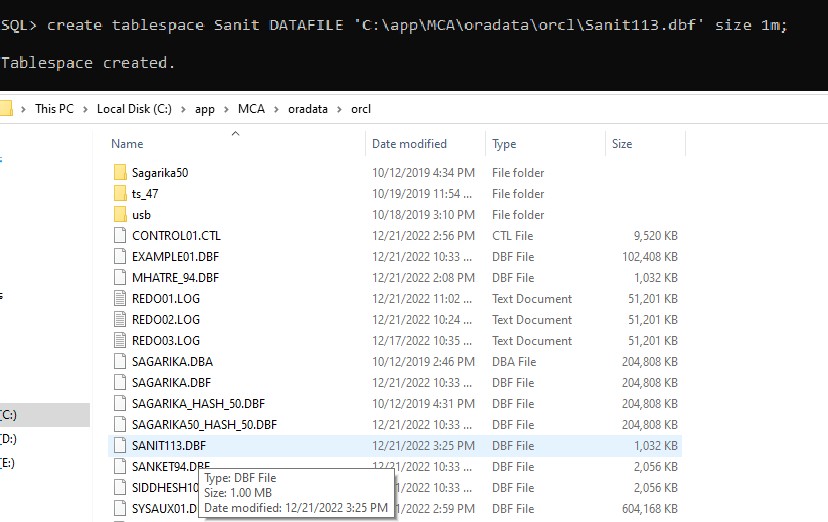
# Practical No: 01 (A)

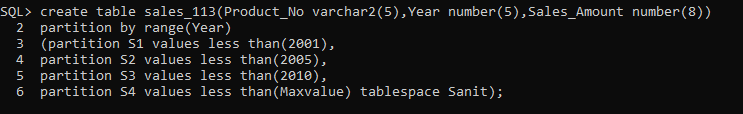
**Q1] Write a query for creating table space to partition**

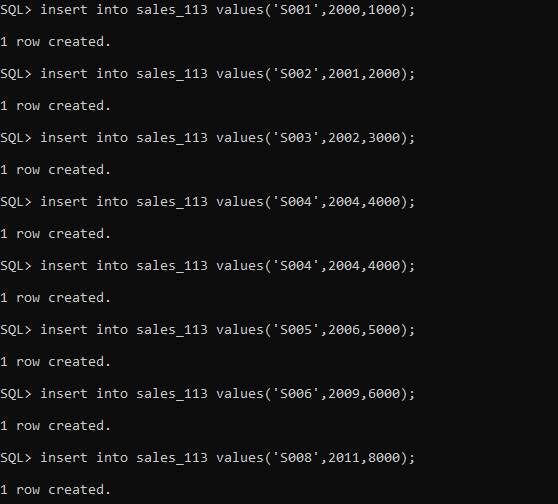
**Output:**

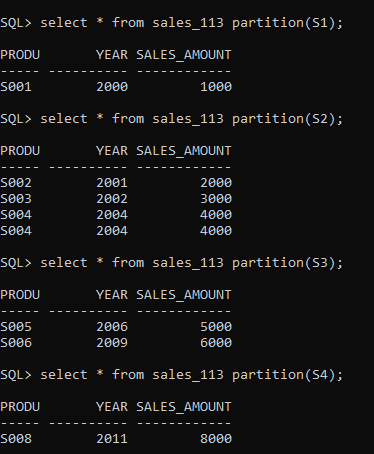


### Q2] Write a query for range partitioning on the table sales\_100 values for partitioning

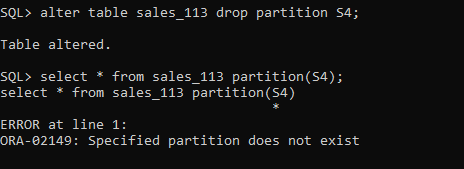
1. Up to 2001
2. Up to 2005
3. Up to 2010



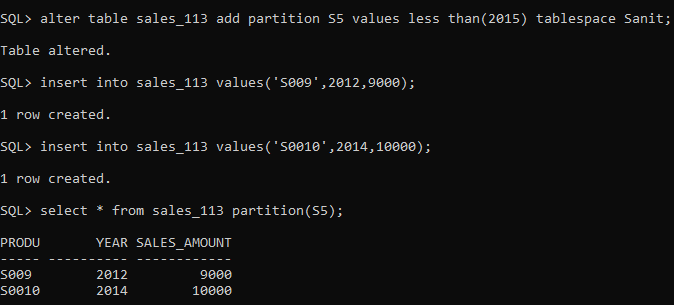




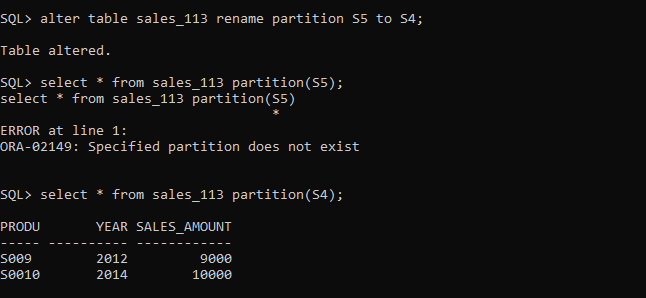
**Q3] Write a query to drop the partition S4.**



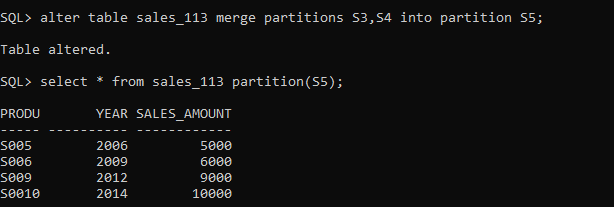
**Q4] Write a query to add new partition into sales\_100 table which has the values less than year(2015).**



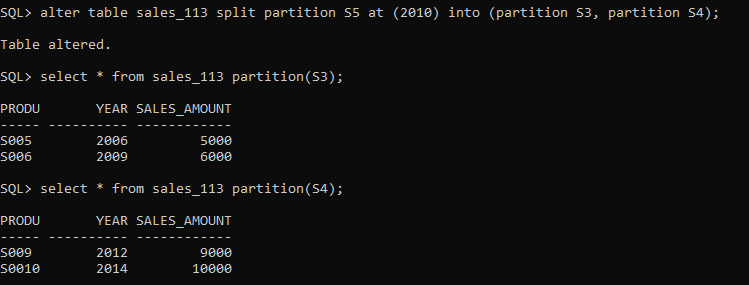
**Q5] Write a query to rename the partition S5 to S4.**



**Q6] Write a query to merge a partition S3 and S4 into partition S5.**



**Q7] Write a query split the partition S5 into partition S3 and S4.**



# Practical No: 1 (B)

* **List Partitioning**

**Aim: To perform LIST PARTITIONING**

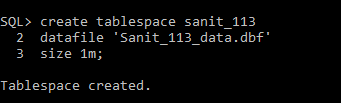
#### Department table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Columns: | dept\_number | dept\_name | dept\_budget | dept\_state |
|  | P001 | sales | 20000 | Goa |
|  | P002 | IT | 25000 | Delhi |
|  | P003 | HR | 10000 | Kerala |
|  | P004 | sales | 12000 | Haryana |
|  | P005 | sales | 15000 | Assam |
|  | P006 | Marketing | 50000 | Maharashtra |

Partition name: east(Assam), west(Goa, Maharashtra ), north(Delhi,Haryana) and south(Kerala)

1. Write a query for creating list partitioning on table Department on Columns name (states)
2. Write a query to display the data partition wise.
3. Write a query to add the new values on the columns state .
4. Write a query to merge partitions for north and east as a north-east .
5. Write a query to split partition north-east into north and east.

**Q1. Create tablespace**

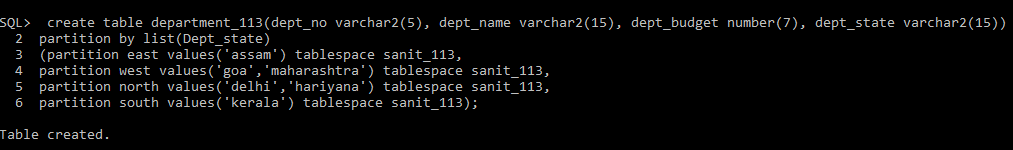


#### Create table with Partitions:

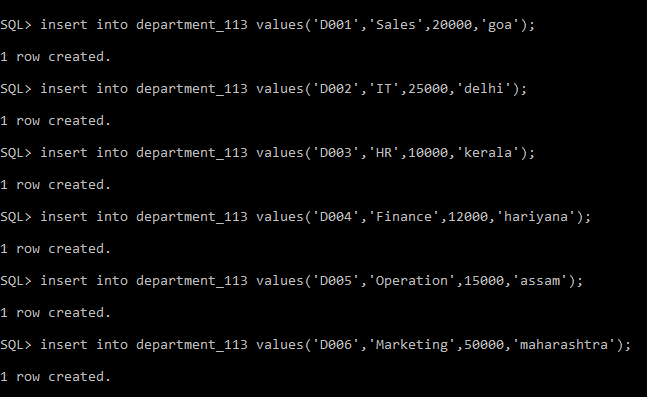
create table department\_113(dept\_no varchar2(5), dept\_name varchar2(15), dept\_budget number(7), dept\_state varchar2(15))

1. partition by list(Dept\_state)
2. (partition east values('assam') tablespace sanit\_113,
3. partition west values('goa','maharashtra') tablespace sanit\_113, 5 partition north values('delhi','hariyana') tablespace sanit\_113,

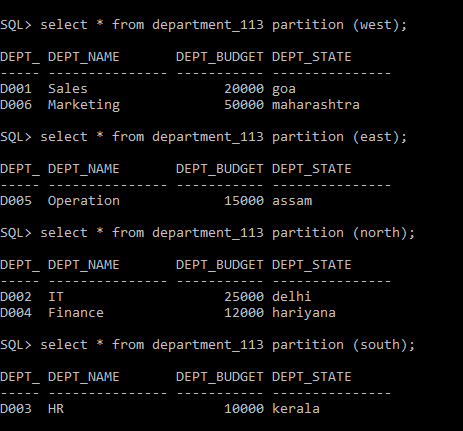
6 partition south values('kerala') tablespace sanit\_113);



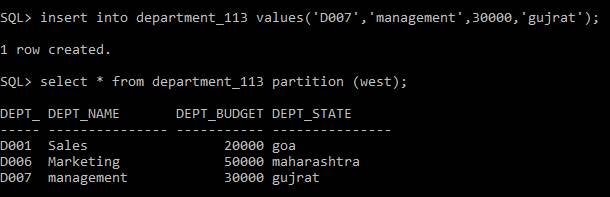
### Insert values:



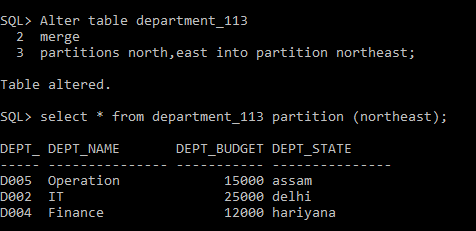
**Q2. Displaying the data partition wise**



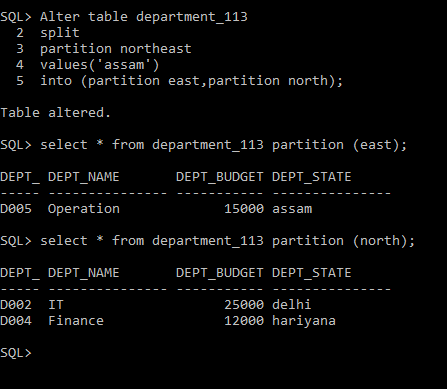
### Q3. Add the new values on the columns state (Gujrat)



**Q4. Merge partitions for north and east as a north-east .**



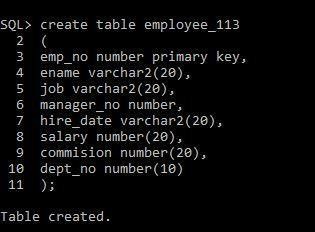
**Q5. Split partition north-east into north and east.**



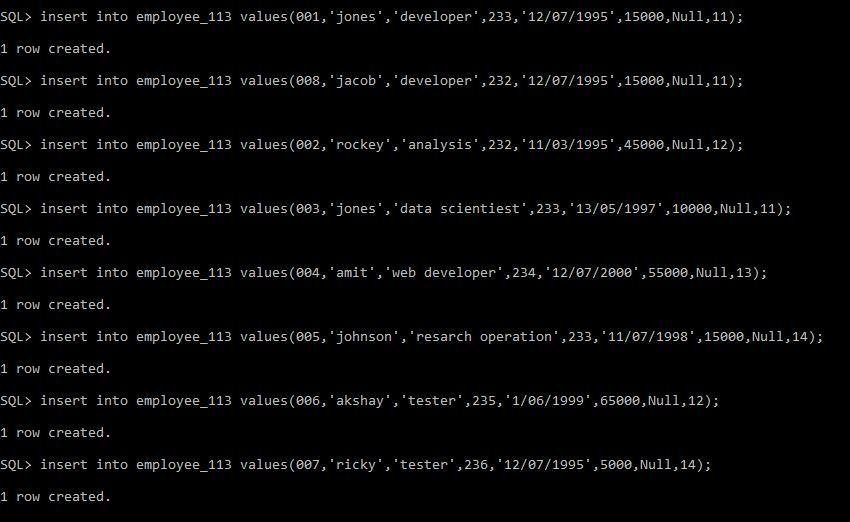
# Practical No: 02

## Implementation of Analytical queries

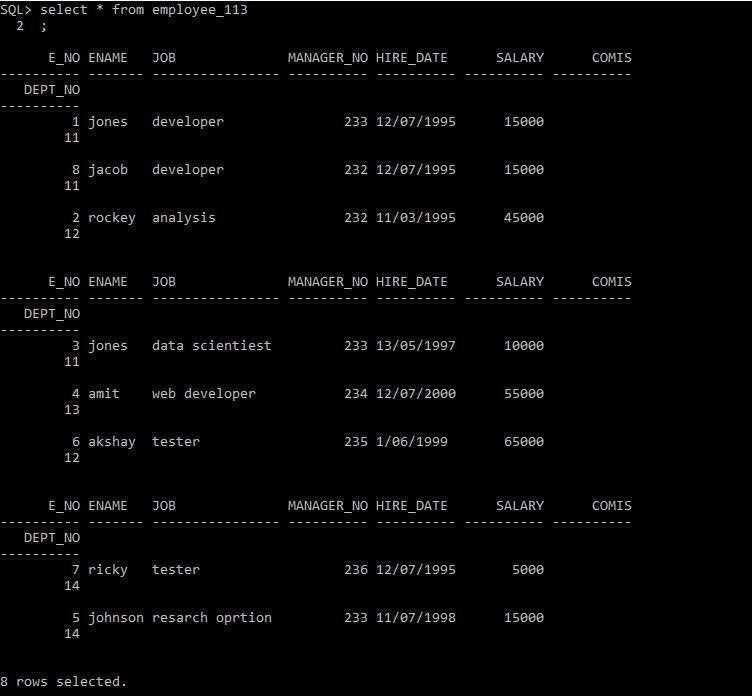
1. **create table employee\_113**



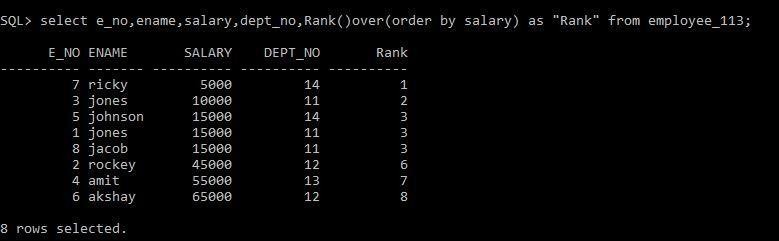
1. **Insert record in employee\_113**



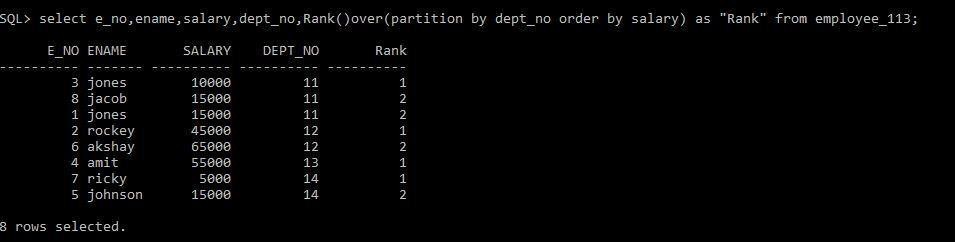
1. **write a query to display element in employee\_113**



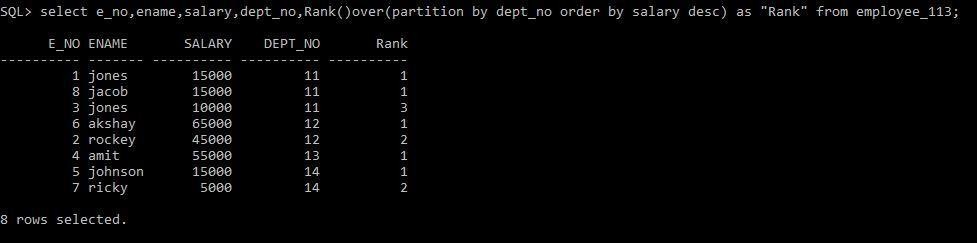
1. **write a to display employee\_6 using Rank order by salary**



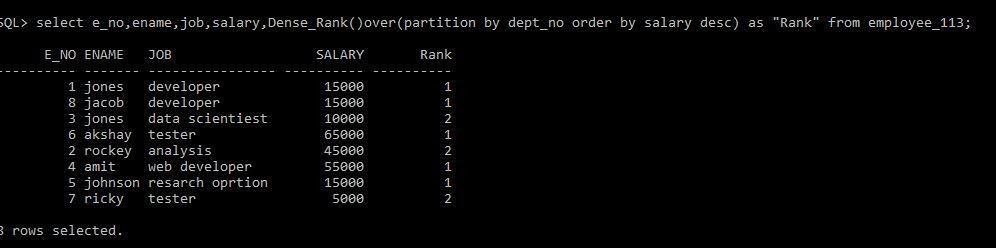
1. **write a to display employee\_6 using Rank order by salary and partition by dept\_no**



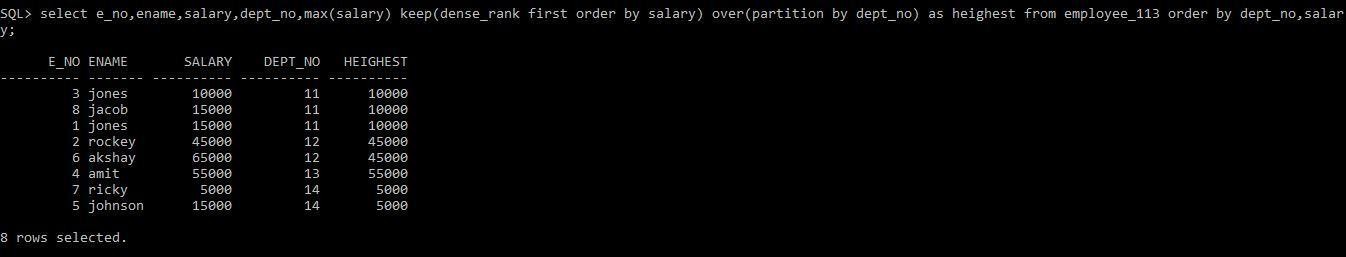
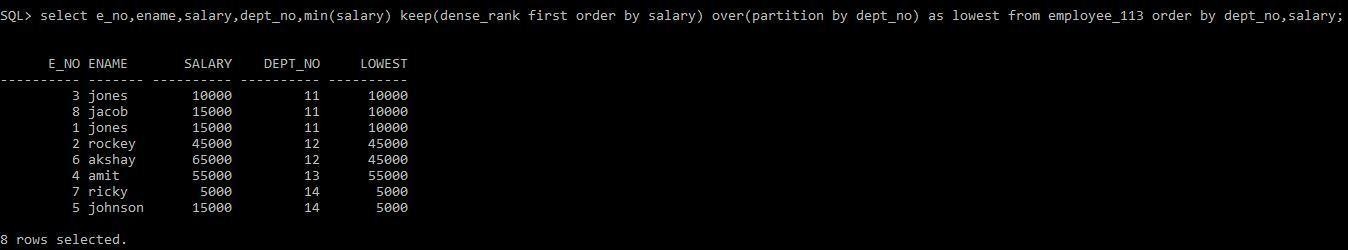
1. **write a to display employee\_6 using Rank order by salary desc and partition bydept\_no**



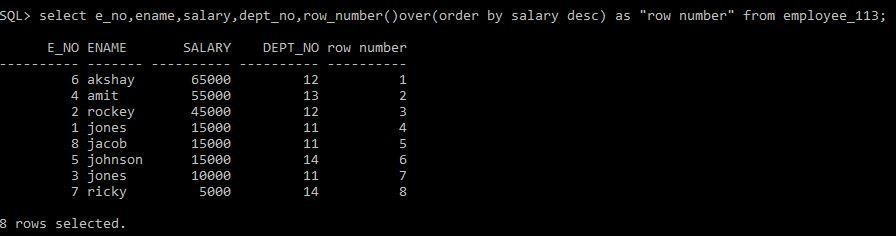
1. **rite a to display employee\_6 using Dense\_Rank order by salary desc and partitionby dept\_no**



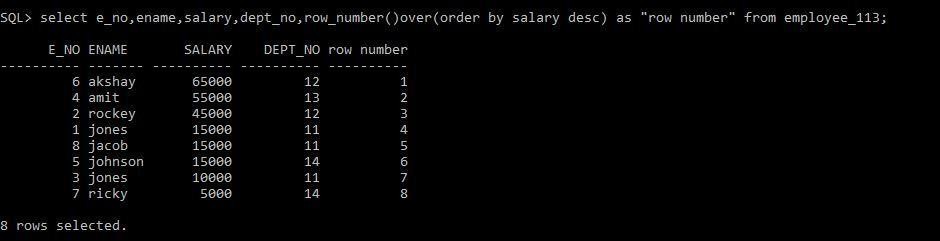
1. **Display the highest and the lowest salary of the employee within their departmentorder by dept\_no and salary. Min and max first value and last value**



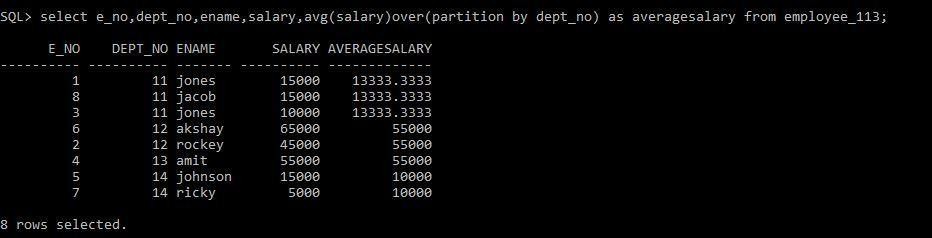
1. **Write a query to display a row number.**



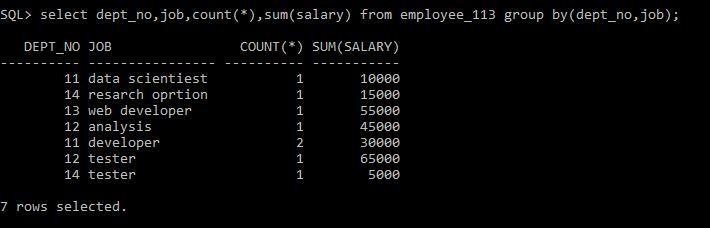
1. **Without Partition**



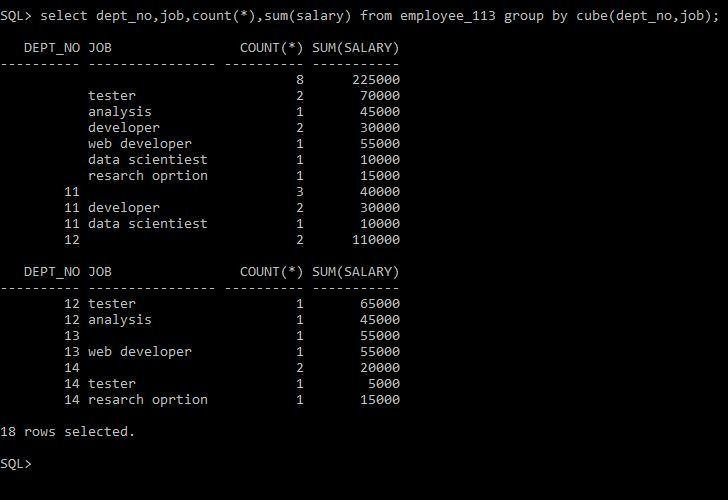
1. **Print the average salary of department.**



1. **Find out the sum of the salary, dept\_no and job wise**



1. **Group by function Cube.**

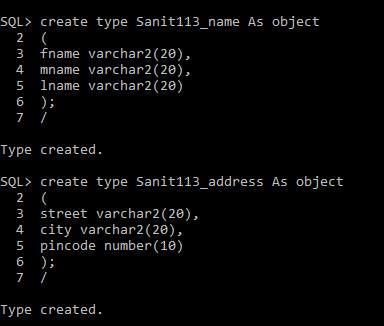


# Practical No: 03

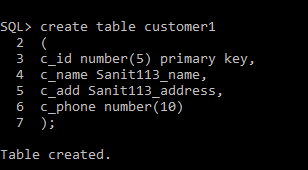
**AIM: Implementation of ORDBMS using ADT (Abstract Data Types), References.**

#### Abstract Data Types

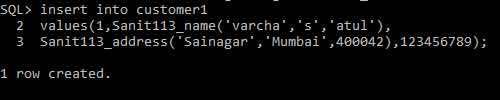
1. **Creating type:**



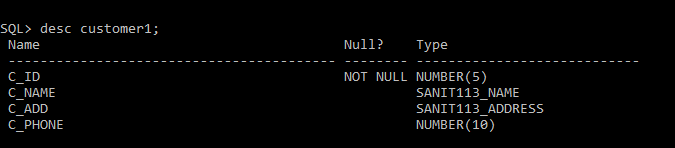
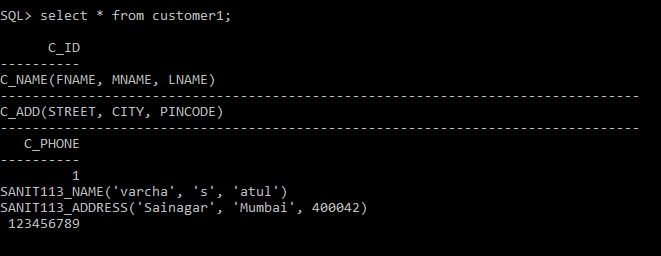
#### : Creating table



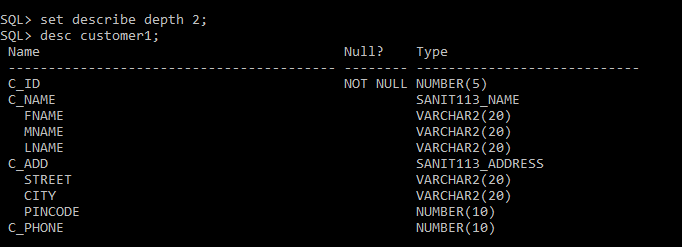
1. **Insert data:**



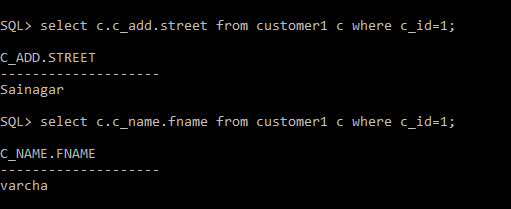
1. **Selecting data from table:**



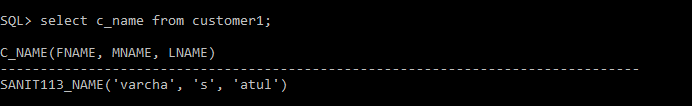
1. **Set Describe Depth 2;**



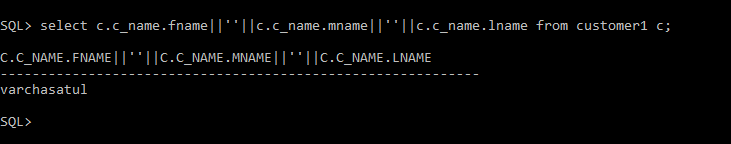
1. **Select c.c\_add.street from customer1 c where c\_id=1;**



### Select c\_name customer1;



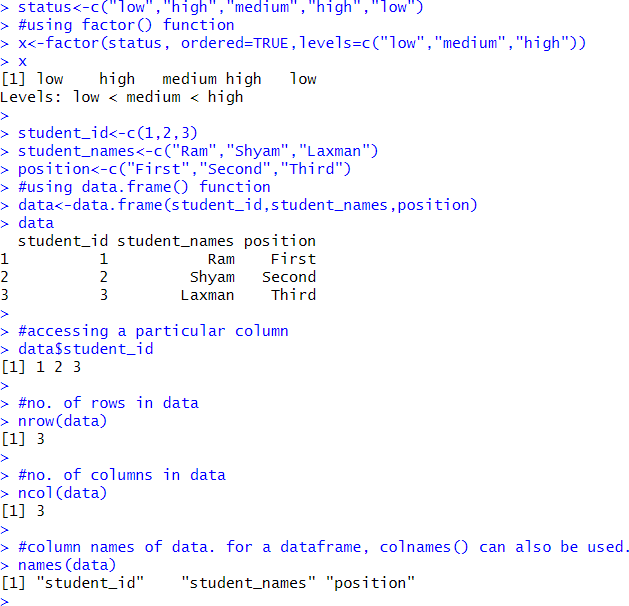
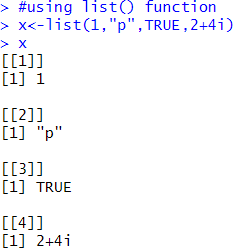
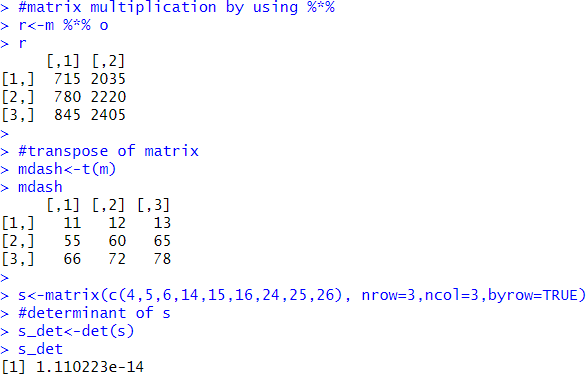
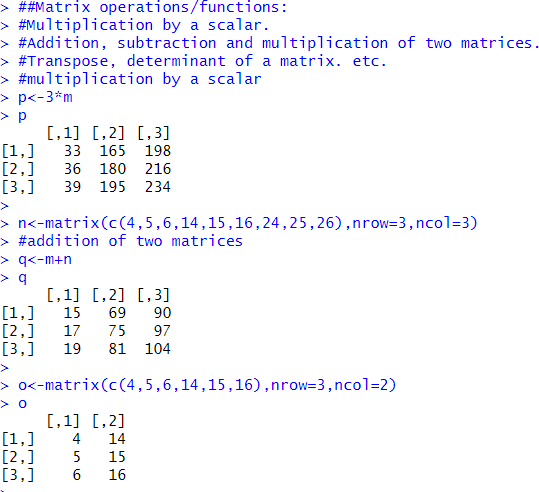
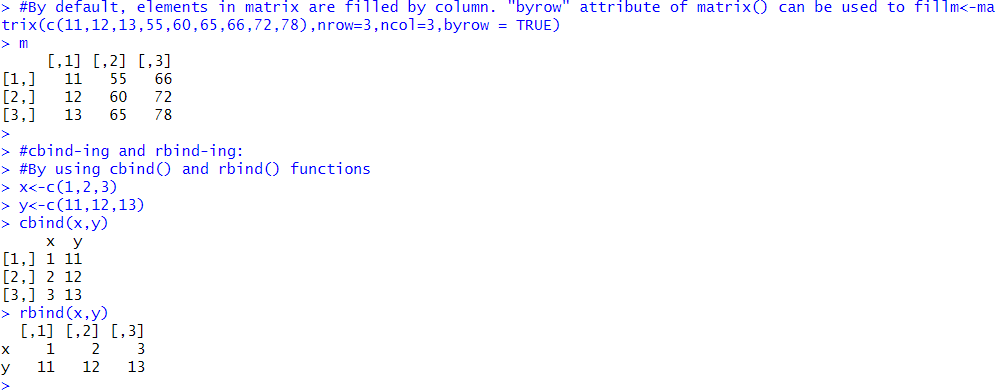
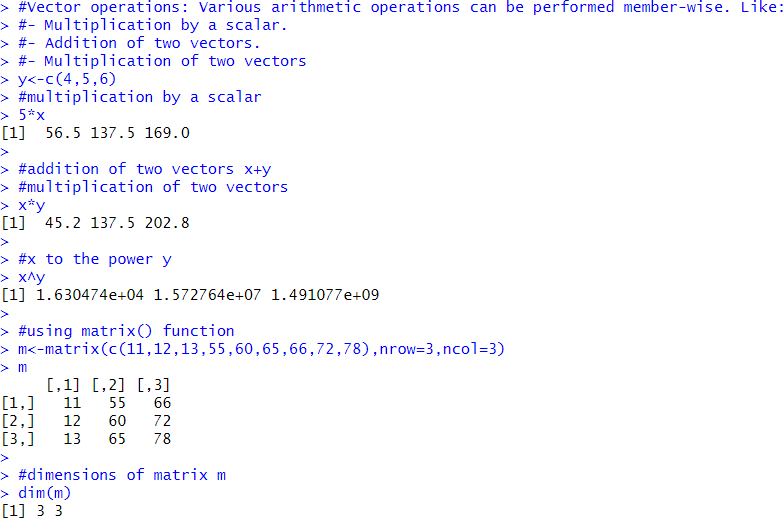
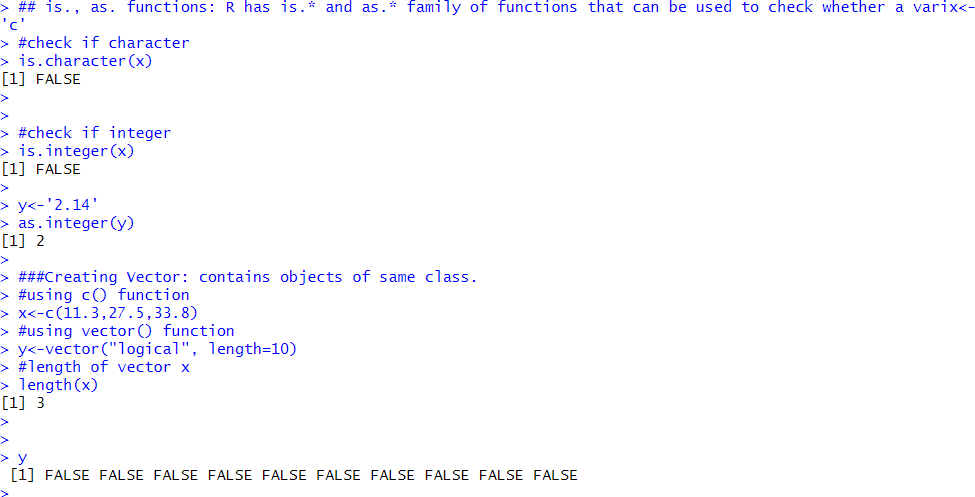
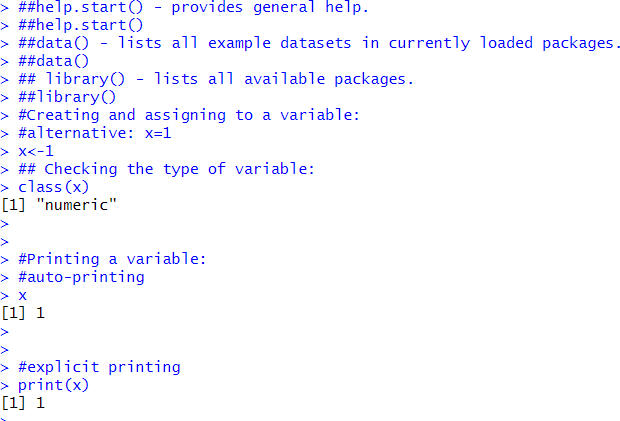
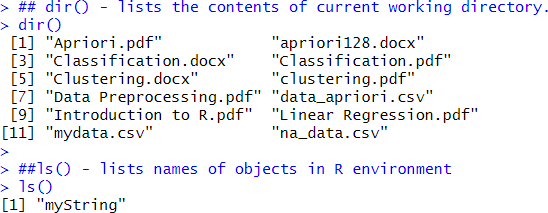
1. **Select c.c\_name.fname||’ ‘||c.c\_name.mname||’ ‘||c.c\_name.lname from customer1 c;**



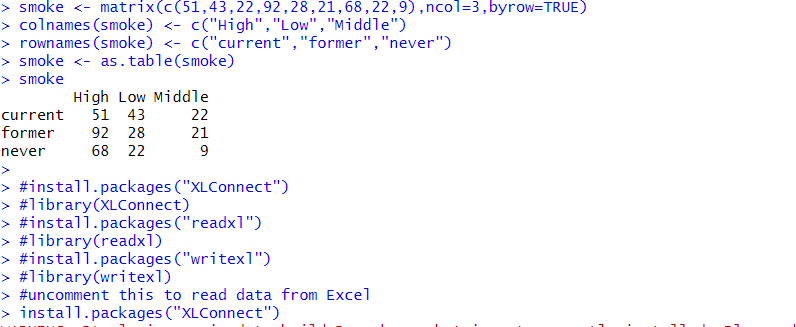
# Practical No: 05

**AIM: Introduction of R.**

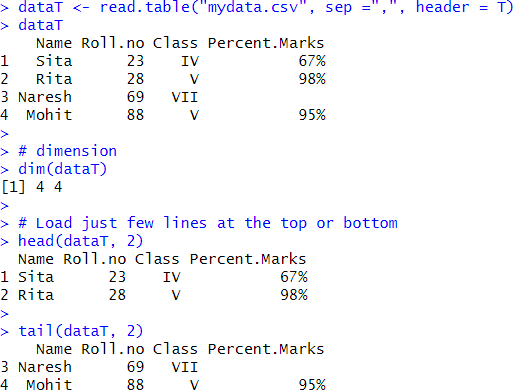
##### Basic R Commands.



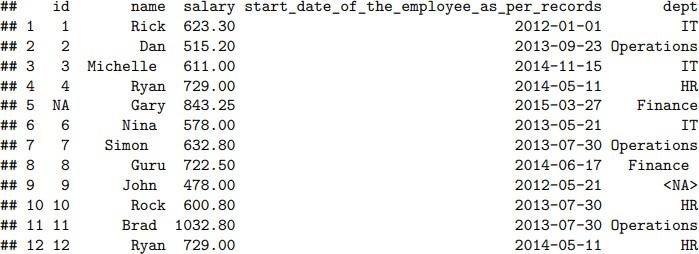
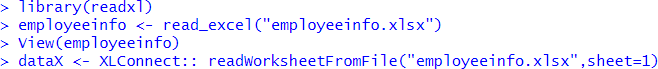
###Table command is used to create a 2dimensional table in R



##### Reading and writing data from csv



Reading and writing data from Excel using XLConnect



# Practical No: 06

**AIM: Data Preprocessing Techniques in R**

Data Preprocessing Techniques:

##### This reference Material is created for Mumbai university MCA Course for ADBMS. Topics Covered are Implementation of Data preprocessing techniques like,

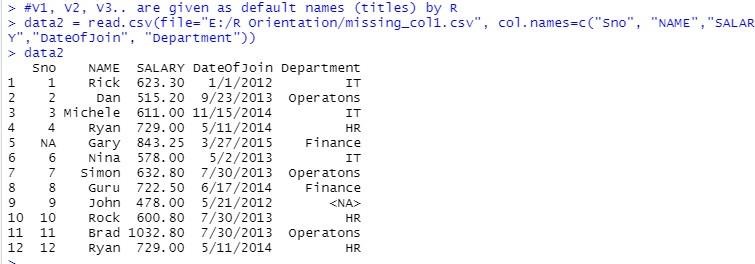
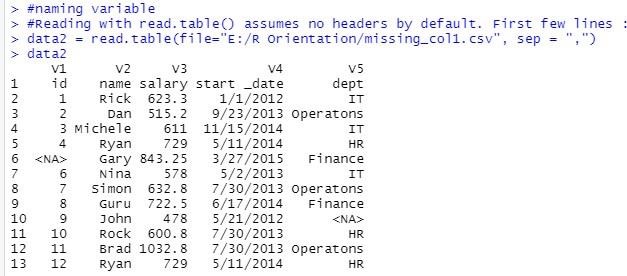
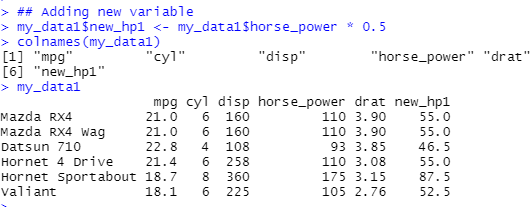
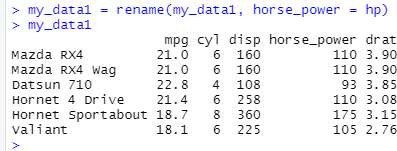
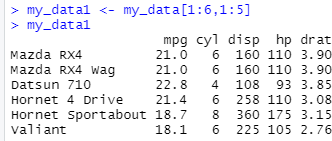
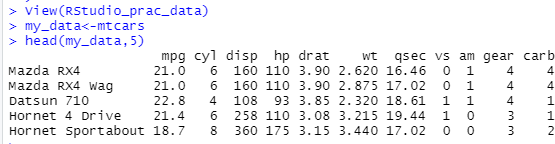
1. Naming and Renaming variables, adding a new variable.

##### Dealing with missing data.

1. Dealing with categorical data.

##### Data reduction using subsetting

setwd(“E:/Sanit Patil/R-Orientation”) getwd()

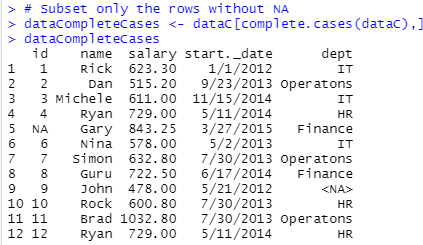
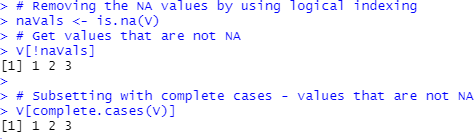
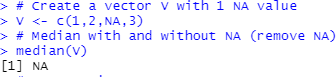


**Error Detection and Correction:**

NA : Not Available - Known as missing values Works as a place holder for something that is ‘missing’ Most basic operations(addition, subtraction, multiplication, etc.) in R deal with it without crashing and return NA if one of the inputs is NA

is.na(VALUE) is used to check if the input value is NA or not.

Returns a TRUE/FALSE vector Whereas in case of Excel like utilities for numeric computations it’s assumed to be 0

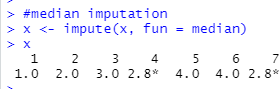
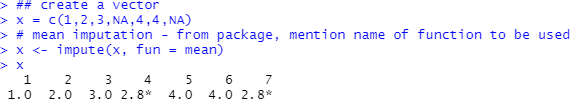
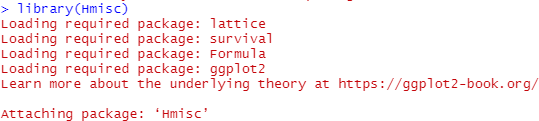


#### Imputation:

The process of estimating or deriving missing values. There are various methods for imputation

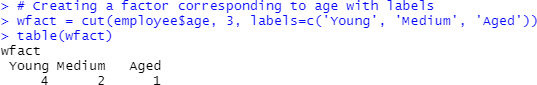
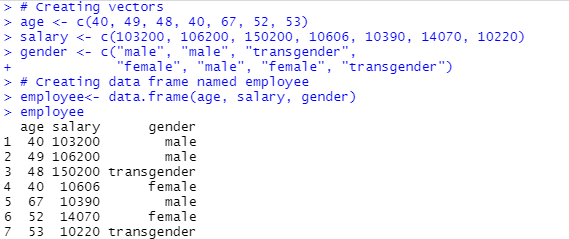
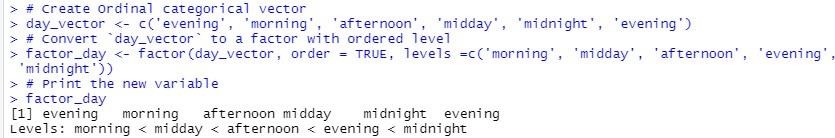
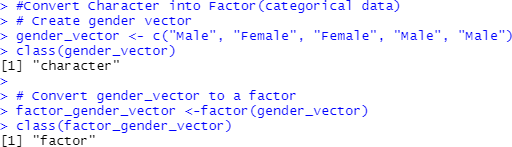
* Imputation of the mean – Imputation of the median
* Imputation using linear regression models

Package Hmisc implments many imputation methods, few examples :



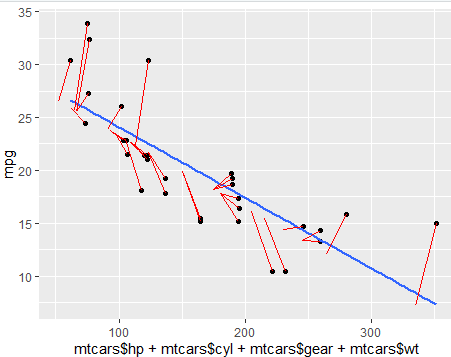
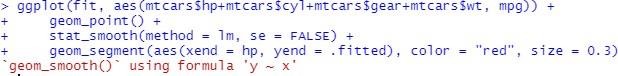
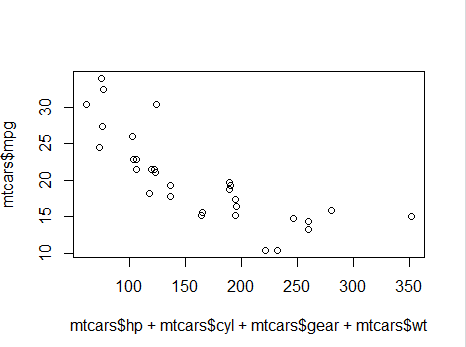
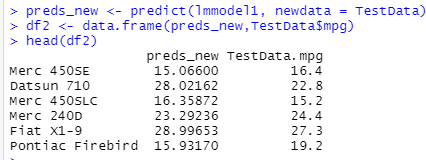
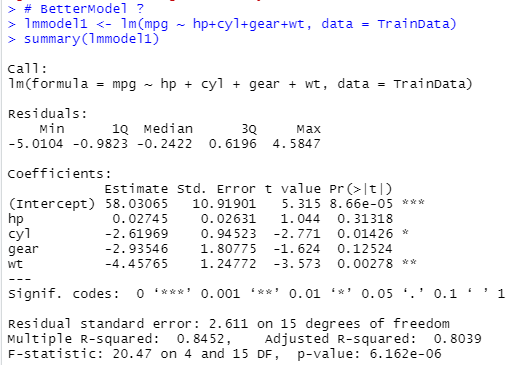
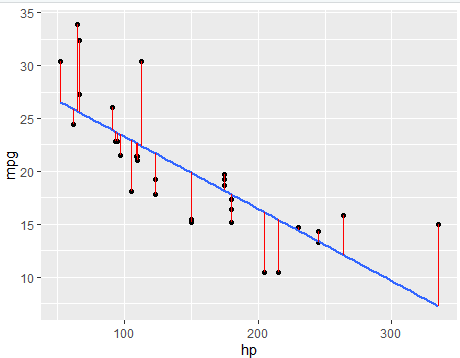
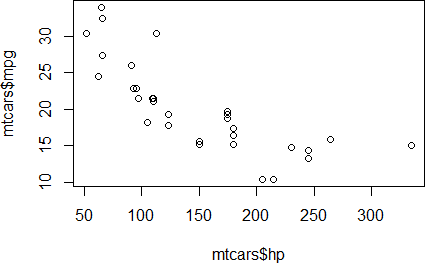
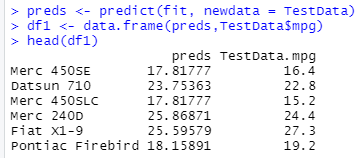
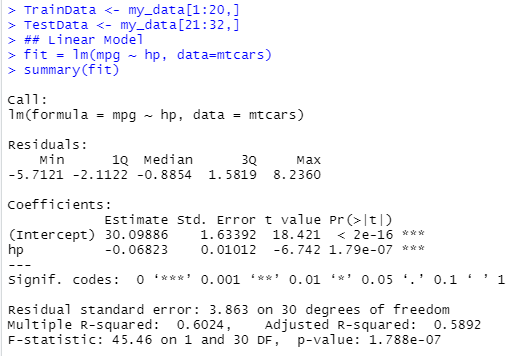
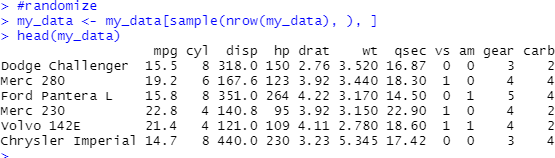
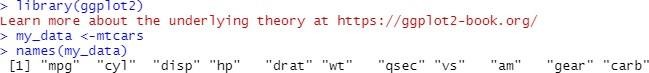
#### Categorical Data:

## Factors are variables in R which take on a limited number of different values; such variables are often referred to as categorical variables.



# Practical No: 07

**AIM: Implementation and Analysis of Linear Regression through graphical methods including plots.**



# Practical No: 08

## AIM: Implementation and Analysis of Classification Model.

Implementation and analysis of Classification algorithms like

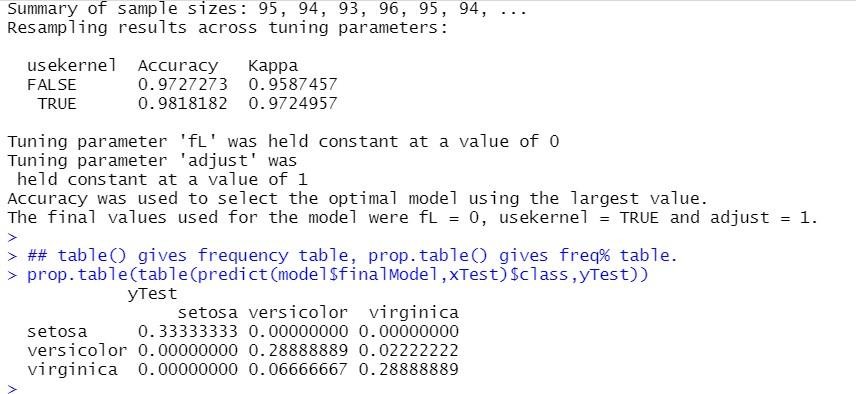
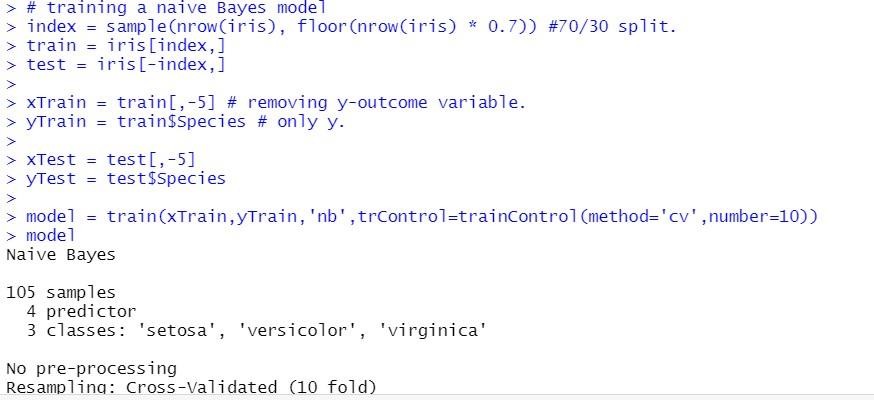
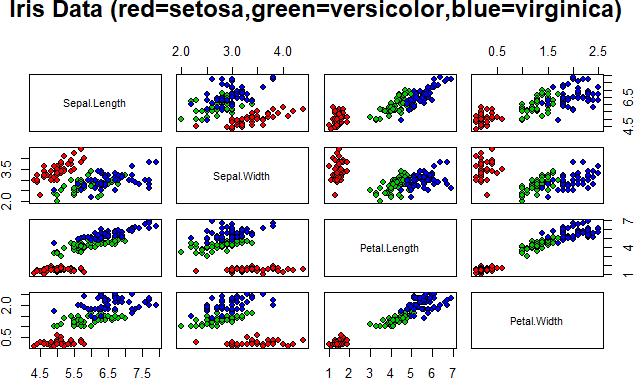
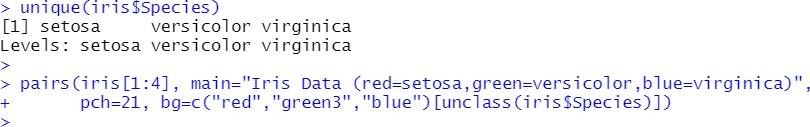
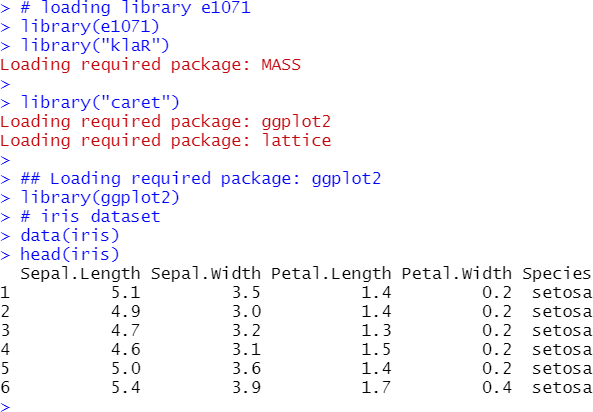
1. Naive Bayesian,
2. K-Nearest Neighbor
3. ID3 4. C4.5 Naive Bayes

* Based on the Bayes theorem
* Predicts based on probabilities from training data P(B|A) = P(A|B) P(B)/P(A)

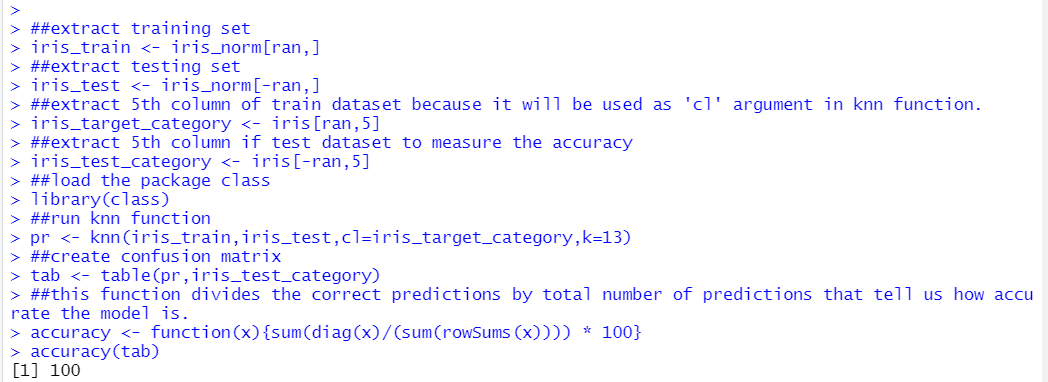
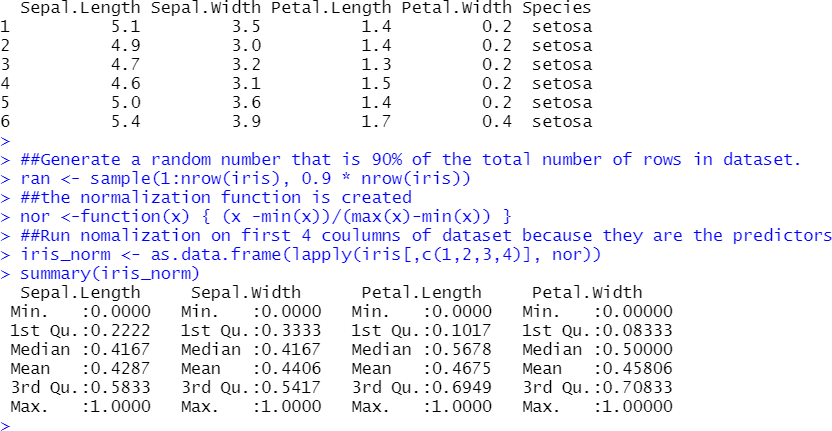
Gives posterior probability of ‘B’ given ‘A’ using prior probability of ‘B’ prior probability of ‘A’ and conditional probability of ‘A’ given ‘B’

* 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

##Naive Bayes setwd(“E:/Sanit Patil/R Orientation”)



**K-nearest Neighour**

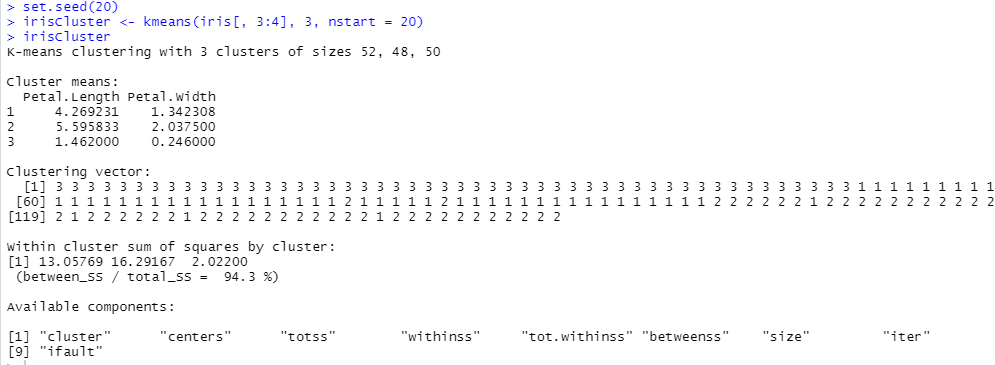
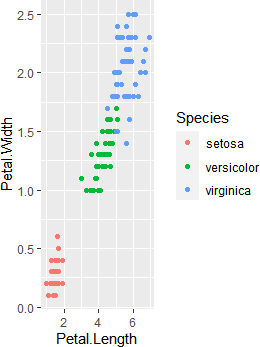
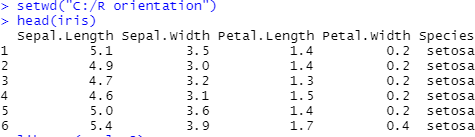


# Practical No: 09

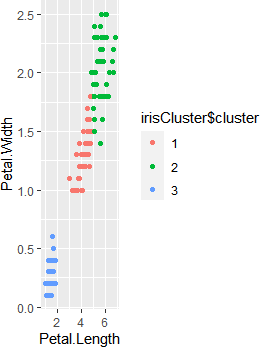
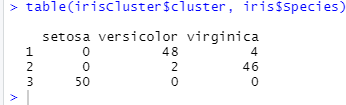
## AIM: Implementation of clustering algorithm.

1. K-Means
2. Agglomerative

\*\* K Means Clustering\*\*







# Practical No: 10

## AIM: Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

