348753 vadodara

23      DRISHTI FEMALE 23 86 64 57 47 91 69.0 7123235454 surat

24      preeti FEMALE 24 98 97 96 95 94 96.0 8235434354 surat

25      JAIMIN MALE 25 33 66 60 50 66       NA 9345454533 junagadh

26      DEVANKSHU MALE 26 56 68 63 53 68 61.6 8454344352 surat

27      PRASHANT MALE 27 87 70 66 56 70 69.8 9656534322 vadodara

28      KARAN MALE 28 66 72 69 59 72 67.6 8886484354 navsari

29      AKSHIT MALE 29 87 74 72 62 74 73.8 9344545334 surat

30      MUSTAKIM MALE 30 32 76 75 65 76 64.8 9981213123 navsari
>       newdata<-na.omit(data)

>       print(newdata)

NAME GENDER ENNO OOP COA  OSV PEM DM          RESULT MOBNO CITY
```

1 AKSHAY MALE 1 88 80 78 68 45 71.8 9545565623 junagadh

2 JIMI MALE 2 96 82 81 70 49 75.6 9656325623 navsari

3 AARSH MALE 3 45 84 84 72 53 67.6 8632326656 vadodara

4 ALPA FEMALE 4 12 86 87 74 57 63.2 7623656566 surat

5 ANJALI FEMALE 5 46 88 90 76 61 72.2 5569556556 vadodara

6 DEVANSH MALE 6 82 90 93 78 65 81.6 9656225452 surat

7 DHRUV MALE 7 32 92 96 80 69 73.8 6852315453 surat

8 HUZEFA MALE 8 98 94 78 82 73 85.0 8656565656 vadodara

9 BHAUTIK MALE 9 56 96 80 84 77 78.6 6532656562 surat

10 SAHIL MALE 10 24 78 82 86 81 70.2 7656565656 surat

11 JAY MALE 11 56 79 84 88 85 78.4 9565656656 surat

12 DHYEY MALE 12 24 80 86 90 89 73.8 8565323623 junagadh

13 JEEL MALE 13 78 81 88 92 93 86.4 7556235635 surat

14 KRISHNA FEMALE 14 46 82 90 94 97 81.8 6992454154 surat

15 AYUSH MALE 15 90 83 92 96 78 87.8 9895565656 vadodara

16 DEVANG MALE 16 78 84 94 98 56 82.0 9681473213 vadodara

17 YASH MALE 17 76 75 84 90 76 80.2 9773170543 navsari

18 MEET MALE 18 56 77 45 65 56 59.8 7612655432 surat

19 DHWANI FEMALE 19 99 98 97 96 95 97.0 9134544134 vadodara

20 ARTH MALE 20 46 56 45 35 56 47.6 8535484545 vadodara

21 HARISH MALE 21 12 58 48 38 58 42.8 7234415454 surat

23 DRISHTI FEMALE 23 86 64 57 47 91 69.0 7123235454 surat

24 preeti FEMALE 24 98 97 96 95 94 96.0 8235434354 surat

26 DEVANKSHU MALE 26 56 68 63 53 68 61.6 8454344352 surat

27 PRASHANT MALE 27 87 70 66 56 70 69.8 9656534322 vadodara

28 KARAN MALE 28 66 72 69 59 72 67.6 8886484354 navsari

29 AKSHIT MALE 29 87 74 72 62 74 73.8 9344545334 surat

30 MUSTAKIM MALE 30 32 76 75 65 76 64.8 9981213123 navsari

> na.action(newdata) 22 25

22 25

attr(,"class")

[1] "omit"

- Creating a function to identify which student doesn’t have avg marks or gender.

```
>       a<-function(x){
 + if(is.numeric(x))!is.finite(x) else is.na(x)

 + }
```

- Finding the students whose gender data is missing.

```
>       sapply(data$GENDER,a)
```

MALE MALE MALE FEMALE FEMALE MALE MALE MALE MALE MALE MALE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSEMALE MALE FEMALE MALE MALE MALE MALE FEMALE MALE MALE <NA> FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FEMALE FEMALE MALE MALE MALE MALE MALE MALE

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

So from above data we can see that student named bhargav's gender data is missing.

- Finding the students whose result data is missing.

```
>       sapply(data$RESULT,a)
```

[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[25] TRUE FALSE FALSE FALSE FALSE FALSE

So from above data we can see that student named jaimin's result data is missing.

Practical 9

Aim: Implement linear Regression to predict the 5th SEM result of Student.

```
> coa <- c(70,60,65,72,89,95)

> os <- c(80,75,59,95,84,79)

>std <- lm(os~coa)

>std

  Call:

  lm(formula = os ~ coa)

  Coefficients:

  (Intercept) coa

56.8344          0.2905

>   predict(std,data.frame(coa=48),interval="confidence")  fit  lwr

upr

  1 70.77605 37.35674 104.1954

  For predicting sem 5 marks,

>          Sem3 <- c(98,99,89,78)

>          Sem4 <- c(78,94,74,90)

>          sem5 <- lm(Sem4~Sem3)
```

```
>          sem5


  Call:

  lm(formula = Sem4 ~ Sem3)  Coefficients:

(Intercept)              Sem3 90.36364      -0.06993

>         predict(sem5,data.frame(Sem3=99),interval="confidence") fit

            lwr      upr

1 83.44056 48.99185          117.8893
```

Practical 10

Aim: Implement logistic regression and decision tree to classify the Student as average or clever.

- LRmodel <- glm(status~PER,data=training_set,family=binomial) summary(LRmodel)

Call:

glm(formula = status ~ PER, family = binomial, data = training_set) Deviance Residuals:

Min 1Q Median 3Q Max

-3.971e-06 -3.971e-06 -3.971e-06 -3.971e-06 -3.971e-06

Coefficients:

Estimate Std. Error z value Pr(>|z|) (Intercept) -2.557e+01 2.852e+05 0 1

PER 1.076e-16 3.650e+03 0 1

(Dispersion parameter for binomial family taken to be 1) Null deviance: 0.0000e+00 on 13 degrees of freedom Residual deviance: 2.2078e-10 on 12 degrees of freedom AIC: 4

Number of Fisher Scoring iterations: 24

LR.pred = predict(LRmodel, test_set, type="response") LR.pred

1 7 8 12 13

7.884924e-12 7.884924e-12 7.884924e-12 7.884924e-12 7.884924e-12

16

7.884924e-12 glm.pred <- ifelse(LR.pred >= 0.5, "Clever", "Average") glm.pred

1 7 8 12 13 16

"Average" "Average" "Average" "Average" "Average" "Average"

- LRmodel<-glm(status~OSV,data=training_set,family=binomial) summary(LRmodel)

Call:

glm(formula = status ~ OSV, family = binomial, data = training_set) Deviance Residuals:

Min 1Q Median 3Q Max

-3.971e-06 -3.971e-06 -3.971e-06 -3.971e-06 -3.971e-06

Coefficients:

Estimate Std. Error z value Pr(>|z|) (Intercept) -2.557e+01 2.802e+05 0 1

OSV 5.250e-16 3.659e+03 0 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 0.0000e+00 on 13 degrees of freedom Residual deviance: 2.2078e-10 on 12 degrees of freedom

AIC: 4

Number of Fisher Scoring iterations: 24

LR.pred = predict(LRmodel, test_set, type="response") LR.pred

```
1          7          8          12         13
  7.884924e-12 7.884924e-12 7.884924e-12 7.884924e-12 7.884924e-12  16
  7.884924e-12


  glm.pred <- ifelse(LR.pred >= 0.5, "Clever", "Average") glm.pred
1          7          8          12         13         16
"Average"  "Average"  "Average"  "Average"  "Average"  "Average"
  table(test_set$status,glm.pred) glm.pred
  Average Average 0

  Clever 6
  accuracy=(nrow(test_set[test_set$status==glm.pred,])/nrow(test_set))*100
  accuracy
  [1] 0 error=100-accuracy error
  [1] 100
```

Beyond Syllabus

Stock market prediction project with R.

Coal India Company.

```
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing

Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.


R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.


Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> ds<-read.csv("stock.csv")
> head(ds)
        Date      Symbol Series Prev.Close Open      High Low Last Close

1 2007-11-27 MUNDRAPORT     EQ      440.00 770.00 1050.00 770 959 962.90


2 2007-11-28 MUNDRAPORT     EQ      962.90 984.00 990.00 874 885 893.90

3 2007-11-29 MUNDRAPORT     EQ      893.90 909.00 914.75 841 887 884.20

4 2007-11-30 MUNDRAPORT     EQ      884.20 890.00 958.00 890 929 921.55

5 2007-12-03 MUNDRAPORT     EQ      921.55 939.75 995.00 922 980 969.30
```

```
6 2007-12-04 MUNDRAPORT     EQ      969.30 985.00 1056.00 976 1049 1041.45

      VWAP Volume       Turnover Trades Deliverable.Volume X.Deliverble

 1 984.72 27294366 2.687719e+15      NA          9859619       0.3612
 2 941.38 4581338 4.312765e+14       NA          1453278       0.3172
 3 888.09 5124121 4.550658e+14       NA          1069678       0.2088
 4 929.17 4609762 4.283257e+14       NA          1260913       0.2735
 5 965.65 2977470 2.875200e+14       NA          816123        0.2741

 6 1015.39 4849250 4.923867e+14  >   NA          1537667       0.3171
ds<-read.csv("stock.csv")

 > head(ds)

       Date  Open  High Low Close

 1 27/11/2007 770.00 1050.00 770 962.90

 2 28/11/2007 984.00 990.00 874 893.90

 3 29/11/2007 909.00 914.75 841 884.20
 4 30/11/2007 890.00 958.00 890 921.55

 5 03/12/2007 939.75 995.00 922 969.30
 6 04/12/2007 985.00 1056.00 976 1041.45

 > print(colnames(ds))

 [1] "Date" "Open" "High" "Low" "Close"

 > final_ds<-ds[,c("Open","High","Low","Close")]

 > head(final_ds)
```

```
   Open High Low Close
1 770.00 1050.00 770 962.90

2 984.00 990.00 874 893.90

3 909.00 914.75 841 884.20

4 890.00 958.00 890 921.55

5 939.75 995.00 922 969.30

6 985.00 1056.00 976 1041.45

> #DS pre-processing

> summary(final_ds)

    Open          High          Low          Close
Min. : 108.0 Min. : 110.5 Min. : 105.7 Min. : 108.0

1st Qu.: 164.8 1st Qu.: 168.0 1st Qu.: 161.6 1st Qu.: 164.3

Median : 325.8 Median : 331.3 Median : 319.9 Median : 324.7 Mean : 344.8
Mean : 351.6 Mean : 337.5 Mean : 344.2

3rd Qu.: 401.0 3rd Qu.: 407.2 3rd Qu.: 395.0 3rd Qu.: 400.9

Max. :1310.2 Max. :1324.0 Max. :1270.0 Max. :1307.5

> class(final_ds$Open)

[1] "numeric"
> class(final_ds$High)
```

```
[1] "numeric"

> class(final_ds$Low)

[1] "numeric"

> class(final_ds$Close)

[1] "numeric"

> final_ds$Close <- as.numeric(final_ds$Close)

> class(final_ds$Close)
[1] "numeric"

> #Checking Null values

> colSums(is.na(final_ds))
Open High Low Close
0    0         0        0

> boxplot(final_ds$Open)

> head(final_ds)
   Open High Low Close

1    770.00 1050.00 770 962.90

2    984.00 990.00 874 893.90

3    909.00 914.75 841 884.20
```

4 890.00 958.00 890 921.55

5 939.75 995.00 922 969.30

6 985.00 1056.00 976 1041.45

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

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/Harshil/Documents/R/win-library/4.1’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/caTools_1.18.2.

zip' Content type 'application/zip' length 316380 bytes (308 KB)

downloaded 308 KB

package ‘caTools’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in C:\Users\Harshil\AppData\Local\Temp\RtmpKUhzXL\downloaded_packages

> library(caTools) Warning message:

package ‘caTools’ was built under R version 4.1.1

> split = sample.split(final_ds,SplitRatio = 0.7)

```
> train = subset(final_ds,split==TRUE)

> test = subset(final_ds,split==FALSE)
> head(train)

     Open High Low Close

1 770.00 1050.0 770 962.90

4   890.00 958.0 890 921.55

5   939.75 995.0 922 969.30

8  1089.00 1109.7 1051 1081.30

9  1100.00 1134.0 1078 1102.40

12 1032.00 1065.0 1016 1036.80
> head(test)

   Open High Low Close
2   984 990.00 874.0 893.90

3   909 914.75 841.0 884.20

6  985 1056.00 976.0 1041.45

7  1061 1099.50 1050.0 1082.45

10 1110 1110.00 1061.1 1075.40

11 1081 1089.00 1041.0 1047.65
```

```
> lm.r = lm(formula = Open~.,data = train)
> coef(lm.r)
(Intercept)        High         Low       Close
     1.1826417 0.8022306 0.8847939 -0.6888091
> summary(lm.r)

Call:
lm(formula = Open ~ ., data = train)  Residuals:
    Min      1Q Median      3Q  Max
-91.562 -1.983 -0.262 1.718 82.424

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.18264    0.33481 3.532 0.000423 ***  High        0.80223 0.02033
39.459 < 2e-16 ***
Low          0.88479  0.01623 54.517 < 2e-16 ***
Close       -0.68881  0.02648 -26.014 < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.618 on 1657 degrees of freedom Multiple

R-squared: 0.9988, Adjusted R-squared: 0.9988

Fstatistic: 4.713e+05 on 3 and 1657 DF, p-value: < 2.2e-16


> lm(formula = Open ~ ., data = train)

Call:
lm(formula = Open ~ ., data = train)


Coefficients:
(Intercept)        High         Low       Close
     1.1826      0.8022      0.8848     -0.6888


> ypred = predict(lm.r,newdata = test)

> plot(test$Open,ypred)

> ds<-read.csv("stock1.csv")
Error in file(file, "rt") : cannot open the connection

In addition: Warning message:

In file(file, "rt") :

  cannot open file 'stock1.csv': No such file or directory


> head(ds)
```

```
      Date  Open  High Low Close

1 27/11/2007 770.00 1050.00 770 962.90

2 28/11/2007 984.00 990.00 874 893.90

3 29/11/2007 909.00 914.75 841 884.20

4 30/11/2007 890.00 958.00 890 921.55

5 03/12/2007 939.75 995.00 922 969.30

6 04/12/2007 985.00 1056.00 976 1041.45

>

>

> ds<-read.csv("stock1.csv")

Error in file(file, "rt") : cannot open the connection

In addition: Warning message:
In file(file, "rt") :

  cannot open file 'stock1.csv': No such file or directory

> head(ds)

      Date  Open  High Low Close

1 27/11/2007 770.00 1050.00 770 962.90

2 28/11/2007 984.00 990.00 874 893.90

3 29/11/2007 909.00 914.75 841 884.20

4 30/11/2007 890.00 958.00 890 921.55
```

```
  5 03/12/2007 939.75 995.00 922 969.30

  6 04/12/2007 985.00 1056.00 976 1041.45

  >

  >

  > ds<-read.csv("stock1.csv")

  > head(ds)

        Date   Open High       Low Volume X X.1 X.2 X.3 X.4

  1 04/11/2010 291.00 344.90 291.00 479716245 NA NA NA NA NA

  2 05/11/2010 343.00 356.50 343.00 31927173 NA NA NA NA NA

  3 08/11/2010 351.80 355.90 329.50 46932779 NA NA NA NA NA

  4 09/11/2010 330.15 333.40 325.00 23741956 NA NA NA NA NA

  5 10/11/2010 325.40 327.80 320.05 21057129 NA NA NA NA NA

  6 11/11/2010 323.00 336.95 321.85 26548372 NA NA NA NA NA

>

  >
  > print(colnames(ds))
[1] "Date" "Open" "High" "Low"           "Volume" "X"       "X.1" "X.2" "X.3"
  "X.4"

  >

  > [1] "Date" "Open" "High" "Low" "volume"
```

```
 Error: unexpected '[' in "["

 >

 > final_ds<-ds[,c("Open","High","Low","volume")]

 Error in `[.data.frame`(ds, , c("Open", "High", "Low", "volume")) :

  undefined columns selected

 > head(final_ds)

    Open High Low Close

1 770.00 1050.00 770 962.90

2 984.00 990.00 874 893.90

3 909.00 914.75 841 884.20

4 890.00 958.00 890 921.55

5 939.75 995.00 922 969.30

6 985.00 1056.00 976 1041.45

 >

 > final_ds<-ds[,c("Open","High","Low","Volume")]

 > head(final_ds)

      Open High  Low  Volume

1 291.00 344.90 291.00 479716245

2 343.00 356.50 343.00 31927173
3 351.80 355.90 329.50 46932779
```

```
4 330.15 333.40 325.00 23741956

5 325.40 327.80 320.05 21057129

6 323.00 336.95 321.85 26548372

>

> DS pre-processing

Error: unexpected symbol in "DS pre"

> summary(final_ds)

      Open          High          Low          Volume

Min. :110.8 Min.   :112.5 Min. :109.5 Min. :              21437

1st Qu.:254.7 1st Qu.:257.0 1st Qu.:251.2 1st Qu.: 2236566

Median :300.1 Median :303.9 Median :295.9 Median : 3471441

Mean :288.2 Mean :291.9 Mean   :284.2 Mean     5617118

3rd Qu.:341.2 3rd Qu.:345.5 3rd Qu.:336.1 3rd Qu.: 6321588

Max. :445.0 Max. :447.1 Max. :437.0 Max. :479716245

>

> class(final_ds$Open)
[1] "numeric"

> class(final_ds$High)
```

```
[1] "numeric"

>


>

> class(final_ds$Low)
[1] "numeric"

>

>


> class(final_ds$Close)

[1] "NULL"


>

> summary(final_ds)


      Open          High          Low          Volume

Min. :110.8 Min.   :112.5 Min. :109.5 Min. :              21437

 1st Qu.:254.7 1st Qu.:257.0 1st Qu.:251.2 1st Qu.: 2236566


 Median :300.1 Median :303.9 Median :295.9 Median : 3471441

 Mean :288.2 Mean :291.9 Mean :284.2   Mean     5617118

 3rd Qu.:341.2 3rd Qu.:345.5 3rd Qu.:336.1 3rd Qu.: 6321588

 Max. :445.0 Max. :447.1 Max. :437.0 Max. :479716245
> colSums(is.na(final_ds))

  Open High Low Volume

0    0    0      0
```

```
> boxplot(final_ds$Open)

> head(final_ds)

   Open High Low  Volume
1  291.00 344.90 291.00 479716245

2  343.00 356.50 343.00 31927173

3  351.80 355.90 329.50 46932779


4  330.15 333.40 325.00 23741956

5  325.40 327.80 320.05 21057129

6  323.00 336.95 321.85 26548372

>    install.packages("caTools")
Error in install.packages : Updating loaded packages

> library(caTools)

> split = sample.split(final_ds,SplitRatio = 0.7)

> train = subset(final_ds,split==TRUE)

> test = subset(final_ds,split==FALSE)

> head(train)

   Open High Low Volume
```

2 343.00 356.50 343.00 31927173

3 351.80 355.90 329.50 46932779

6 323.00 336.95 321.85 26548372

7 330.00 332.95 318.00 15004107

10 319.60 328.95 310.00 23983896

11 330.25 339.95 328.10 51723739

```
> head(test)
```

```
      Open High Low      Volume
```

1 291.00 344.90 291.00 479716245

4 330.15 333.40 325.00 23741956

5 325.40 327.80 320.05 21057129

8 321.20 322.90 315.05 9917395

9 320.00 323.95 316.10 12100114

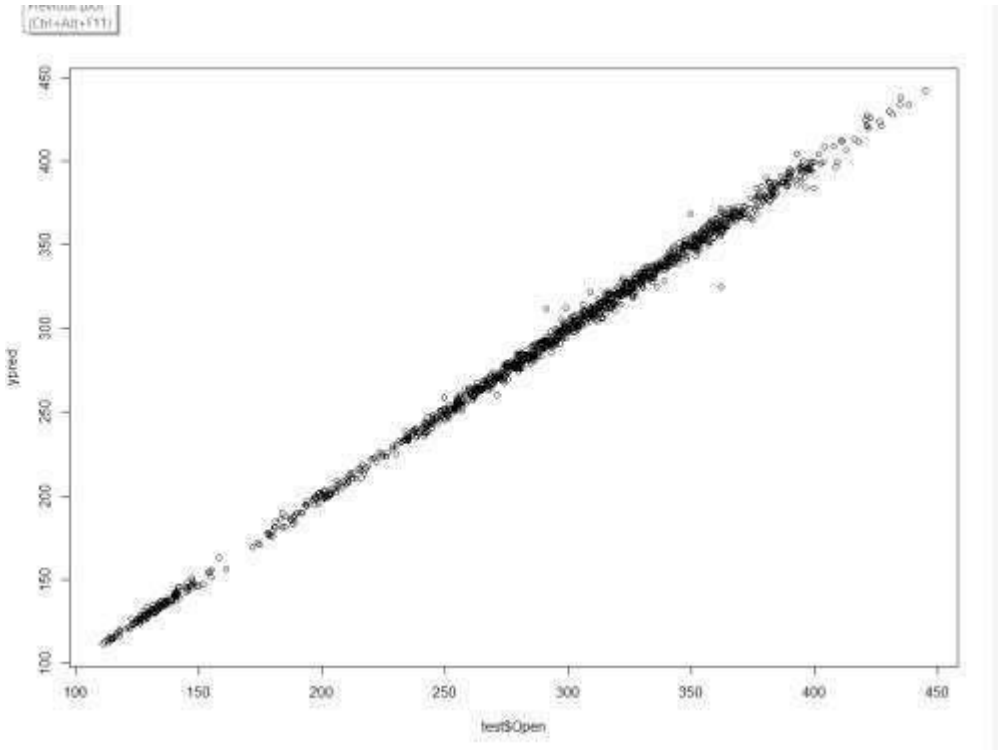
12 336.00 337.00 315.00 11494521

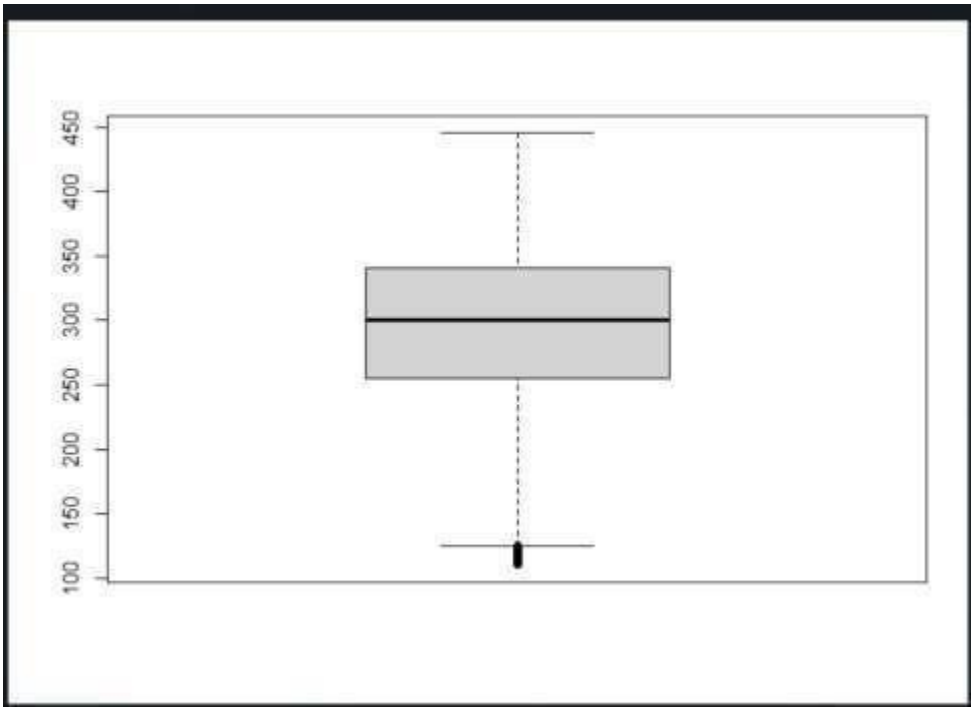
```
> lm.r = lm(formula = Open~.,data = train)
> coef(lm.r)
```

```
  (Intercept)         High          Low       Volume
4.460421e-01 4.333354e-01 5.672740e-01 -6.767636e-09
```

```
>
> summary(lm.r)

Call:
lm(formula = Open ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-30.0409  -1.8223  -0.0346   1.6620  17.9039

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.460e-01  4.563e-01   0.978    0.328
High        4.333e-01  2.149e-02  20.168   <2e-16 ***
Low         5.673e-01  2.228e-02  25.458   <2e-16 ***
Volume     -6.768e-09  1.685e-08  -0.402    0.688
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.186 on 1295 degrees of freedom
Multiple R-squared:  0.9982,    Adjusted R-squared:  0.9982
Fstatistic: 2.339e+05 on 3 and 1295 DF,  p-value: < 2.2e-16
```

```
> lm(formula = Open ~ ., data = train)


Call:

lm(formula = Open ~ ., data = train)


Coefficients:

(Intercept)        High         Low      Volume

   4.460e-01    4.333e-01 5.673e-01 -6.768e-09


>

> ypred = predict(lm.r,newdata = test)

> plot(test$Open,ypred)
```


ypred
test$Open
100
150
200
250
300
350
400
450


100
150
200
250
300
350
400
450