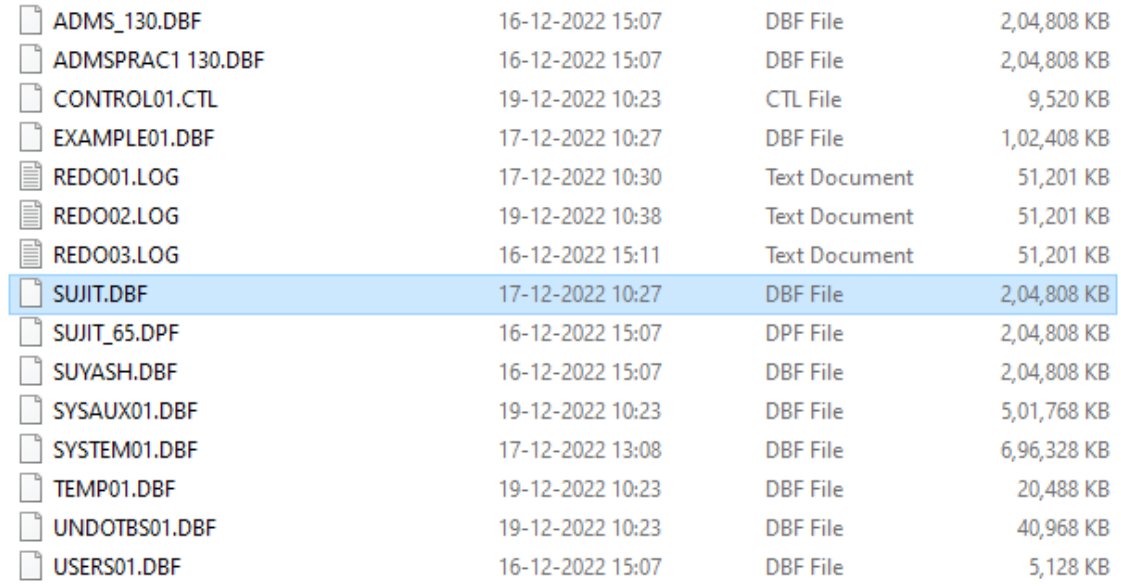
Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 1(A)

# Range partitioning

Q1. Write a query to create a table space for partition

```
SQL> create tablespace Sujit_varghese_65 datafile 'C:\app\Lab2-26\oradata\orcl\Sujit.dbf' size 200m;

Tablespace created.
```



| | | | |
|---|---|---|---|
| ADMS_130.DBF | 16-12-2022 15:07 | DBF File | 2,04,808 KB |
| ADMSPRAC1 130.DBF | 16-12-2022 15:07 | DBF File | 2,04,808 KB |
| CONTROL01.CTL | 19-12-2022 10:23 | CTL File | 9,520 KB |
| EXAMPLE01.DBF | 17-12-2022 10:27 | DBF File | 1,02,408 KB |
| REDO01.LOG | 17-12-2022 10:30 | Text Document | 51,201 KB |
| REDO02.LOG | 19-12-2022 10:38 | Text Document | 51,201 KB |
| REDO03.LOG | 16-12-2022 15:11 | Text Document | 51,201 KB |
| SUJIT.DBF | 17-12-2022 10:27 | DBF File | 2,04,808 KB |
| SUJIT_65.DPF | 16-12-2022 15:07 | DPF File | 2,04,808 KB |
| SUYASH.DBF | 16-12-2022 15:07 | DBF File | 2,04,808 KB |
| SYSAUX01.DBF | 19-12-2022 10:23 | DBF File | 5,01,768 KB |
| SYSTEM01.DBF | 17-12-2022 13:08 | DBF File | 6,96,328 KB |
| TEMP01.DBF | 19-12-2022 10:23 | DBF File | 20,488 KB |
| UNDOTBS01.DBF | 19-12-2022 10:23 | DBF File | 40,968 KB |
| USERS01.DBF | 16-12-2022 15:07 | DBF File | 5,128 KB |

Q2 Write a query for range partitioning on tablespace. Values for partitioning:

- upto 2000
- upto 2005
- upto 2010

Columns for the table:

Product no varchar(2)

Year number(4)

Sales_Amt number(5)

```
SQL> create table part_sale_65(product_no varchar(5),year number(5),sales_amount number(6))
  2  partition by range(year)
  3  (partition p1 values less than(2001) tablespace Sujit_varghese_65,partition p3 values less than(2010) tablespace Su
jit_varghese_65,partition p4 values less than(MAXVALUE) tablespace Sujit_varghese_65);

Table created.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
SQL> insert into part_sale_65 values('p001',2000,1000);

1 row created.

SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000

SQL> insert into part_sale_65 values('p002',2001,2000);

1 row created.

SQL> insert into part_sale_65 values('p003',2002,3000);

1 row created.

SQL> insert into part_sale_65 values('p004',2004,4000);

1 row created.

SQL> insert into part_sale_65 values('p005',2006,5000);

1 row created.

SQL> insert into part_sale_65 values('p006',2009,6000);

1 row created.

SQL> insert into part_sale_65 values('p007',2010,7000);

1 row created.

SQL> insert into part_sale_65 values('p008',2011,8000);

1 row created.
```

Q3 Write a query to display data partition-wise.

```
SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000
p007        2010         7000
p008        2011         8000

8 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
SQL> select *from part_sale_65 partition(p1);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000

SQL> select *from part_sale_65 partition(p2);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p002        2001         2000
p003        2002         3000
p004        2003         4000
p005        2004         5000

SQL> select *from part_sale_65 partition(p3);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p006        2005         6000
p007        2006         7000
p008        2007         8000

SQL> select *from part_sale_65 partition(p4);

no rows selected
```

Q4 Write a query to drop partition p4.

```
SQL> alter table part_sale_65 drop partition(p4);

Table altered.

SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000

6 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

Q5 Write a query to add new partition p5 into part_sales_54 table which has the values less than 2015.

```
SQL> alter table part_sale_65 add partition p5 values less than(2015) tablespace Sujit_varghese_65;

Table altered.

SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000

6 rows selected.
```

```
SQL> insert into part_sale_65 values('p009',2012,9000);

1 row created.

SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000
p009        2012         9000

7 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

Q6 Write a query to rename the partition p5 to p4.

```
SQL> alter table part_sale_65 rename partition p5 to p4;

Table altered.

SQL> select * from part_sale_65;

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p001        2000         1000
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000
p009        2012         9000

7 rows selected.
```

Q7 Write a query to merge partition P3 and P4 into partition P5.

```
SQL> alter table part_sale_65 merge partitions p3,p4 into partition p5;

Table altered.
```

```
SQL> select * from part_sale_65 partition(p5);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000
p009        2012         9000

6 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

Q8 Write a query to split the partition P5 at range 2010 into partition p3 and partition P4.

```
SQL> alter table part_sale_65 split partition p5 at (2010)into (partition p3,partition p4);

Table altered.
```

```
SQL> select * from part_sale_65 partition(p3);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p002        2001         2000
p003        2002         3000
p004        2004         4000
p005        2006         5000
p006        2009         6000

SQL> select * from part_sale_65 partition(p4);

PRODU       YEAR SALES_AMOUNT
----- ---------- ------------
p009        2012         9000
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 1(B)

# List partition

Q1) Write a query to create list partitioning on table dept_54 on columns dept _state and insert the values into the the table.

dept_54

columns:4

dept_no- p001,p002,p003,p004,p005,p006

dept_name=sales,IT,HR,finance,operations,marketing

budget=20,00,25000,10000,20000,10000,30000

dept_state=goa,Delhi,Kerala,haryana,assam,Maharashtra

partition east values(assam, west bengal)

west(goa,maharashtra)

north(delhi,haryana.jammu kashmir)

south(kerala,karnataka,tamil nadu)

(in table format)

Q.1 write a query for creating list partition on table department on column And insertion

```
SQL> create table department_65(department_no varchar2(5),department_name varchar2(10),budget number(10),state varchar2(
20))
  2  partition by list(state)
  3  (partition east values('Assam','West Bengal') tablespace Sujit_varghese_65,partition west values('Goa','Maharastra'
) tablespace Sujit_varghese_65,partition north values('Delhi','Haryana','Jammu kashmir') tablespace Sujit_varghese_65,pa
rtition south values('Kerala','Karnataka','Tamil Nadu') tablespace Sujit_varghese_65);

Table created.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
SQL> insert into department_65 values('D001','sales',20000,'Goa');

1 row created.

SQL> insert into department_65 values('D002','IT',20000,'Delhi');

1 row created.

SQL> insert into department_65 values('D003','HR',10000,'Kerala');

1 row created.

SQL> insert into department_65 values('D004','Finance',20000,'Haryana');

1 row created.
```

```
SQL> insert into department_65 values('D005','Operation',10000,'Assam');

1 row created.

SQL> insert into department_65 values('D006','Marketing',30000,'Maharastra');

1 row created.
```

```
SQL> select * from department_65;

DEPAR DEPARTMENT     BUDGET STATE
----- ---------- ---------- --------------------
D005  Operation       10000 Assam
D001  sales           20000 Goa
D006  Marketing       30000 Maharastra
D002  IT              20000 Delhi
D004  Finance         20000 Haryana
D003  HR              10000 Kerala

6 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

Q2 Write a query to display data partition-wise.

```
SQL> select *from department_65 partition(east);

DEPT_ DEPT_NAME      BUDGET STATE
----- ---------- ---------- --------------------
p005  operations      10000 Assam
p006  marketing       30000 West Bengal

SQL> select *from department_65 partition(west);

DEPT_ DEPT_NAME      BUDGET STATE
----- ---------- ---------- --------------------
p001  sales           20000 Goa

SQL> select *from department_65 partition(north);

DEPT_ DEPT_NAME      BUDGET STATE
----- ---------- ---------- --------------------
p002  it              25000 Delhi

SQL> select *from department_65 partition(south);

no rows selected
```

Q3 Write a query to add the new value for column dept_state.

```
SQL> alter table department_65 modify partition west add values('Gujrat');

Table altered.

SQL> insert into department_65 values('D010','Production',15500,'Gujrat');

1 row created.
```

```
SQL> select * from department_65  partition(west);

DEPAR DEPARTMENT     BUDGET STATE
----- ---------- ---------- --------------------
D001  sales           20000 Goa
D006  Marketing       30000 Maharastra
D010  Production      15500 Gujrat
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

Q4 Write a query to merge partitions north and east as a north-east.

```
SQL> alter table department_65 merge partitions north, east into partition northeast;

Table altered.

SQL> select * from department_65  partition(northeast);

DEPAR DEPARTMENT     BUDGET STATE
----- ---------- ---------- --------------------
D005  Operation       10000 Assam
D002  IT              20000 Delhi
D004  Finance         20000 Haryana
```

Q5 Write a query to split partitions north-east into north and east.

```
SQL> alter table department_65 split partition northeast values('Assam','West Bengal')into (partition east,partition nor
th);

Table altered.

SQL> select * from department_65  partition(north);

DEPAR DEPARTMENT     BUDGET STATE
----- ---------- ---------- --------------------
D002  IT              20000 Delhi
D004  Finance         20000 Haryana

SQL> select * from department_65  partition(east);

DEPAR DEPARTMENT     BUDGET STATE
----- ---------- ---------- --------------------
D005  Operation       10000 Assam
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 2

## Implementation of analytical queries

1. Create table employee_65

```
SQL> create table employee_65 (
  2  emp_no number primary key,
  3  ename varchar(20),
  4  job varchar(20),
  5  manager_no number,
  6  hire_data varchar(20),
  7  salary number(20),
  8  commision number(20),
  9  dept_no number(10)
 10  );

Table created.
```

2. Insert record in employee_65

```
SQL> insert into employee_65 values(001,'jones','developer',233,'12/07/1995',15000,NULL,11);
1 row created.
SQL> insert into employee_65 values(008,'jacob','developer',232,'12/07/1995',15000,NULL,11);
1 row created.
SQL> insert into employee_65 values(002,'rocky','analysis',232,'11/03/1995',45000,NULL,12);
1 row created.
SQL> insert into employee_65 values(003,'jones','data scientiest',233,'13/05/1997',10000,NULL,11);
1 row created.
SQL> insert into employee_65 values(004,'amit','web developer',234,'12/07/2000',55000,NULL,13);
1 row created.
SQL> insert into employee_65 values(005,'johnson','research operation',233,'11/07/1998',15000,NULL,14);
1 row created.
SQL> insert into employee_65 values(006,'akshay','tester',235,'1/06/1999',65000,NULL,12);
1 row created.
SQL> insert into employee_65 values(007,'ricky','tester',236,'12/07/1995',5000,NULL,14);
1 row created.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

3. write a query to display element in employee_65

```
SQL> select * from employee_65;

    EMP_NO ENAME                JOB                  MANAGER_NO
---------- -------------------- -------------------- ----------
HIRE_DATA            SALARY  COMMISION     DEPT_NO
---------- ---------- ---------- ----------
         1 jones                developer                 233
12/07/1995              15000                 11

         8 jacob                developer                 232
12/07/1995              15000                 11

         2 rocky                analysis                  232
11/03/1995              45000                 12

    EMP_NO ENAME                JOB                  MANAGER_NO
---------- -------------------- -------------------- ----------
HIRE_DATA            SALARY  COMMISION     DEPT_NO
---------- ---------- ---------- ----------
         3 jones                data scientiest           233
13/05/1997              10000                 11

         4 amit                 web developer             234
12/07/2000              55000                 13

         5 johnson              research operation        233
11/07/1998              15000                 14

    EMP_NO ENAME                JOB                  MANAGER_NO
---------- -------------------- -------------------- ----------
HIRE_DATA            SALARY  COMMISION     DEPT_NO
---------- ---------- ---------- ----------
         6 akshay               tester                    235
1/06/1999               65000                 12

         7 ricky                tester                    236
12/07/1995               5000                 14

8 rows selected.
```

4. write a to display employee_65 using Rank order by salary

```
SQL> select emp_no,ename,salary,dept_no,Rank()over(order by salary) as "Rank" from employee_65;

    EMP_NO ENAME                SALARY    DEPT_NO       Rank
---------- -------------------- ---------- ---------- ----------
         7 ricky                  5000         14          1
         3 jones                 10000         11          2
         1 jones                 15000         11          3
         5 johnson               15000         14          3
         8 jacob                 15000         11          3
         2 rocky                 45000         12          6
         4 amit                  55000         11          7
         6 akshay                65000         12          8

8 rows selected.
```

5. write a to display employee_65 using Rank order by salary and partition by dept_no

```
SQL> select emp_no,ename,salary,dept_no,rank()over(partition by dept_no order by salary) as "Rank" from employee_65;

    EMP_NO ENAME                SALARY    DEPT_NO       Rank
---------- -------------------- ---------- ---------- ----------
         3 jones                 10000         11          1
         8 jacob                 15000         11          2
         1 jones                 15000         11          2
         4 amit                  55000         11          4
         2 rocky                 45000         12          1
         6 akshay                65000         12          2
         7 ricky                  5000         14          1
         5 johnson               15000         14          2

8 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

6. write a to display employee_65 using Rank order by salary desc and partition by dept_no

```
SQL> select emp_no,ename,salary,dept_no,Rank()over(partition by dept_no order by salary desc) as "Rank" from employee_65;

    EMP_NO ENAME                SALARY    DEPT_NO       Rank
---------- -------------------- ---------- ---------- ----------
         4 amit                  55000         11          1
         1 jones                 15000         11          2
         8 jacob                 15000         11          2
         3 jones                 10000         11          4
         6 akshay                65000         12          1
         2 rocky                 45000         12          2
         5 johnson               15000         14          1
         7 ricky                  5000         14          2

8 rows selected.
```

7. write a to display employee_65 using Dense_Rank order by salary desc and partition by dept_no

```
SQL> select emp_no,ename,salary,Dense_Rank()over(partition by dept_no order by salary desc) as "Rank" from employee_65;

    EMP_NO ENAME                SALARY       Rank
---------- -------------------- ---------- ----------
         4 amit                  55000          1
         1 jones                 15000          2
         8 jacob                 15000          2
         3 jones                 10000          3
         6 akshay                65000          1
         2 rocky                 45000          2
         5 johnson               15000          1
         7 ricky                  5000          2

8 rows selected.
```

8. Display the highest and the lowest salary of the employee within their department order by dept_no and salary. Min and max first value and last value

```
SQL> select emp_no,ename,salary,dept_no,min(salary) keep(dense_rank first order by salary)over(partition by dept_no) as lowest from employee_65 order by dept_no,salary;

    EMP_NO ENAME                SALARY    DEPT_NO     LOWEST
---------- -------------------- ---------- ---------- ----------
         3 jones                 10000         11      10000
         8 jacob                 15000         11      10000
         1 jones                 15000         11      10000
         4 amit                  55000         11      10000
         2 rocky                 45000         12      45000
         6 akshay                65000         12      45000
         7 ricky                  5000         14       5000
         5 johnson               15000         14       5000

8 rows selected.
```



```
SQL> select emp_no,ename,salary,dept_no,max(salary) keep(dense_rank last order by salary)over(partition by dept_no) as highest from employee_65 order by dept_no,salary;

    EMP_NO ENAME                SALARY    DEPT_NO    HIGHEST
---------- -------------------- ---------- ---------- ----------
         3 jones                 10000         11      55000
         8 jacob                 15000         11      55000
         1 jones                 15000         11      55000
         4 amit                  55000         11      55000
         2 rocky                 45000         12      65000
         6 akshay                65000         12      65000
         7 ricky                  5000         14      15000
         5 johnson               15000         14      15000

8 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

9. Write a query to display a row number.

```
SQL> select emp_no,ename,salary,dept_no,row_number()over(order by salary desc) as "row number" from employee_65;

    EMP_NO ENAME                SALARY    DEPT_NO row number
---------- -------------------- ---------- ---------- ----------
         6 akshay                65000         12          1
         4 amit                  55000         11          2
         2 rocky                 45000         12          3
         1 jones                 15000         11          4
         8 jacob                 15000         11          5
         5 johnson               15000         14          6
         3 jones                 10000         11          7
         7 ricky                  5000         14          8

8 rows selected.
```

10. Without Partition

```
SQL> select emp_no,ename,salary,dept_no,row_number()over(order by salary desc) as "row number" from employee_65;

    EMP_NO ENAME                SALARY    DEPT_NO row number
---------- -------------------- ---------- ---------- ----------
         6 akshay                65000         12          1
         4 amit                  55000         11          2
         2 rocky                 45000         12          3
         1 jones                 15000         11          4
         8 jacob                 15000         11          5
         5 johnson               15000         14          6
         3 jones                 10000         11          7
         7 ricky                  5000         14          8

8 rows selected.
```

11. Print the average salary of department.

```
SQL> select emp_no,dept_no,ename,salary,avg(salary)over(partition by dept_no) as average_salary from employee_65;

    EMP_NO    DEPT_NO ENAME                SALARY AVERAGE_SALARY
---------- ---------- -------------------- ---------- --------------
         1         11 jones                 15000          23750
         8         11 jacob                 15000          23750
         3         11 jones                 10000          23750
         4         11 amit                  55000          23750
         6         12 akshay                65000          55000
         2         12 rocky                 45000          55000
         5         14 johnson               15000          10000
         7         14 ricky                  5000          10000

8 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

12. Find out the sum of the salary, dept_no and job wise

```
SQL> select dept_no,job,count(*),sum(salary)over(partition by dept_no) as sum_Salary from employee_65 group by dept_no,job,salary;

   DEPT_NO JOB                  COUNT(*) SUM_SALARY
---------- ------------------- ---------- ----------
        11 data scientiest              1      80000
        11 developer                    2      80000
        11 web developer                1      80000
        12 analysis                     1     110000
        12 tester                       1     110000
        14 reasearch operation          1      20000
        14 tester                       1      20000

7 rows selected.
```

13. Group by function Cube.

```
SQL> select dept_no,job,count(*),sum(salary) from employee_65 group by cube(dept_no,job);

   DEPT_NO JOB                  COUNT(*) SUM(SALARY)
---------- ------------------- ---------- -----------
                                        8      225000
           tester                       2       70000
           analysis                     1       45000
           developer                    2       30000
           web developer                1       55000
           data scientiest              1       10000
           reasearch operation          1       15000
        11                              4       95000
        11 developer                    2       30000
        11 web developer                1       55000
        11 data scientiest              1       10000

   DEPT_NO JOB                  COUNT(*) SUM(SALARY)
---------- ------------------- ---------- -----------
        12                              2      110000
        12 tester                       1       65000
        12 analysis                     1       45000
        14                              2       20000
        14 tester                       1        5000
        14 reasearch operation          1       15000

17 rows selected.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 3

## Implementation of ORDBMS(Object)

Implementation of ORDBMS using ADT(Abstract Data Types), References, etc.

- Abstract data types

1) Create data type type_name and type_address

```
SQL> create type type_sujit65 As object(fname varchar2(20),mname varchar2(20),lname varchar2(20));
  2  /

Type created.
```

```
SQL> create type sujit65_address As object(city varchar2(20),street varchar2(20),pincode varchar2(20));
  2  /

Type created.
```

2. Create table.

```
SQL> create table customer65(cid number(5) primary key,cname type_sujit65,cadd sujit65_address,cphn number(10));

Table created.
```

3. Insert data into the table.

```
SQL> insert into customer65 values (1,type_sujit65('sujit','s','varghese'),sujit65_address('bhandup','mumbai','400078'),7987445477);

1 row created.
```

```
SQL> insert into customer65 values (2,type_sujit65('suyash','p','harad'),sujit65_address('kalyan','mumbai','400605'),7545477454);

1 row created.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

4. Show inserted data from the table.

```
SQL> select * from customer65;

       CID
----------
CNAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
CADD(CITY, STREET, PINCODE)
--------------------------------------------------------------------------------
      CPHN
----------
         1
TYPE_SUJIT65('sujit', 's', 'varghese')
SUJIT65_ADDRESS('bhandup', 'mumbai', '400078')
7987445477


       CID
----------
CNAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
CADD(CITY, STREET, PINCODE)
--------------------------------------------------------------------------------
      CPHN
----------
         2
TYPE_SUJIT65('suyash', 'p', 'harad')
SUJIT65_ADDRESS('kalyan', 'mumbai', '400605')
7545477454
```

5. Describe the table created.

```
SQL> desc customer65;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(5)
 CNAME                                              TYPE_SUJIT65
 CADD                                               SUJIT65_ADDRESS
 CPHN                                               NUMBER(10)
```

6. Display the name of the street from the table where the cid=1.

```
SQL> select c.cadd.city from customer65 c where cid=1;

CADD.CITY
--------------------
bhandup
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
SQL> select c.cname.fname from customer65 c where cid=2;

CNAME.FNAME
--------------------
suyash
```

7. Display the first name from the table where the cid=1.

```
SQL> select cname from customer65;

CNAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
TYPE_SUJIT65('sujit', 's', 'varghese')
TYPE_SUJIT65('suyash', 'p', 'harad')
```

8. Display the full name from the table.

```
SQL> select c.cname.lname from customer65 c;

CNAME.LNAME
--------------------
varghese
harad
```

9. Display the first name, middle name and last name from the table.

```
SQL> select c.cname.fname||' '||c.cname.mname||' '||c.cname.lname from customer65 c;

C.CNAME.FNAME||''||C.CNAME.MNAME||''||C.CNAME.LNAME
------------------------------------------------------------
sujit s varghese
suyash p harad
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# 2. REF and DREF function

## 1. Create object table

```
SQL> create or replace type ANIMAL_TY as object(breed varchar2(25),name varchar2(25),birthdate DATE);
  2  /

Type created.
```

## 2. Create table

```
SQL> create table ANIMAL65 of ANIMAL_TY;

Table created.
```

## 3. Insert rows into table.

```
SQL> insert into ANIMAL65 values(ANIMAL_TY('tommy','German shepherd','04-APR-2003'));

1 row created.

SQL> insert into ANIMAL65 values(ANIMAL_TY('Tom','Persian','30-oct-2006'));

1 row created.
```

REF function

## 4. REF function

```
SQL> select REF(A) from ANIMAL65 A;

REF(A)
--------------------------------------------------------------------------------
0000280209100A2CEA8DBF45B8A0C67C63C7131302AE7B7C389C8D495C8350D75C67DBC4C5004152
E90000

00002802093FDD5C62011F46D3AAA538D81B1CF74FAE7B7C389C8D495C8350D75C67DBC4C5004152
E90001
```

Using DREF function

## 5.Create a table with the name KEEPER.

```
SQL> create table keeper65(keeperName varchar(25),AnimalKept REF ANIMAL_TY);

Table created.
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

6. Describe keeper table

```
SQL> describe keeper65
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 KEEPERNAME                                         VARCHAR2(25)
 ANIMALKEPT                                         REF OF ANIMAL_TY
```

7. Insert values into tha table

```
SQL> insert into keeper65 select 'CATHERINE',REF(A) from ANIMAL65 A where breed='tommy';

1 row created.
```

8. Show values inserted into the table keeper

```
SQL> select * from keeper65;

KEEPERNAME
-------------------------
ANIMALKEPT
--------------------------------------------------------------------------------
CATHERINE
0000220208100A2CEA8DBF45B8A0C67C63C7131302AE7B7C389C8D495C8350D75C67DBC4C5
```

9. Select keeperName, deref(k.Animalkept) from keeper65 k

```
SQL> select keeperName, deref(k.Animalkept) from keeper65 k;

KEEPERNAME
-------------------------
DEREF(K.ANIMALKEPT)(BREED, NAME, BIRTHDATE)
--------------------------------------------------------------------------------
CATHERINE
ANIMAL_TY('tommy', 'German shepherd', '04-APR-03')
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 4

## Introduction to R Programming and data acquisition.

### Basics of R

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

```
> setwd("D:/R Orientation 65")
> getwd()
[1] "D:/R Orientation 65"
```

```
> dir()
character(0)
> ls()
[1] "myString"
> x<-1
> class(x)
[1] "numeric"
> x<-'c'
> is.character(x)
[1] TRUE
> is.integer(x)
[1] FALSE
> y<-'2.14'
> as.integer(y)
[1] 2
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> x<-c(11.3,27.5,33.8)
> y<-vector("logical", length=10)
> length(x)
[1] 3
> y
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> y<-c(4,5,6)
> 5*x
[1]  56.5 137.5 169.0
> x*y
[1]  45.2 137.5 202.8
> x^y
[1] 1.630474e+04 1.572764e+07 1.491077e+09
> m<-matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3)
> m
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> dim(m)
[1] 3 3
> attributes(m)
$dim
[1] 3 3

> m<-matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3,byrow = TRUE)
> m
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> x<-c(1,2,3)
> y<-c(11,12,13)
> cbind(x,y)
     x  y
[1,] 1 11
[2,] 2 12
[3,] 3 13
> rbind(x,y)
  [,1] [,2] [,3]
x    1    2    3
y   11   12   13
>
> p<-3*m
> p
     [,1] [,2] [,3]
[1,]   33   36   39
[2,]  165  180  195
[3,]  198  216  234
> n<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
> q<-m+n
> q
     [,1] [,2] [,3]
[1,]   15   26   37
[2,]   60   75   90
[3,]   72   88  104
```

```
> o<-matrix(c(4,5,6,14,15,16),nrow=3,ncol=2)
> o
     [,1] [,2]
[1,]    4   14
[2,]    5   15
[3,]    6   16
> r<-m %*% o
> r
     [,1] [,2]
[1,]  182  542
[2,]  910 2710
[3,] 1092 3252
> mdash<-t(m)
> mdash
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> s<-matrix(c(4,5,6,14,15,16,24,25,26), nrow=3,ncol=3,byrow=TRUE)
> s_det<-det(s)
> s_det
[1] 1.110223e-14
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> x<-list(1,"p",TRUE,2+4i)
> x
[[1]]
[1] 1

[[2]]
[1] "p"

[[3]]
[1] TRUE

[[4]]
[1] 2+4i

> status<-c("low","high","medium","high","low")
> x<-factor(status, ordered=TRUE,levels=c("low","medium","high"))
> x
[1] low    high   medium high   low
Levels: low < medium < high
```

```
> status<-c("low","high","medium","high","low")
> x<-factor(status, ordered=TRUE,levels=c("low","medium","high"))
> x
[1] low    high   medium high   low
Levels: low < medium < high
> student_id<-c(1,2,3)
> student_names<-c("Ram","Sujit","Laxman")
> position<-c("First","Second","Third")
> data<-data.frame(student_id,student_names,position)
> data
  student_id student_names position
1          1           Ram    First
2          2         Sujit   Second
3          3        Laxman    Third
> data$student_id
[1] 1 2 3
> nrow(data)
[1] 3
> ncol(data)
[1] 3
> names(data)
[1] "student_id"    "student_names" "position"
> smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)
> colnames(smoke) <- c("High","Low","Middle")
> rownames(smoke) <- c("current","former","never")
> smoke <- as.table(smoke)
> smoke
        High Low Middle
current   51  43     22
former    92  28     21
never     68  22      9
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> install.packages("XLConnect")
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/USER/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'rJava'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/rJava_1.0-6.zip'
Content type 'application/zip' length 1245703 bytes (1.2 MB)
downloaded 1.2 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/XLConnect_1.0.7.zip'
Content type 'application/zip' length 22487850 bytes (21.4 MB)
downloaded 21.4 MB

package 'rJava' successfully unpacked and MD5 sums checked
package 'XLConnect' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\USER\AppData\Local\Temp\RtmpEF0QX4\downloaded_packages
> library(XLConnect)
XLConnect 1.0.7 by Mirai Solutions GmbH [aut],
  Martin Studer [cre],
  The Apache Software Foundation [ctb, cph] (Apache POI),
  Graph Builder [ctb, cph] (Curvesapi Java library),
  Brett Woolridge [ctb, cph] (SparseBitSet Java library)
https://mirai-solutions.ch
https://github.com/miraisolutions/xlconnect
>
```

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

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/USER/AppData/Local/R/win-library/4.2'
```

```
> library(readxl)
```

```
> install.packages("writexl")
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/USER/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/writexl_1.4.2.zip'
Content type 'application/zip' length 190418 bytes (185 KB)
downloaded 185 KB

package 'writexl' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\USER\AppData\Local\Temp\RtmpEF0QX4\downloaded_packages
```

```
> library(writexl)
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
  cannot open file 'mydata.csv.csv': No such file or directory
> dataT <- read.table("mydata1.csv", sep =",", header = T)
> dataT
  X   Name Rollno Class Percent.Marks
1 1   Sita     23    IV           67%
2 2   Rita     28     V           98%
3 3 Naresh     69   VII
4 4  Mohit     88     V           95%
>
```

```
> dim(dataT)
[1] 4 5
> head(dataT, 2)
  X Name Rollno Class Percent.Marks
1 1 Sita     23    IV           67%
2 2 Rita     28     V           98%
> tail(dataT, 2)
  X   Name Rollno Class Percent.Marks
3 3 Naresh     69   VII
4 4  Mohit     88     V           95%
> z <- data.frame(a = 5, b = 10, c = pi)
> write.csv(z,file="data.csv")
```

## Reading and writing data from Excel using XLConnect

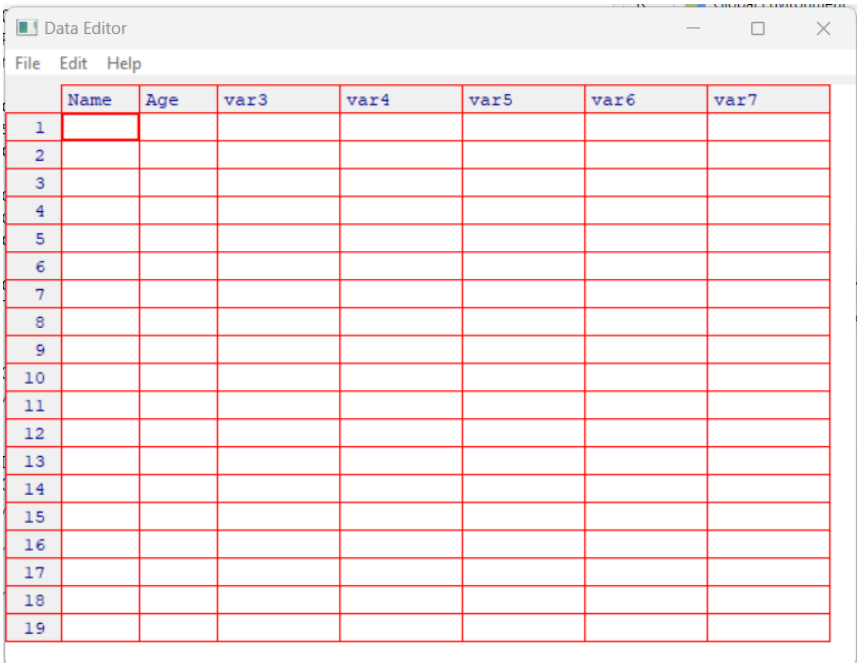
```
> dataX <- XLConnect:: readWorksheetFromFile("employeeinfo.xlsx",sheet=1)
  dataX
```

| | id | name | salary | start_date_of_the_employee_as_per_records | dept |
|---|---|---|---|---|---|
| 1 | 1 | Rick | 623.30 | 2012-01-01 | IT |
| 2 | 2 | Dan | 515.20 | 2013-09-23 | Operations |
| 3 | 3 | Michelle | 611.00 | 2014-11-15 | IT |
| 4 | 4 | Ryan | 729.00 | 2014-05-11 | HR |
| 5 | NA | Gary | 843.25 | 2015-03-27 | Finance |
| 6 | 6 | Nina | 578.00 | 2013-05-21 | IT |
| 7 | 7 | Simon | 632.80 | 2013-07-30 | Operations |
| 8 | 8 | Guru | 722.50 | 2014-06-17 | Finance |
| 9 | 9 | John | 478.00 | 2012-05-21 | <NA> |
| 10 | 10 | Rock | 600.80 | 2013-07-30 | HR |
| 11 | 11 | Brad | 1032.80 | 2013-07-30 | Operations |
| 12 | 12 | Ryan | 729.00 | 2014-05-11 | HR |

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Reading and writing data from Excel using readXL and writeXL

```
> data2 <- read_excel("employeeinfo.xlsx", sheet = 1)
New names:
• `` -> `...1`
> data2
# A tibble: 12 × 6
    ...1 id    name     salary start_date_of_the_employee_as_per_records dept
   <dbl> <chr> <chr>     <dbl> <dttm>                                    <chr>
 1     1 1     Rick       623. 2012-01-01 00:00:00                       IT
 2     2 2     Dan        515. 2013-09-23 00:00:00                       Operations
 3     3 3     Michelle   611  2014-11-15 00:00:00                       IT
 4     4 4     Ryan       729  2014-05-11 00:00:00                       HR
 5     5 NA    Gary       843. 2015-03-27 00:00:00                       Finance
 6     6 6     Nina       578  2013-05-21 00:00:00                       IT
 7     7 7     Simon      633. 2013-07-30 00:00:00                       Operations
 8     8 8     Guru       722. 2014-06-17 00:00:00                       Finance
 9     9 9     John       478  2012-05-21 00:00:00                       <NA>
10    10 10    Rock       601. 2013-07-30 00:00:00                       HR
11    11 11    Brad      1033. 2013-07-30 00:00:00                       Operations
12    12 12    Ryan       729  2014-05-11 00:00:00                       HR
> data3<- data2[1:2,]
> write_xlsx(data3, "e2.xlsx")
> data <- data.frame(Name=character(), Age=numeric())
> data <- edit(data)
>
> data
[1] Name Age
<0 rows> (or 0-length row.names)
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 5

# Implementation of Data Preprocessing techniques.

Implementation of Data preprocessing techniques like,

1. Naming and Renaming variables, adding a new variable.

2. Dealing with missing data.

3. Dealing with categorical data.

4. Data reduction using subsetting.

```
> setwd("F:/R Orientation 65")
> getwd()
[1] "F:/R Orientation 65"
```

```
> my_data<-mtcars
> head(my_data,5)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
>
```

```
> my_data1 <- my_data[1:6,1:5]
> my_data1
                   mpg cyl disp  hp drat
Mazda RX4         21.0   6  160 110 3.90
Mazda RX4 Wag     21.0   6  160 110 3.90
Datsun 710        22.8   4  108  93 3.85
Hornet 4 Drive    21.4   6  258 110 3.08
Hornet Sportabout 18.7   8  360 175 3.15
Valiant           18.1   6  225 105 2.76
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> require(dplyr)
Loading required package: dplyr

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
> my_data1 = rename(my_data1, horse_power = hp)
> my_data1
                   mpg cyl disp horse_power drat
Mazda RX4         21.0   6  160         110 3.90
Mazda RX4 Wag     21.0   6  160         110 3.90
Datsun 710        22.8   4  108          93 3.85
Hornet 4 Drive    21.4   6  258         110 3.08
Hornet Sportabout 18.7   8  360         175 3.15
Valiant           18.1   6  225         105 2.76
```

```
> my_data1$new_hp1 <- my_data1$horse_power * 0.5
> colnames(my_data1)
[1] "mpg"        "cyl"        "disp"       "horse_power" "drat"
[6] "new_hp1"
>
```

```
> my_data1
                   mpg cyl disp horse_power drat new_hp1
Mazda RX4         21.0   6  160         110 3.90    55.0
Mazda RX4 Wag     21.0   6  160         110 3.90    55.0
Datsun 710        22.8   4  108          93 3.85    46.5
Hornet 4 Drive    21.4   6  258         110 3.08    55.0
Hornet Sportabout 18.7   8  360         175 3.15    87.5
Valiant           18.1   6  225         105 2.76    52.5
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> data2 = read.table(file="D:/Rohit1_62/missing_col1.csv", sep = ",")
> data2
   V1          V2      V3          V4          V5
1   1        Rick  623.30 01-01-2012          IT
2   2         Dan  515.20 23-09-2013  Operations
3   3   Michelle   611.00 15-11-2014          IT
4   4        Ryan  729.00 11-05-2014          HR
5  NA        Gary  843.25 27-03-2015     Finance
6   6       Nina       NA 21-05-2013          IT
7   7     Simon    632.80 30-07-2013  Operations
8   8        Guru  722.50 17-06-2014    Finance
9   9       John       NA 21-05-2012
10 10        Rock  600.80 30-07-2013          HR
11 11       Brad  1032.80 30-07-2013  Operations
12 12        Ryan  729.00 11-05-2014          HR
> 
```

```
> data2 = read.csv(file="D:/Rohit1_62/missing_col1.csv", col.names=c("Sno", "NAME","SAL
ARY","DateOfJoin", "Department"))
> data2
   Sno        NAME  SALARY DateOfJoin Department
1    2         Dan  515.20 23-09-2013  Operations
2    3   Michelle   611.00 15-11-2014          IT
3    4        Ryan  729.00 11-05-2014          HR
4   NA        Gary  843.25 27-03-2015     Finance
5    6       Nina       NA 21-05-2013          IT
6    7     Simon    632.80 30-07-2013  Operations
7    8        Guru  722.50 17-06-2014    Finance
8    9       John       NA 21-05-2012
9   10        Rock  600.80 30-07-2013          HR
10  11       Brad  1032.80 30-07-2013  Operations
11  12        Ryan  729.00 11-05-2014          HR
> 
```

```
> V <- c(1,2,NA,3)
>
> median(V)
[1] NA
```

```
> median(V, na.rm = T)
[1] 2
```

```
> is.na(V)
[1] FALSE FALSE  TRUE FALSE
```

```
> naVals <- is.na(V)
> V[!naVals]
[1] 1 2 3
> 
```

```
> V[complete.cases(V)]
[1] 1 2 3
> 
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> dataC <- read.csv(file ="D:/Rohit1_62/na_data.csv", na.strings = "")
> dataC
   X1       Rick  X623.3 X01.01.2012         IT
1   2        Dan  515.20  23-09-2013 Operations
2   3  Michelle   611.00  15-11-2014         IT
3   4       Ryan  729.00  11-05-2014         HR
4  NA       Gary  843.25  27-03-2015    Finance
5   6      Nina       NA  21-05-2013         IT
6   7     Simon    632.80  30-07-2013 Operations
7   8       Guru  722.50  17-06-2014   Finance
8   9      John       NA  21-05-2012       <NA>
9  10       Rock  600.80  30-07-2013         HR
10 11      Brad  1032.80  30-07-2013 Operations
11 12       Ryan  729.00  11-05-2014         HR
```

```
> dataCompleteCases <- dataC[complete.cases(dataC),]
> dataCompleteCases
   X1       Rick X623.3 X01.01.2012         IT
1   2        Dan  515.2  23-09-2013 Operations
2   3  Michelle   611.0  15-11-2014         IT
3   4       Ryan  729.0  11-05-2014         HR
6   7     Simon    632.8  30-07-2013 Operations
7   8       Guru  722.5  17-06-2014   Finance
9  10       Rock  600.8  30-07-2013         HR
10 11      Brad  1032.8  30-07-2013 Operations
11 12       Ryan  729.0  11-05-2014         HR
```

Imputation

```
> library(Hmisc)
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2

Attaching package: 'Hmisc'

The following objects are masked from 'package:dplyr':

    src, summarize

The following objects are masked from 'package:base':

    format.pval, units
```

```
> x = c(1,2,3,NA,4,4,NA)
> x <- impute(x, fun = mean)
> x
   1    2    3    4    5    6    7
 1.0  2.0  3.0 2.8*  4.0  4.0 2.8*
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> x <- impute(x, fun = median)
> x
   1    2    3    4    5    6    7
 1.0  2.0  3.0 2.8*  4.0  4.0 2.8*
> |
```

```
> gender_vector <- c("Male", "Female", "Female", "Male", "Male")
> class(gender_vector)
[1] "character"
> |
```

```
> factor_gender_vector <-factor(gender_vector)
> class(factor_gender_vector)
[1] "factor"
> |
```

```
> day_vector <- c('evening', 'morning', 'afternoon', 'midday', 'midnight', 'evening')
> factor_day <- factor(day_vector, order = TRUE, levels =c('morning', 'midday', 'aftern
oon', 'evening'))
> factor_day
[1] evening   morning   afternoon midday    <NA>      evening
Levels: morning < midday < afternoon < evening
> |
```

```
> age <- c(40, 49, 48, 40, 67, 52, 53)
> salary <- c(103200, 106200, 150200, 10606, 10390, 14070, 10220)
> gender <- c("male", "male", "transgender","female", "male", "female", "transgender")
> employee<- data.frame(age, salary, gender)
> employee
  age salary      gender
1  40 103200        male
2  49 106200        male
3  48 150200 transgender
4  40  10606      female
5  67  10390        male
6  52  14070      female
7  53  10220 transgender
> |
```

```
> wfact = cut(employee$age, 3, labels=c('Young', 'Medium', 'Aged'))
> table(wfact)
wfact
 Young Medium   Aged
     4      2      1
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 6

## Implementation and analysis of Linear regression through graphical methods including Plots

## Regression Analysis:

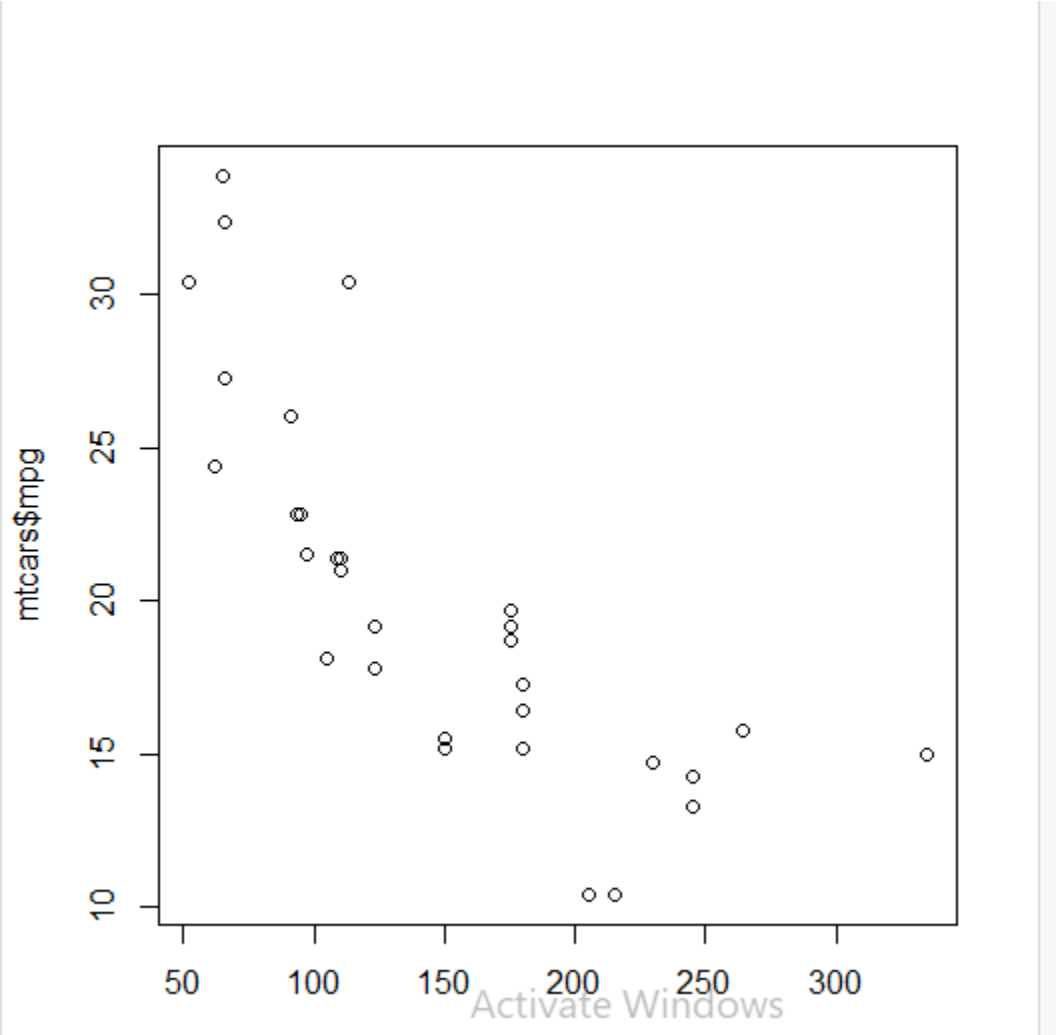
Implementation and analysis of Linear regression through graphical methods including Plots.

```
> library(ggplot2)
> my_data <-mtcars
> names(my_data)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
> dim(my_data)
[1] 32 11
> my_data <- my_data[sample(nrow(my_data), ), ]
> head(my_data)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
>
```

```
> TrainData <- my_data[1:20,]
> TestData <- my_data[21:32,]
> fit = lm(mpg ~ hp, data=mtcars)
> summary(fit)

Call:
lm(formula = mpg ~ hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7121 -2.1122 -0.8854  1.5819  8.2360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.09886    1.63392  18.421  < 2e-16 ***
hp          -0.06823    0.01012  -6.742 1.79e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.863 on 30 degrees of freedom
Multiple R-squared:  0.6024,    Adjusted R-squared:  0.5892
F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07
```

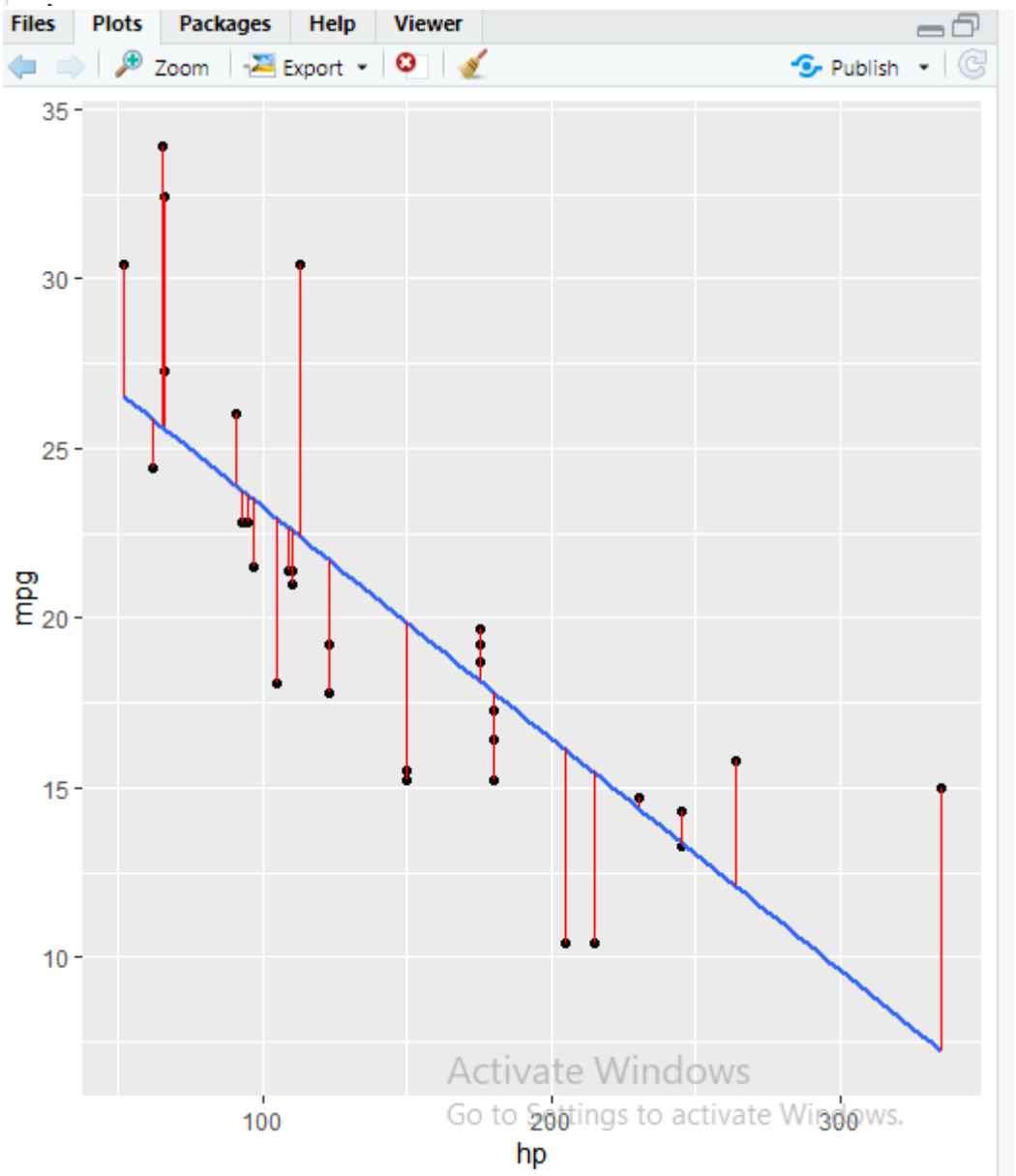
Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> preds <- predict(fit, newdata = TestData)
> df1 <- data.frame(preds,TestData$mpg)
> head(df1)
                    preds TestData.mpg
Dodge Challenger 19.86462         15.5
Porsche 914-2    23.89009         26.0
Ford Pantera L   12.08660         15.8
Ferrari Dino     18.15891         19.7
Duster 360       13.38293         14.3
Merc 450SLC      17.81777         15.2
> cor(preds,TestData$mpg)
[1] 0.7982277
>
> plot(mtcars$hp, mtcars$mpg)
```



Name: sujit saji Varghese
Roll no:65
FYMCA-A

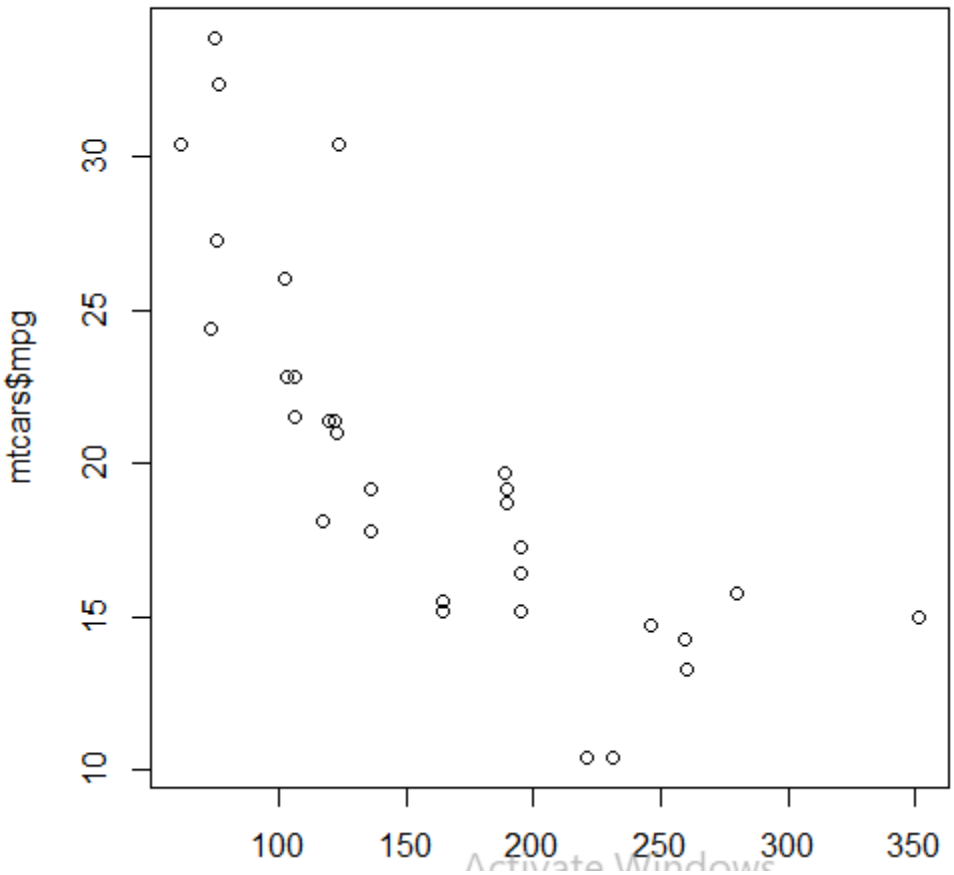
```
> ggplot(fit, aes(hp, mpg)) +
+ geom_point() +
+ stat_smooth(method = lm, se = FALSE) +
+ geom_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
>
```



Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> preds_new <- predict(lmmodel1, newdata = TestData)
> df2 <- data.frame(preds_new,TestData$mpg)
> head(df2)
                  preds_new TestData.mpg
Dodge Challenger  17.40926         15.5
Porsche 914-2     27.99226         26.0
Ford Pantera L    18.59358         15.8
Ferrari Dino      23.04961         19.7
Duster 360        14.70188         14.3
Merc 450SLC       15.89388         15.2
> cor(preds_new,TestData$mpg)
[1] 0.8576077
```

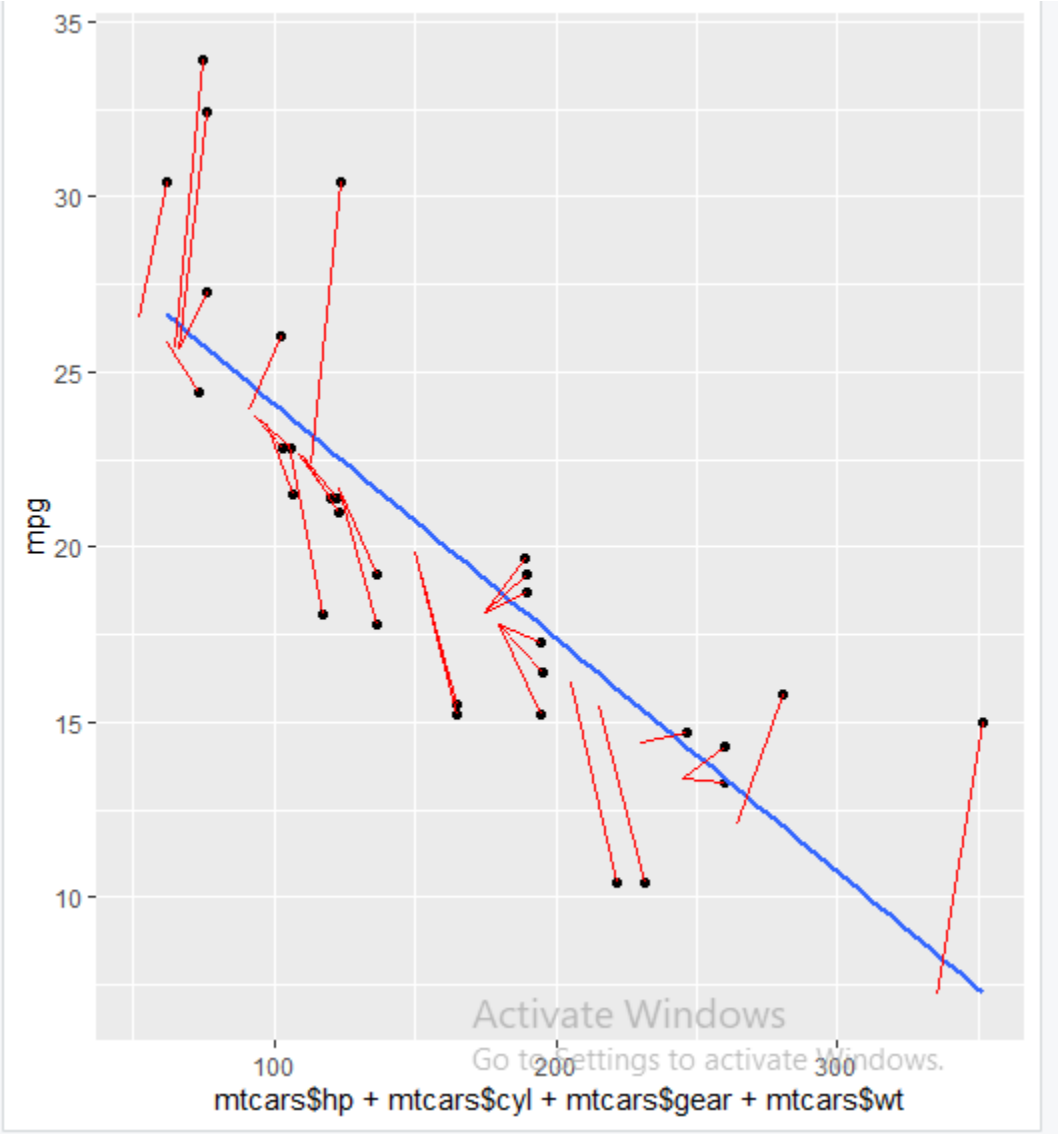
```
> plot(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mtcars$mpg)
```






```
> plot(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mtcars$mpg)
> ggplot(fit, aes(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mpg)) +
+     geom_point() +
+     stat_smooth(method = lm, se = FALSE) +
+     geom_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)
`geom_smooth()` using formula = 'y ~ x'
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A



Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 7

## Implementation and analysis of Classification and algorithms.

Classification:

Implementation and analysis of Classification algorithms like

1. Naive Bayesian,

2. K-Nearest Neighbor

3. ID3

4. C4.5

Naive Bayes

• Based on the Bayes theorem

• Predicts based on probabilities from training

data $P(B|A) = P(A|B)\ P(B)/P(A)$

Gives posterior probability of ‘B’ given ‘A’ using prior probability of ‘B’ prior probability of ‘A’ and conditional probability of ‘A’ given ‘B’.

• Takes two step approach

– Calculates the posterior probability of the Class given the input - for every class

– Assigns the class with higher posterior probability

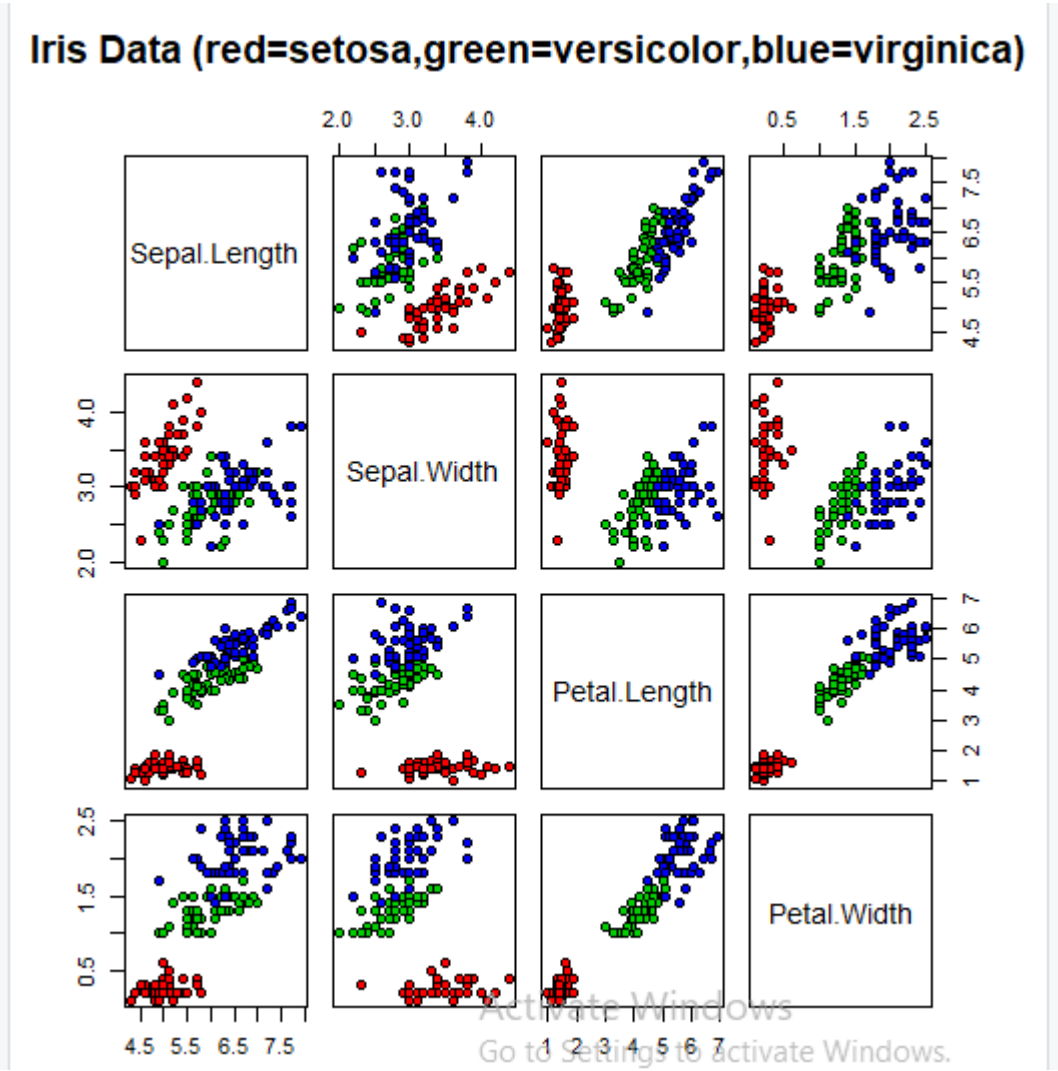
• More suited when dimensionality of input is high the - widely used for document classification

• Also good for the multiclass classifications

• Works well with less datasets also, but the assumption that predictor variables are independent should hold.

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> library(e1071)
> library("klaR")
Loading required package: MASS
> library("caret")
Loading required package: lattice
> library(ggplot2)
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> unique(iris$Species)
[1] setosa     versicolor virginica
Levels: setosa versicolor virginica
> pairs(iris[1:4], main="Iris Data (red=setosa,green=versicolor,blue=virginica)",
+       pch=21, bg=c("red","green3","blue")[unclass(iris$Species)])
```



Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> index = sample(nrow(iris), floor(nrow(iris) * 0.7))
> train = iris[index,]
> test = iris[-index,]
> xTrain = train[,-5]
> yTrain = train$Species
> xTest = test[,-5]
> yTest = test$Species
> model = train(xTrain,yTrain,'nb',trControl=trainControl(method='cv',number=10))
> model
Naive Bayes

105 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 96, 95, 94, 94, 95, 93, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.9825758  0.9735759
   TRUE      0.9916667  0.9875000

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was
 held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

```
> prop.table(table(predict(model$finalModel,xTest)$class,yTest))
            yTest
                 setosa versicolor  virginica
  setosa     0.28888889 0.00000000 0.00000000
  versicolor 0.00000000 0.28888889 0.02222222
  virginica  0.00000000 0.06666667 0.33333333
> df <- data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> ran <- sample(1:nrow(iris), 0.9 * nrow(iris))
>
> nor <-function(x) { (x -min(x))/(max(x)-min(x)) }
> iris_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], nor))
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# K Nearest Neighbour

```
> summary(iris_norm)
  Sepal.Length     Sepal.Width      Petal.Length     Petal.Width
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333
 Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000
 Mean   :0.4287   Mean   :0.4406   Mean   :0.4675   Mean   :0.45806
 3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
> iris_train <- iris_norm[ran,]
> iris_test <- iris_norm[-ran,]
> iris_target_category <- iris[ran,5]
> iris_test_category <- iris[-ran,5]
> library(class)
> pr <- knn(iris_train,iris_test,cl=iris_target_category,k=13)
> tab <- table(pr,iris_test_category)
> accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
> accuracy(tab)
[1] 93.33333
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 8

## Implementation and analysis of Apriori Algorithm using Market Basket Analysis

```
> setwd("D:/R Orientation 65");
> mba_data<-read.csv("data_apriori.csv")
> trans <- split(mba_data$Products, mba_data$Customer_Id,"transactions")
> head(trans)
$`1`
[1] "bread"  "butter" "eggs"   "milk"

$`2`
[1] "beer"   "bread"  "cheese" "chips"  "mayo"   "soda"

$`3`
[1] "bread"   "butter"  "eggs"    "milk"    "oranges"

$`4`
[1] "bread"  "butter" "eggs"   "milk"   "soda"

$`5`
[1] "buns"    "chips"   "beer"    "mustard" "pickels" "soda"

$`6`
[1] "bread"     "butter"    "chocolate" "eggs"      "milk"
```

```
> library(arules)
>
> rules = apriori(trans, parameter=list(support=0.5, confidence=0.9,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.9    0.1    1 none FALSE            TRUE       5     0.5      2      3  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 7

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 15 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [11 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
Warning messages:
1: In asMethod(object) : removing duplicated items in transactions
2: In apriori(trans, parameter = list(support = 0.5, confidence = 0.9,  :
  Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> inspect(rules)
     lhs               rhs      support   confidence coverage  lift     count
[1]  {eggs}         => {milk}   0.6000000 1          0.6000000 1.666667 9
[2]  {milk}         => {eggs}   0.6000000 1          0.6000000 1.666667 9
[3]  {butter}       => {bread}  0.6000000 1          0.6000000 1.250000 9
[4]  {butter, eggs} => {milk}   0.5333333 1          0.5333333 1.666667 8
[5]  {butter, milk} => {eggs}   0.5333333 1          0.5333333 1.666667 8
[6]  {bread, eggs}  => {milk}   0.5333333 1          0.5333333 1.666667 8
[7]  {bread, milk}  => {eggs}   0.5333333 1          0.5333333 1.666667 8
[8]  {butter, eggs} => {bread}  0.5333333 1          0.5333333 1.250000 8
[9]  {bread, eggs}  => {butter} 0.5333333 1          0.5333333 1.666667 8
[10] {butter, milk} => {bread}  0.5333333 1          0.5333333 1.250000 8
[11] {bread, milk}  => {butter} 0.5333333 1          0.5333333 1.666667 8
>
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

# Practical 9

## Implementation and analysis of clustering and algorithm.

Implementation and analysis of clustering algorithms like

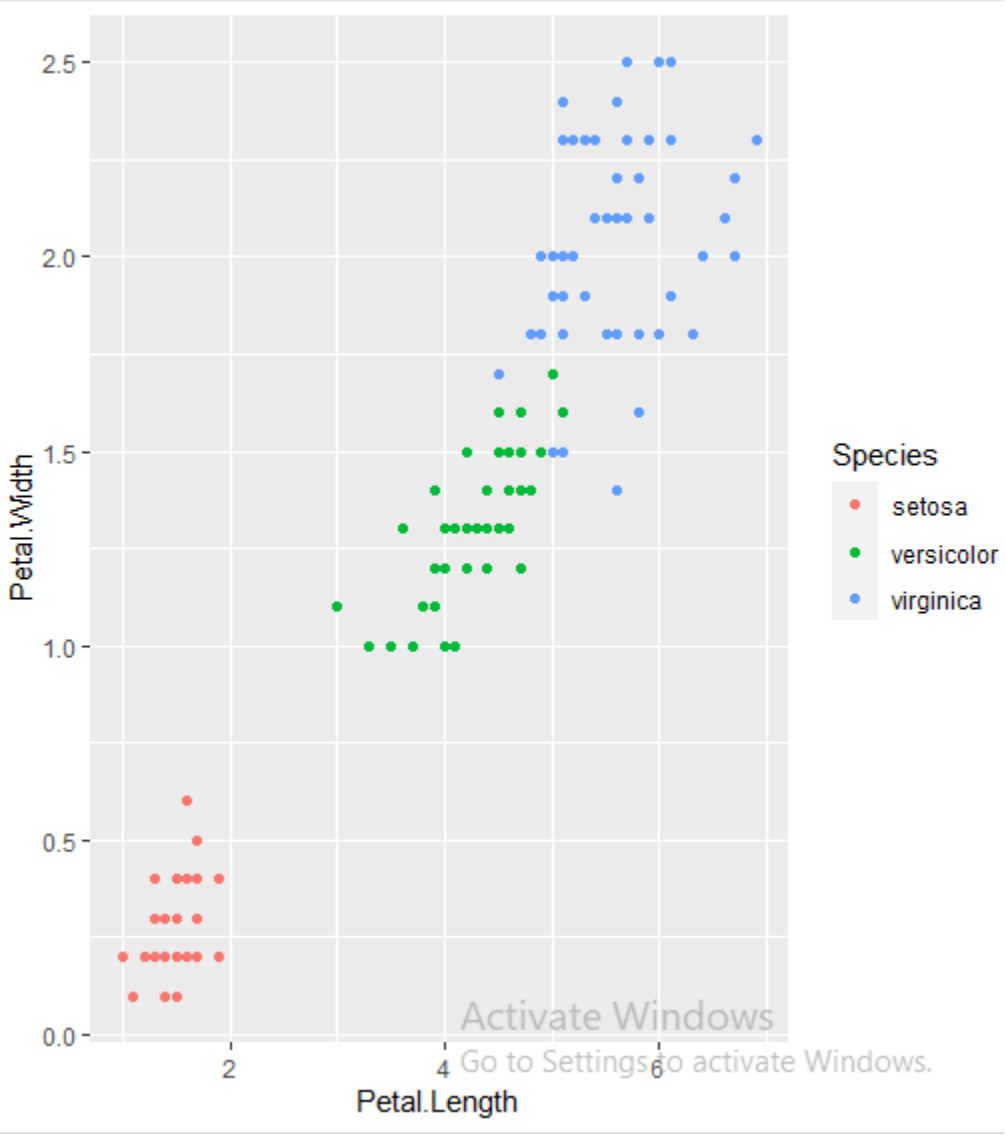
1. K-Means

2. Agglomerative

**K Means Clustering**

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> library(ggplot2)
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom_point()
>
```

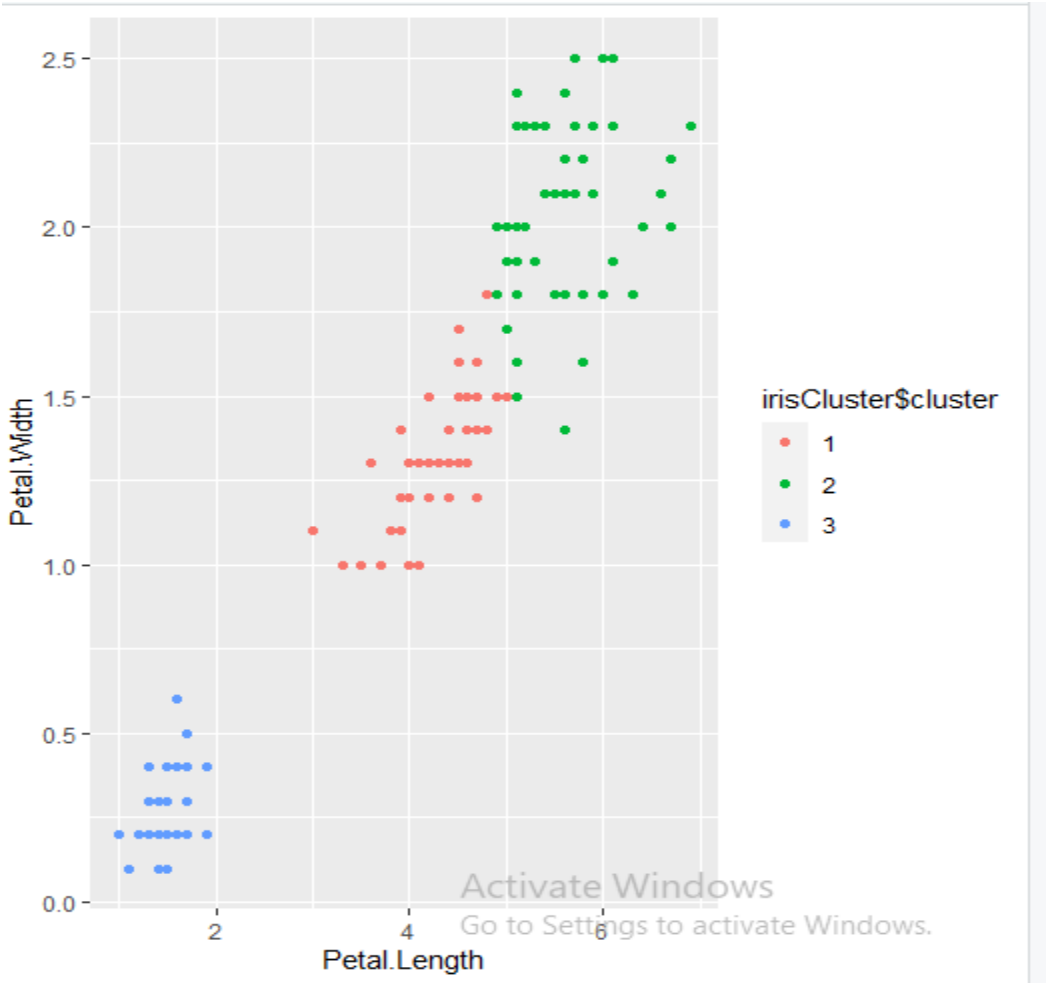
Name: sujit saji Varghese
Roll no:65
FYMCA-A



Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> set.seed(20)
> irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)
> irisCluster
K-means clustering with 3 clusters of sizes 52, 48, 50

Cluster means:
  Petal.Length Petal.Width
1     4.269231    1.342308
2     5.595833    2.037500
3     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [46] 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1
 [91] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2
[136] 2 2 2 1 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 13.05769 16.29167  2.02200
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

```
> irisCluster$cluster <- as.factor(irisCluster$cluster)
> ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom_point()
```

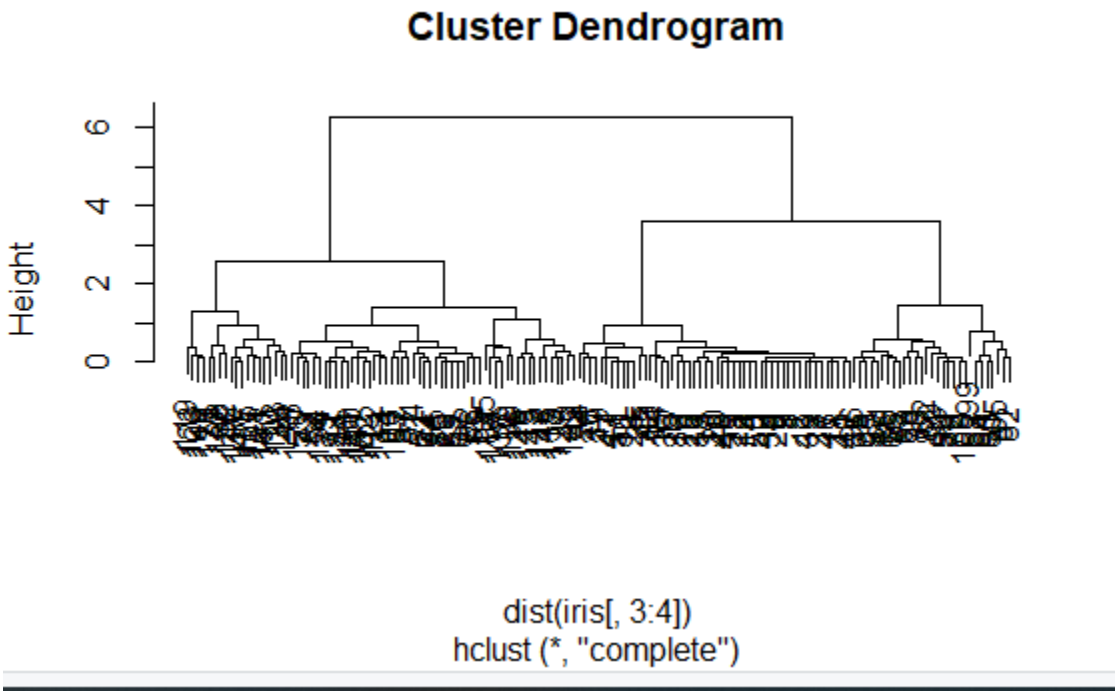


Name: sujit saji Varghese
Roll no:65
FYMCA-A

**Agglomerative Clustering**

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

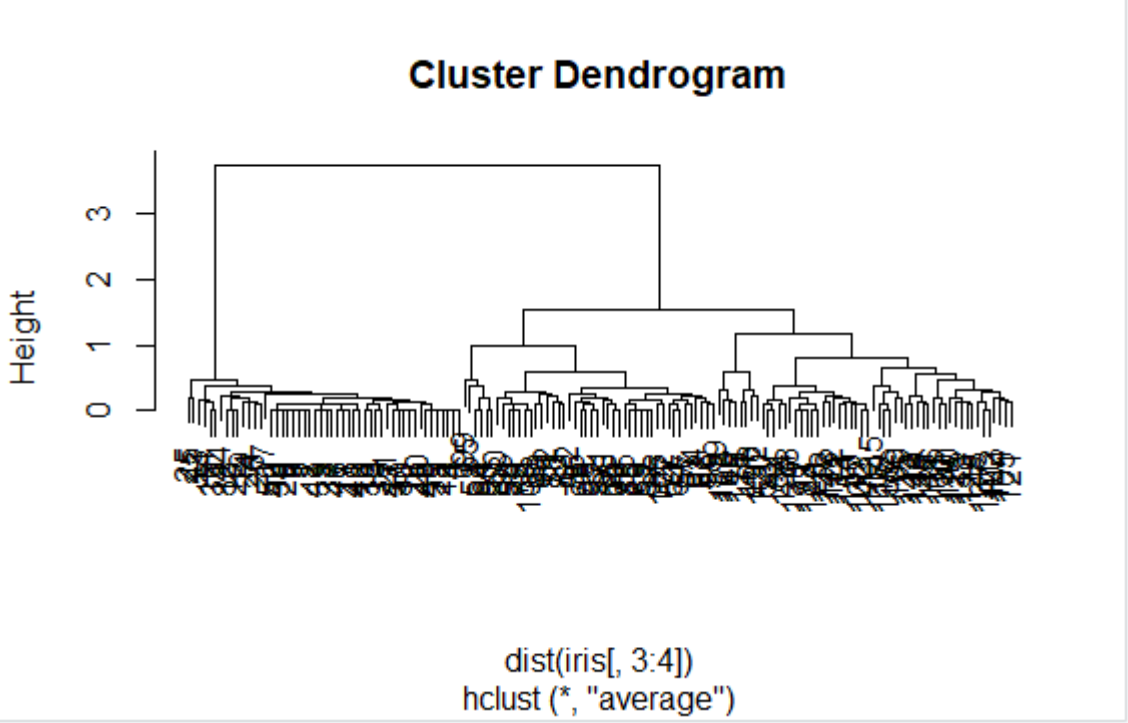
```
> clusters <- hclust(dist(iris[, 3:4]))
> plot(clusters)
>
```



Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> clusterCut<- cutree(clusters, 3)
> table(clusterCut, iris$Species)

clusterCut setosa versicolor virginica
         1     50          0         0
         2      0         21        50
         3      0         29         0
> |
```

```
> clusters <- hclust(dist(iris[,3:4]), method = 'average')
> plot(clusters)
> |
```




```
> clusterCut <- cutree(clusters, 3)
> table(clusterCut,iris$Species)

clusterCut setosa versicolor virginica
         1     50          0         0
         2      0         45         1
         3      0          5        49
> |
```

Name: sujit saji Varghese
Roll no:65
FYMCA-A

```
> ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) + geom_point(alpha = 0.
4, size = 3.5) + geom_point(col = clusterCut) + scale_color_manual(values = c('black','red',
 'green'))
```