# II - ASSIGNMENT

(Start Writing From Here)

1) List out different types of clustering techniques and explain K-Means Algorithm.

A) Clustering is a type of unsupervised learning wherein data points are grouped into different sets based on their degree of similarity.

Types of clustering techniques :-

1. Connectivity-based clustering (Hierarchical clustering)
   - (i) Agglomerative clustering
   - (ii) Divisive clustering
2. Centroids-based clustering (Partitioning clustering)
   - i) K-Means clustering
   - ii) Fuzzy C-Means clustering
3. Distribution based clustering
4. Density based clustering (Model based Methods)
5. Constraint-based (supervised clustering)
6. Unsupervised clustering
   - (i) K-Medoid clustering
7. Grid-based Methods

K-Means :-

⇒ It is a partitional clustering approach

⇒ Each cluster is associated with a centroid (center point)

⇒ Each point is assigned to the cluster with the closest centroid

⇒ Number of clusters, K, Must be specified.

K-Means Algorithm :-

In this each cluster's center is represented by the Mean value of the objects in the cluster.

Input :-

⇒ K : the number of clusters

⇒ D : a data set containing n objects.

output : A set of K clusters.

Algorithm :-

1) arbitrarily choose K objects from D as the initial cluster centers;

2) repeat

3) reassign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

4) update the cluster Means, that is, calculate the mean value of the objects for each cluster;

5) until no change;

Details of K-Means

1. Initial centroids are often chosen randomly (They vary from one to another)

2. The centroid is the Mean of the points in the cluster.

3. 'closeness' is measured by "Euclidean distance", cosine similarity, correlation etc...

4. K-Means will converge for common similarity measures mentioned above.

5. Most of the convergence happens in the first few iterations

6. Euclidean Distance

$$d(i,j) = \sqrt{|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \cdots + |x_{i_p} - x_{j_p}|^2}$$

Ex: Find the distance b/w 2 points, the original & the point (3,4)

$$d_E(O, A) = \sqrt{3^2 + 4^2} = 5$$

7. Update centroid:

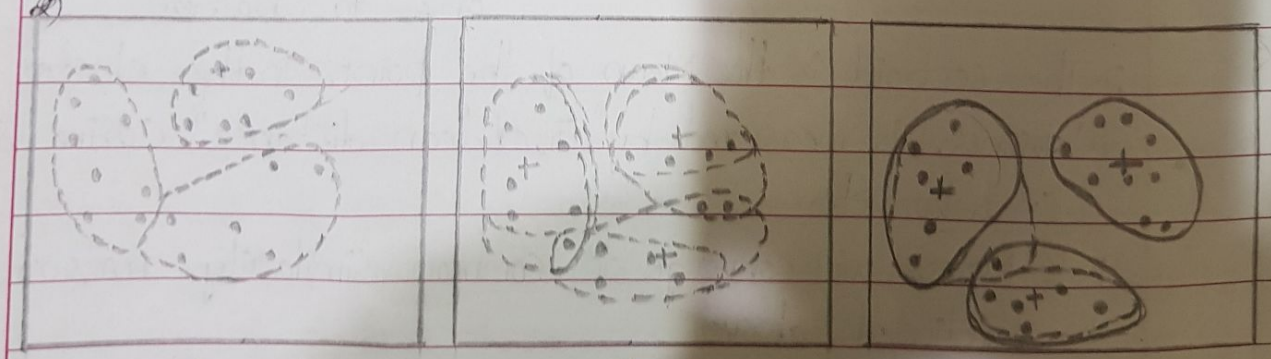
we use the following equation to calculate the n dimensional centroid point amid K n-dimensional points.

$$CP(x_1, x_2 \cdots x_K) = \left( \frac{\sum_{i=1}^{K} x1st_i}{K}, \frac{\sum_{i=1}^{K} x2nd_i}{K}, \ldots, \frac{\sum_{i=1}^{K} xnth_i}{K} \right)$$

Ex: Find the centroid of 3 2D points, (2,4), (5,2) & (8,9)

$$CP = \left( \frac{2+5+8}{3}, \frac{4+2+9}{3} \right) = (5, 5)$$

2)



(a) Initial clustering (b) Iterate (c) Final clustering

2) Discuss the following terms with example
a) Elements and variable b) Data categorization
c) Levels of Measurement

A) a) Elements :-

Elements are the entities, based on which the data is collected. Elements are very important feature of the data.

Variable :-

A characteristic of interest for the element. A variable is anything that can take on different values across your data set

Ex :-

| Earnings (in thousand Rs) | Teacher | Doctor | Painter |
|---|---|---|---|
| Weekly | 5 | 8 | 2 |
| Monthly | 150 | 100 | 80 |
| Annual | 1800 | 2000 | 800 |

In the above table,

⇒ Elements are : Teacher, Doctor, Painter

⇒ Variables are : Weekly, Monthly and Annual earning

b) Data Categorization :-

Categorization is the process of dividing the world into groups of entities whose members are in some way similar to each other.

So data could then be categorized as high sensitivity data, Medium sensitivity data and low sensitivity data. The difference is that these groups referred in the data categorization



CMRIT

don't need to be mutually exclusive, but in data classification they have to.

Data categorization :- 1 to Many relationship :-

✓ High risk

X Medium risk

✓ low risk

⇒ for this, we use cut() and discretize() and group_var() functions to categorize data.

c) levels of Measurement :-

Level of Measurement refers to how precisely a Variable has been Measured. It determines what type of analysis can be carried out. Within your dataset, you will have different Variables and these Variables can be recorded to varying degrees of precision. This is what's Known as the LOM.

There are 4 levels of Measurement :-

There is a hierarchy in the complexity and precision of LOM, from low (nominal) to high (ratio)

Going from lowest to highest, the 4 levels of M are cumulative. This Means that they each take on the properties of lower levels and add new properties.

1) Nominal :- The data can only be categorized. you can categorize data by labelling them in mutually exclusive groups, but there is no order between categories.

Ex :- City of birth, gender, car brands, Marital status.

2) Ordinal :- The data can be categorized and ranked but you cannot say anything about the intervals b/w



CMRIT

the rankings.

Ex: Top 5 olympic Medallists, Likert-type questions).

3) Interval: The data can be categorized, ranked and evenly spaced between neighboring data points, but there is no true zero point.

Ex: Test scores, Personality inventories, Temperature in celsius or Fahrenheit

4) Ratio: The data can be categorized, ranked, evenly spaced and has a natural zero (true zero) point.

A true zero means there is an absence of the variable of interest. In ratio scales, zero does mean an absolute lack of the variable.

Ex: Height, Age, weight, Temperature in Kelvin

The Four levels of Measurement:

| | Nominal | ordinal | Interval | Ratio |
|---|---|---|---|---|
| Categorizes and labels variables | ✓ | ✓ | ✓ | ✓ |
| Ranks categories in order | | ✓ | ✓ | ✓ |
| Has known, equal intervals | | | ✓ | ✓ |
| Has a true zero | | | | ✓ |

3) Discuss about Decision Making and loop statements in R programming with example.

A) Decision Making Statements :-

Decision making structures require the programmer to specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

if statement :-

```
syntax:- if (boolean_expression){
            // Statement(s) will execute if the boolean exp is true
         }
```

If statement consists of a boolean exp followed by one or more statements. If the BE evaluates to be true, then the block of code inside the if stmt will be executed. If it evaluates to false, then the first set of code after the end of if stmt (after brace) will be executed.

```
Ex:-  X <- 30L
      if (is.integer(x)){
          Print ("X is an integer")
      }
```

⇒ When it is compiled, result will be "X is an integer"

if-else statement :-

```
Syntax:- if (boolean_exp){
            // stmt(S) will execute if BE is true
         }else {
```



CMRIT

```
//stmt(s) will execute if B.E is false
}
```

⇒ if the BE evaluates to be true, then the if block of code will be executed, otherwise else block of code will be executed.

Ex:-
```
X <- c("what","is","truth")
if ("Truth" %in% x){
    Print("Truth is found")
}else{
    Print("Truth is not found")
}
```

⇒ When this is compiled, result will be "Truth is not found"

if...else if...else statement:-

An if stmt can be followed by an optional else if... else stmt, which is very useful to test various conditions using single if...else if stmt.

⇒ An if can have zero or one else & it must come after any else if's.

⇒ An if can have zero or Many else if's and they must come before the else.

⇒ once an else if succeeds, none of the remaining else if's or else's will be tested.

Syntax:-
```
if(boolean_exp1){
        //Executes when the BE1 is true
}else if(boolean_exp 2){
        //Executes when the BE 2 is true
}else if (boolean_exp3){
        //Executes when the BE3 is true
}else{
```

```
        //executes when none of the above condition is true.
    }
```

Ex:-

```
x <- c("what","is","truth")
if("Truth"%in%x){
    Print("Truth is found in 1st time")
}else if("truth"%in%x){
    Print("truth is found the 2nd time")
}else{
    Print("No truth found")
}
```

⇒ When this is compiled, result will be "truth is found the 2nd time"

switch statement:-

A switch stmt allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case

Syntax:- switch(expression, case1, case2, case3 ...)

Rules:-

⇒ if the value of exp is not a character string it is coerced to integer.

⇒ you can have any number of case stmts within a switch. Each case is followed by the value to be compared to and a colon.

⇒ If the value of the integer is b/w 1 and nargs()-1 (Max no. of args) then the corresponding element of case condition is evaluated & the result returned.

⇒ If exp evaluates to a character string then that string is matched to the names of the elements.

⇒ If there is more than one Match, the first Matching element is



CMRIT

returned.

⇒ No Default argument is available.

⇒ In the case of noMatch, if there is a unnamed element of ... its value is returned. (if there is more than one such argument an error is returned).

Ex:-
```
X <- switch(
    3,
    "first",
    "second",
    "third",
    "fourth",
  )
print(x)
```

⇒ When this is compiled result will be "third"

Loop statements:-

A loop stmt allows us to execute a stmt or group of stmts Multiple times and the following is the general form of a loop stmt in most of the programming languages-

R programming language provides the following kinds of loop to handle looping requirements.

Repeat loop:-

Executes a sequence of stmts multiple times and abbreviates the code that manages the loop variable. until a stop condition is met.

syntax:-
```
repeat {
    commands
    if (condition) {
        break
    }
}
```

Ex:-
```
V <- c("Hello", "loop")
cnt <- 2
repeat {
    Print(V)
```



CMRIT

```
cnt <- cnt+1
if(cnt > 5){
    break
}
```

output: [1] "Hello" "Loop"
[1] "Hello" "Loop"
[1] "Hello" "loop"
[1] "Hello" "Loop"

While loop:

While loop executes the same code again and again until a stop condition is met. When the condition is tested and the result is false, the loop body will be skipped and the first stmt after the while loop will be executed.

Syntax:
```
While(test_exp){
    stmt
}
```

Ex:
```
v <- c("Hello", "while Loop")
cnt <- 2
while (cnt < 7){
    Print(v)
    cnt = cnt +1
}
```

output: "Hello" "While loop"
"Hello" "While loop"
"Hello" "While loop"
"Hello" "while loop"
"Hello" "while loop"

For loop: It allows you to efficiently write a loop that needs to execute a specific number of times. They are not limited to integers, or even numbers in the input. we can pass vectors, logical vectors, lists or expressions

Syntax:
```
for(value in vector){
    stmts
}
```

Ex:
```
v <- LETTERS[1:4]
for(i in v){
    Print(i)
}
```

output: "A"
"B"
"C"
"D"

4) Discuss about chi-square test in R with example

A) The hypothesis testing in which there is no information of population and the data will be in nominal or order form period. In such condition we will use non-parametric hypothesis test or distribution free test.

In the non-parametric or distribution free test there is no information about population parameter. The hypothesis testing is applied to ordinal or nominal data.

$$\chi^2 = \sum \frac{(F_o - F_e)^2}{F_e}$$

$F_o$ = observed frequency, $F_e$ = expected frequencies

⇒ It is a very prominent test of non-parametric test

Ex:- A company has both day and night shift employees. To know how many of the employees wanted to shift to nightshift

conditions for applying $\chi^2$-test:

⇒ $\chi^2$ test is based on $\chi^2$ distribution

⇒ In the $\chi^2$ distribution there is only one parameter:

The degree of freedom of the $\chi^2$ distribution

1) Total frequency should be very large

$\sum F_o = N > 50$ where N = total frequency

2) Total of observed frequency and total of expected frequency should be equal $\sum F_o = \sum F_e$

3) No expected frequencies should not be less than 5

⇒ $\chi^2$ distribution is a continous distribution.

⇒ If any E.F is less than 5 then the continuity of the



CMRIT

$\chi^2$ distribution will be broken and the result getting from the test will not be that effective or reliable and so, no expected frequency should be less than 5.

→ If any expected frequency is less than 5 then add either before or after frequency to this E.F so that, the E.F becomes larger than 5 and in this case the dof will be less than 1

Whenever we do pulling (Making expected frequency greater than 5), dof becomes less than 1.

4) no information about parent population.

Applications of $\chi^2$ test :-

1) $\chi^2$ test of independence of attributes or characteristics it tests whether two different attributes or two different characteristics.

2) $\chi^2$ test of goodness of fit in case of the goodness of fit also we are going to apply the same $\chi^2$ test formula.

3) $\chi^2$ test of homogenity. Here also we use the same $\chi^2$ test formula.

4) $\chi^2$ test of inference about population Variance.

Ex:- Table of observed Values

| Qualification/ Marital Status | Middle School | High School | Bachelor's | Master's | PhD | Total |
|---|---|---|---|---|---|---|
| Never Married | 18 | 36 | 21 | 9 | 6 | 90 |
| Married | 12 | 36 | 45 | 36 | 21 | 150 |
| Divorced | 6 | 9 | 9 | 3 | 3 | 30 |
| Widowed | 3 | 9 | 9 | 6 | 3 | 30 |
| Total | 39 | 90 | 84 | 54 | 33 | 300 |

Table of Expected Values

| Qualification/ M.S | Middle Scl | High Scl | Bachelor's | Master's | PhD |
|---|---|---|---|---|---|
| Never Married | $\frac{90\times39}{300}=11.7$ | $\frac{90\times90}{300}=27$ | 25.2 | 16.2 | 9.9 |
| Married | 19.5 | 45 | 42 | 27 | 16.5 |
| Divorced | 3.9 | 9 | 8.4 | 5.4 | 3.3 |
| Widowed | 3.9 | 9 | 8.4 | 5.4 | 3.3 |

Total Calculation of $\chi^2$

| observed Values (O) | Expected Values (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|
| 18 | 11.7 | 6.3 | 39.69 | 3.39 |
| 36 | 27 | 9 | 81 | 3 |
| 21 | 25.2 | -4.2 | 17.64 | 0.7 |
| 9 | 16.2 | -7.2 | 51.84 | 3.2 |
| 6 | 9.9 | -3.9 | 15.21 | 1.53 |
| 12 | 19.5 | -7.5 | 56.25 | 2.88 |
| 36 | 45 | -9 | 81 | 1.8 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 3 | 3.3 | -0.3 | 0.09 | 0.02 |

$\chi^2_{calculated} = 23.57$

$\Rightarrow$ DOF = (columns -1) (rows -1) = (5-1)(4-1) = 4x3 = 12

$\chi^2_{tab} = 21.03$ from table with probability 0.05 significance level ($\alpha$) = 0.05

$\therefore \chi^2_{cal} > \chi^2_{tab}$ ($\chi^2_{critical}$)

$\therefore$ We reject null hypothesis & accept alternate hypothesis

$\therefore$ There is significant relation b/w the Marital status & educational qualification



CMRIT

5) Define Regression? Explain linear Regression in R with ex?

A) Regression :-

A Regression is a statistical technique that relates a dependent Variable to one or more independent Variables.

Linear Regression :-

Linear Regression is used to predict the value of an outcome variable Y on the basis of one or more input predictor Variables X. It is used to establish a linear relationship between the predictor and response Variables.

In LR, predictor and response Variables are related through an equation in which the exponent of both these Variables, is 1. Mathematically, a linear relationship denotes a straight line, when plotted as a graph.

$$Y = aX + b$$

Here, ⇒ Y is a response variable

⇒ X is a predictor Variable

⇒ a and b are constants that are called the coefficients.

Steps for establishing the Regression :-

The prediction of the weight of a person when his height is known, is an example of regression. To predict the weight, we need to have a relationship between the height and weight of a person.

Steps to create relationship

1. In the 1st step, we carry out the experiment of gathering a sample of observed values of height & Weight.
2. After that, we create a relationship model using the lm()



CMRIT

function of R.

3) Next, we will find the coefficient with the help of the model and create the Mathematical equation using this coefficient.

4) We will get the summary of the relationship model to understand the average error in prediction, known as residuals.

5) At last, we use the predict() function to predict the weight of the new person.

Syntax for lm() function : lm(formula, data)

Formula : It is a symbol that presents the relationship between X & Y.

Data : It is a vector on which we will apply the formula.

Creating Relationship Model and Getting the coefficients :

let's start performing the second & third steps. we will use the lm() function and pass the X and Y input vectors and store the result in a variable named relationship_model.

Ex :
```
# Creating input vector for lm() function
X <- c(141, 134, 178, 156, 108, 116, 119, 143, 162, 130)
Y <- c(62, 85, 56, 21, 47, 17, 76, 92, 62, 58)
# Applying the lm() function
relationship_model <- lm(y~x)
# Printing the Coefficient
Print(relationship_model)
```

output : lm(formula = y ~ x)

coefficients :

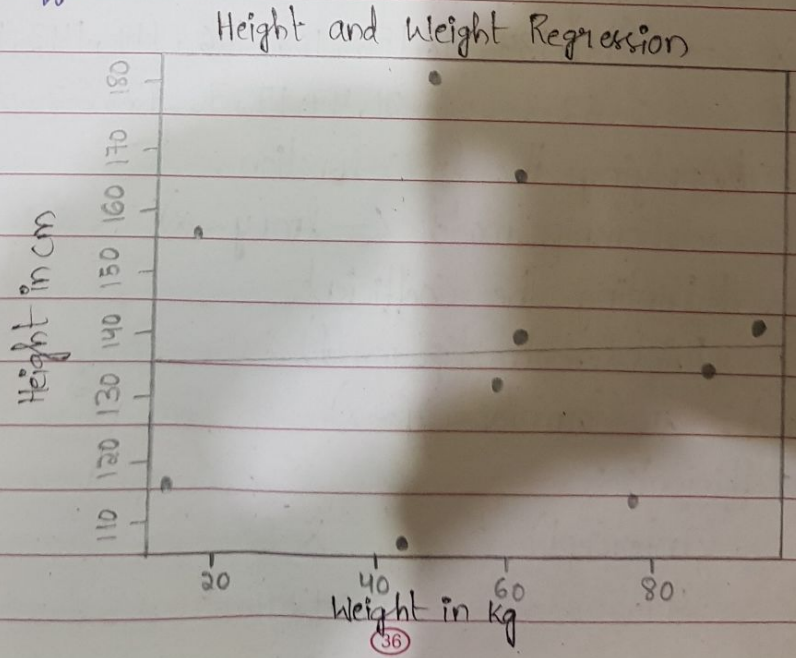
| (Intercept) | X |
|---|---|
| 47.50833 | 0.07276 |

Plotting Regression:-

Now, we plot our prediction results with the help of plot() function. This function takes parameter X and Y as an input vector and many more arguments.

Ex:-
```
X <- c(141,134,178,156,108,116,119,143,162,130)
Y <- c(62,85,56,21,47,17,76,92,62,58)
relationship_model <- lm(y~x)
# Giving a name to the chart file
Png (file = "linear_Regression.png")
# Plotting the chart
Plot (Y,X,col ="red", main = "Height and Weight Regression",
        abline(lm(x~y)), cex = 1.3, pch =16, xlab =
    "Weight in Kg", ylab ="Height in cm")
# Saving the file
dev.off()
```

output:-






CMRIT

# II - ASSIGNMENT

(Start Writing From Here)

1) Discuss K-medoids clustering with an example?

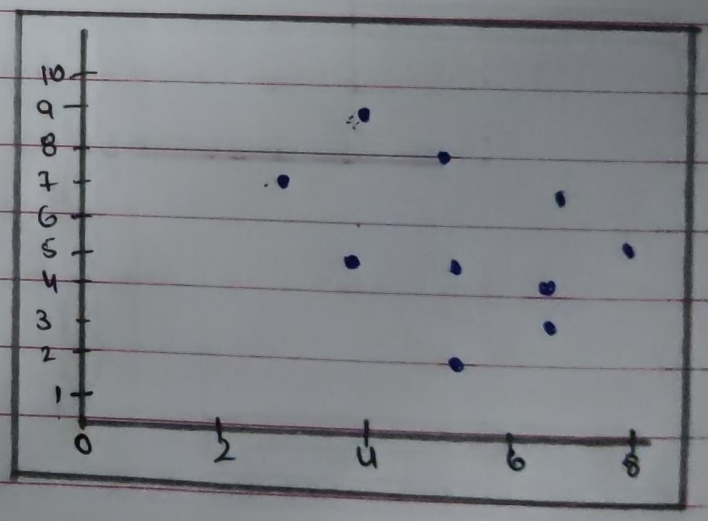
K-medoids (also called partitioning Around medoid) algorithm was proposed ~~by~~ in 1987 by Kaufman and Rousseeuw. A medoid can be defined as a point in the cluster, whose dissimilarities with all the other points in the cluster are minimum. The dissimilarity of the method (ci) and object (pi) is calculated by using $E = |pi - ci|$

Example:- If a graph is drawn using the data points,

Data set

| S.No. | X | Y. |
|---|---|---|
| 0 | 8 | 7 |
| 1 | 3 | 7 |
| 2 | 4 | 9 |
| 3 | 9 | 6 |
| 4 | 8 | 5 |
| 5 | 5 | 8 |
| 6 | 7 | 3 |
| 7 | 8 | 4 |
| 8 | 7 | 5 |
| 9 | 4 | 5 |

Graph :-




Step 1:- Let the randomly selected 2 medoids, so select K = 2, and let C1 - (4,5) and C2 - (8,5) are the two ~~meth~~ medoids.



CMRIT

Step 2:- calculating cost. The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

| S.NO | X | Y | Dissimilarity from $C_1$ | Dissimilarity from $C_2$ |
|---|---|---|---|---|
| 0 | 8 | 7 | 6 | 2 |
| 1 | 3 | 7 | 3 | 7 |
| 2 | 4 | 9 | 4 | 8 |
| 3 | 9 | 6 | 6 | 2 |
| 4 | 8 | 5 | - | - |
| 5 | 5 | 8 | 4 | 6 |
| 6 | 7 | 3 | 5 | 3 |
| 7 | 8 | 4 | 5 | 1 |
| 8 | 7 | 5 | 3 | 1 |
| 9 | 4 | 5 | - | - |

Here, we have used manhattan distance formula to calculate the distance matrices between medoid and non-medoid points

The formula tell that Distance $= |X_1 - X_2| + |Y_1 - Y_2|$

Each point is assigned to the cluster of that medoid whose dissimilarity is less. points 1, 2, and 5 go to cluster $C_1$ and 0, 3, 6, 7, 8 go to cluster $C_2$.

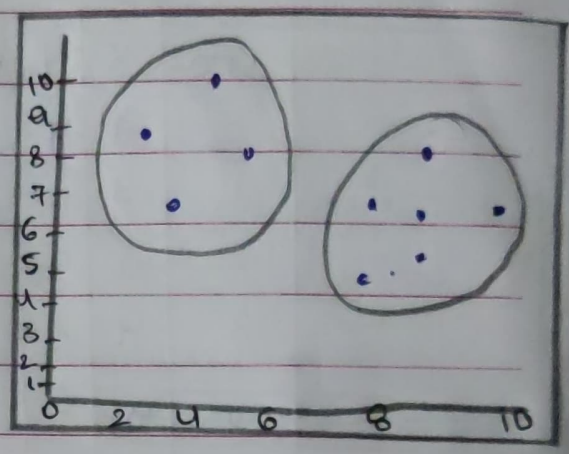
The cost $= (3+4+4) + (3+1+1+2+2) = 20$.

Step 3:- randomly select one non-medoid point and recalculate the cost. Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids $C_1$ (4, 5) and $C_2$ (8, 4) is calculated and tabulated.



CMRIT

| S.NO | X | Y | Dissimilarity from C1 | Dissimilarity from C2 |
|---|---|---|---|---|
| 0 | 8 | 7 | 6 | 3 |
| 1 | 3 | 7 | 3 | 8 |
| 2 | 4 | 9 | 4 | 9 |
| 3 | 9 | 6 | 6 | 3 |
| 4 | 8 | 5 | 4 | 1 |
| 5 | 5 | 8 | 4 | 7 |
| 6 | 7 | 3 | 5 | 2 |
| 7 | 8 | 4 | - | - |
| 8 | 7 | 5 | 3 | 2 |
| 9 | 4 | 5 | - | - |




Each point is assigned to that cluster whose dissimilarity is less. So, points 1, 2, and 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The New cost = (3+4+4) + (2+2+1+3+3) = 22

Swap cost = New cost − previous cost = 22 − 20 = (2 > 0).

→ As the swap cost is not less than zero, we undo the swap. (4, 5) and (8, 5) are the final medoids. The clustering would be in the above graph.

∴ $\boxed{\text{The time complexity is } O(k * (n-k)^2)}$



CMRIT

2) Explain about Data Management and Indexing in R programming.

Data Management:-

Indexing is a technique to optimize our performance or processing speed of querying records in the database by minimizing the number of searches or scans required. It is a data structure technique that is used to quickly locate and access the data in a database. Users cannot see the indexes, they are just used to speed searches and queries.

→ It is nothing but like our book indexes, it does not totally scan our entire rows, it just searches for an index range and finds the result within the index.

Indexing in R programming:-

(1) clustering index:-

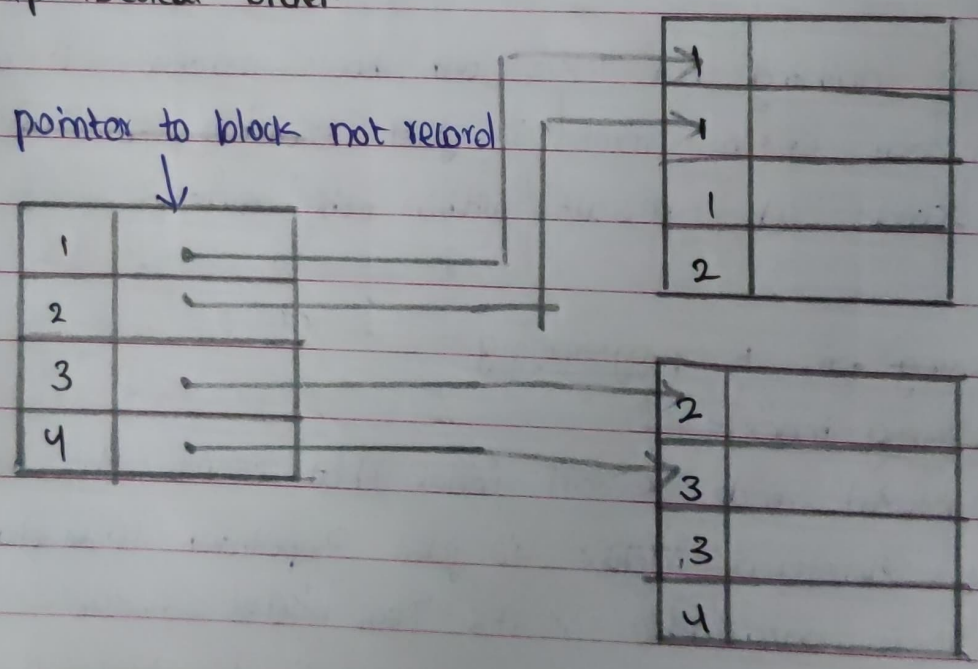
A clustered index is an index that stores the actual data & non clustered index is just a pointer to a data. This data is available in its leaf nodes. A table can only have one clustered index and up to 249 non-clustered indexes. If a table does not have a clustered index it is refered to as a heap

Clustered index is created only when both the following condition satisfy:-

(i) The data (or) file, that you are moving into Secondary memory should be in sequential (or) sorted order.

(ii) There should be a key value, meaning it can not have repeated values.

Here, the we can create the one primary key is also called (or) like as dictionary where the data is arranged in alphabetical order.




we can have only one clustered index in one table, but we can have one clustered index on multiple columns, and that type of index is called composite index.

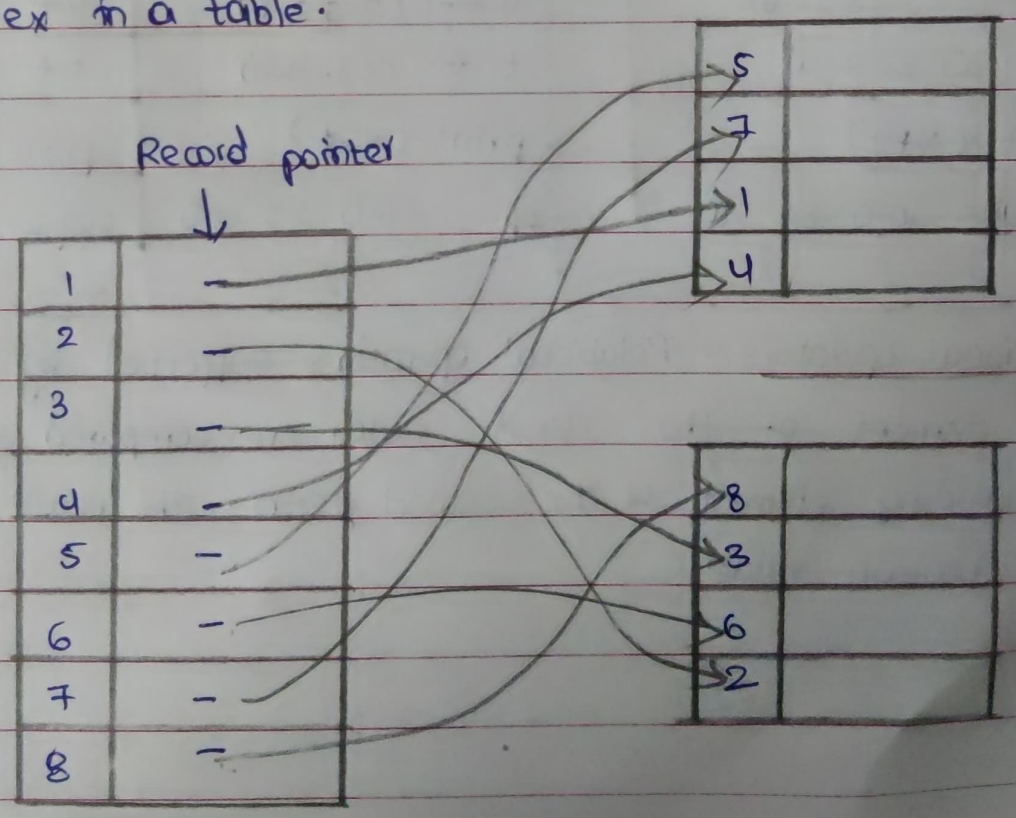


CMRIT

2) Non-clustered index:-

Non-clustered index is similar to the index of a book. The index of a book consists of a chapter name and page number, if you want to read any topic (or) chapter then you can directly go to that page by using index of that book. No need to go through each and every page of a book.

The data is stored in one place, and index is stored in another place. Since, the data and non-clustered index is stored separately, then you can have multiple non-clustered index in a table.






CMRIT

3) Discuss about Different types of operators in R programming with Example.

Operators:-

An operator is a symbol that tells the compiler to perform specific mathematical (or) logical manipulations. R language is rich in built-in operators and provides.

Types of operators:-

we have the following types of operators in 'R' programming

1. Arithmetic operators:- Arithmetic operators supported by R language. The operators act on each element of the vector.

| "+" Adds two vectors | "−" subtracts two vectors | "*" multiplies both vectors |
|---|---|---|
| v ← c(2, 5.5, 6) | v ← c(2, 5.5, 6) | v ← c(4, 5.2, 7) |
| t ← c(8, 3, 4) | t ← c(8, 3, 4) | t ← c(6, 4, 3) |
| print (v+t) | print (v−t) | print (v*t) |
| outputs:- 10.0 8.5 10. | op:- −6.0 2.5 2.0 | output:- 24.0 20.8 21 |

2) Relational operators:- Relational operators supported by R language. Each element of the first vector is compared with the corresponding element of the second vector. The result of comparision is a boolean value.

| operator | Symbol | Example. |
|---|---|---|
| < | ~~greater than~~ less than | ex:- v ← c(2, 5.5, 6, 9) \| t ← c(8, 2.5, 14, 9)<br>print (v<t)<br>output:- [1] TRUE FALSE TRUE FALSE |
| > | greater than | ex:- v ← (2, 5.5, 6, 9) \| t ← c(8, 2.5, 14, 9)<br>print (v>t)<br>output:- [1] FALSE TRUE FALSE FALSE. |
| <= | less than are equal to | ex:- v ← (2, 6, 9, 12) \| t ← c(8, 5, 9, 14)<br>print (v<=t)<br>output:- [1] TRUE FALSE TRUE TRUE. |
| >= | greater than or equal to | ex:- v ← c(2, 5.5, 6, 9) \| t ← c(8, 2.5, 14, 9)<br>print (v>=t)<br>output:- [1] FALSE TRUE FALSE TRUE |
| == | equal to | Ex:- v ← c(2, 5.5, 6, 9) \| t ← c(8, 2.5, 14, 9)<br>print (v==t)<br>output:- FALSE FALSE FALSE TRUE. |

3) Logical operator:- logical operators supported by R language. It is applicable only to vectors of type logical, numeric (or) complex, All numbers greater than 1 are considered as logical value TRUE.



CMRIT

| operator | symbol | Example: |
|---|---|---|
| ~~logical~~ logical And | && | v ← c(3,1, TRUE, 2+3i)<br>t ← c(4,1, FALSE, 2+3i) print (v&&t)<br>output :- [1] TRUE TRUE FALSE TRUE |
| logical OR | \| | v ← c(3,0, TRUE, 2+2i)<br>t ← c(4,0, FALSE, 2+3i) print (v\|t)<br>output :- [1] TRUE FALSE TRUE TRUE. |
| logical not | ! | v ← c(3,0, TRUE, 2+2i)<br>print (!v)<br>Output: [1] FALSE TRUE FALSE FALSE |
| logical and | && and \|\| | v ← c(3,0,TRUE, 2+2i)<br>t ← c(1,3, TRUE, 2+3i)<br>print (v && t) \| print (v \|\| t)<br>Output: [1] TRUE \| output: FALSE. |

4) <u>Assignment operators</u>:- These operators are used to assign values to vectors.

<u>operator</u>:

(i) <-

<<- called left Assignment.

$V_1$ ← c(3, 1, TRUE, 2+3i)

$V_2$ ← c(3, 1, TRUE, 2+3i)

$V_3$ = c(3, 1, TRUE, 2+3i)



CMRIT

print (v1)

print (v2)

print (v3)

output: [1] 3+0.5 1+0.5 1+0.5 2+3i

[1] 3+0.5 1+0.5 1+0.5 2+3i

[1] 3+0.5 1+0.5 1+0.5 2+3i

(ii) ->> called Right Assignment.

c(3, 1, TRUE, 2+3i) → v1

c(3, 1, TRUE, 2+3i) →> v2

print (v1)

print (v2)

output:-

[1] 3+0.5 1+0.5 1+0.5 2+3i

[1] 3+0.5 1+0.5 1+0.5 2+3i

(5) miscellaneous operators:-

These operators are used to for specific purpose and general mathematical (or) logical computation.

operator:- : colon operators it creates the series of numbers in sequence for a vector.

v ← 2:8

print (v)

Output: [1] 2 3 4 5 6 7 8.



CMRIT

(ii) %. In %. – This operator is used to Identify if an belongs to a vector

Eg: $v_1 \leftarrow 8$

$v_2 \leftarrow 12$

$t \leftarrow 1:10$

print ($v_1$ %. in %.t)

print ($v_2$ %. in %.t)

Output:

[1] TRUE

[1] FALSE.

4) Discuss about T-test in R with Example.

In statistics, the T-test is one of the most common test which is used to determine whether the mean of the two groups is equal to each other. The assumption for the test is that both groups are sampled from a normal distribution with Equal fluctuation. The null hypothesis is that the two means are the same, and the alternative is that they are not identical. It is Known that under the null hypothesis, we can compute a t-statistic that will follow a t-distribution with $n1+n2-2$ degrees of freedom.

Example:- (T- tests perform in R programming by 3 types):-

i) one-sample Test:-

one-sample T-test which compares the mean of a vector against a theoretical Mean. There is a following formula which is used to compute the T-test:-

$$t = \frac{m-\mu}{\frac{s}{\sqrt{n}}}$$

eg:- set.seed(0)

ship_vol ← (rnorm(70, mean = 35000, sd = 2000))

t.test(ship_vol, mu = 35000)

output:- data : ship_vol

t = -0.27702, df = 69, p-value = 0.7826

alternative hypothesis :- true mean is not equal to 35000

95 percent confidence interval :-

34498.18 35379.45

sample estimates :

mean of x

34938.81

2) paired - sample Test :- To perform a paired - sample test, we need two vectors data y1 and y2. Then, we will run the code using the syntax. test [y1, y2, paired = TRUE]

example :-

```
set.speed (2800)
pre.treatment ← c(rnorm (2000, mean = 130, sd = 5))
post. treatment ← c(rnorm (2000, mean = 144, sd = 4))
t.test (pre_Treatment, post_Treatment, paired = TRUE)
```

paired t-test.

Output :- data : pre_treatment and post_Treatment

t = 20.789, df = 1999, p-value < 2.2e-16

altneranative hypothesis: true difference in means is not equal to 0.

95 percent confidence interval:

5.593910 6.759259

sample estimates :

mean of the differences

6.176585



CMRIT

3) Independent - sample Test :-

Depending on the structure of our data, and the equality of their variance, the independent - sample T-test can take one of the three forms.

Example: set.speed (0)

spenders.cleve ← rnorm (50, mean = 300, sd = 70)

spenders.NY ← rnorm (50, mean = 350, sd = 70)

t.test (spenders.cleve, spenders.NY, var.equal = TRUE)

Two sample t-test.

output: data : spenders.cleve and spenders.NY

t = -4.0115, df = 98, p-value = 0.0001179

alternative hypothesis :- true difference in means is not equal to 0.

95 percent confidence interval:

-74.47029 -25.17621

sample estimates:

mean of x mean of y

801.6752 351.4984

5) Define regression? Explain logistic regression in R with Example.

**Regression:-**

Regression analysis is a powerful statistical method that allows you to examine the relationship between two or more variables of interest. while there are many types of regression analysis, at their core they all examine the influence of one or more independent variables on a dependent variable.

**Logistic Regression:-**

Logistic Regression is another widely used regression analysis teachnique and predicts the value with a range. More over, it is used for predicting the values for predicting the values for categorical data. For example, Email is either spam or non-spam, winner or loser, male or female, etc.

Mathematically,

$$\boxed{y = \frac{1}{1 + e^{-x}}}$$

**Implementation in R:-**

In R programming, glm() function is used to create a logistic regression model.

Syntax:- glm (formula, data, family)



CMRIT

Example:-

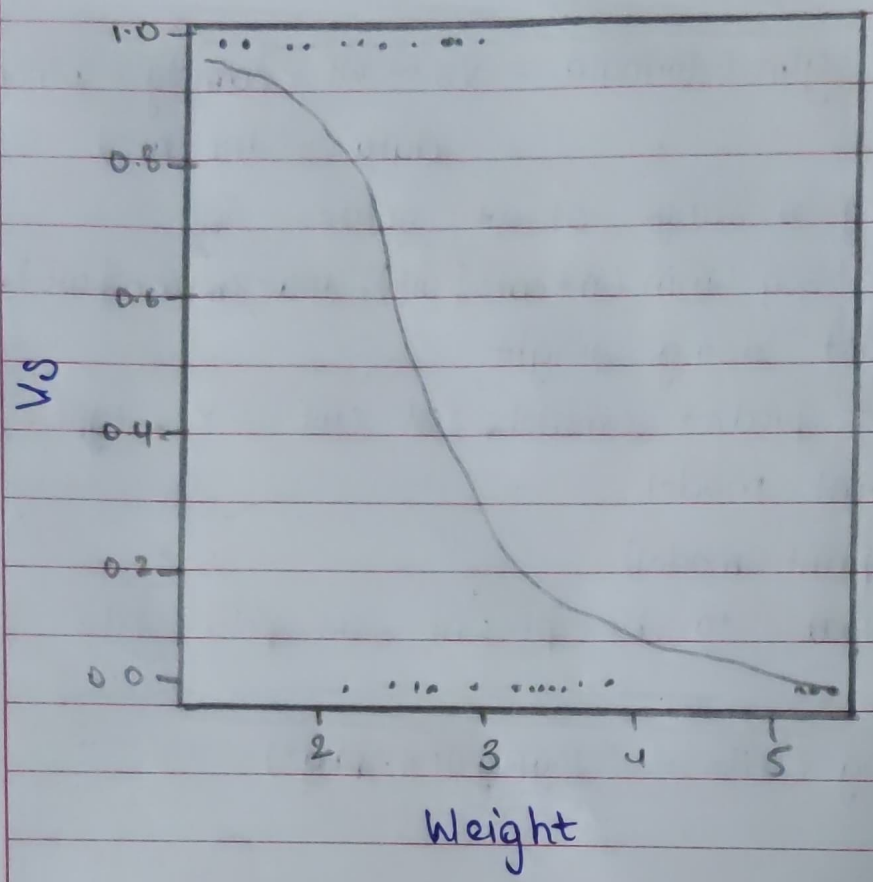
```
model <- glm (formula = vs ~ wt, family = binomial,
                                data = mtcars)
# creating a range of wt values
  x <- seq (min (mtcars$wt), max (mtcars$wt), 0.01)
# predict using weight
  y <- predict (model, list (wt = x), type = "response")
# print model
    print (model)
# output to be present as PNG file

    png (file = "LogReg GFG.png")

    # plot
    plot (mtcars$wt, mtcars$vs, pch = 16,
                  xlab = "weight", ylab = "vs")
      lines (x, y)

    dev.off ().
```

Output:- call: glm (formula = vs ~ wt, family = binomial,
data = mtcars)

coefficients:

| (Intercept) | wt |
|---|---|
| 5.715 | -1.911 |

Null Deviance : 43.86 | Residual Deviance : 31.37






CMRIT

# II - ASSIGNMENT

(Start Writing From Here)

1) What is cluster analysis ? Explain Hierarchal methods?

A) cluster analysis is a process to find similar groups of objects in order to form clusters. It is an unsupervised machine learning-based algorithm that acts on unlabelled data. A group of data points would comprise together to form a cluster in which all the objects would belong to the same group.

Hirarechal clustering:-

Hirarechal clustering refers to an unsupervised learning procedure that determine successive clusters based on previously defined clusters. It works via grouping data into a tree of clusters. Hirarechal clustering starts by treating each data points as an individual cluster. The endpoint refers to a different set of clusters, where each cluster is different from the other cluster, and the objects within each other are the same as one another.

There are two types of Hirarchical clustering:-

- Agglomerative Hirarchical clustering
- Divisive clustering

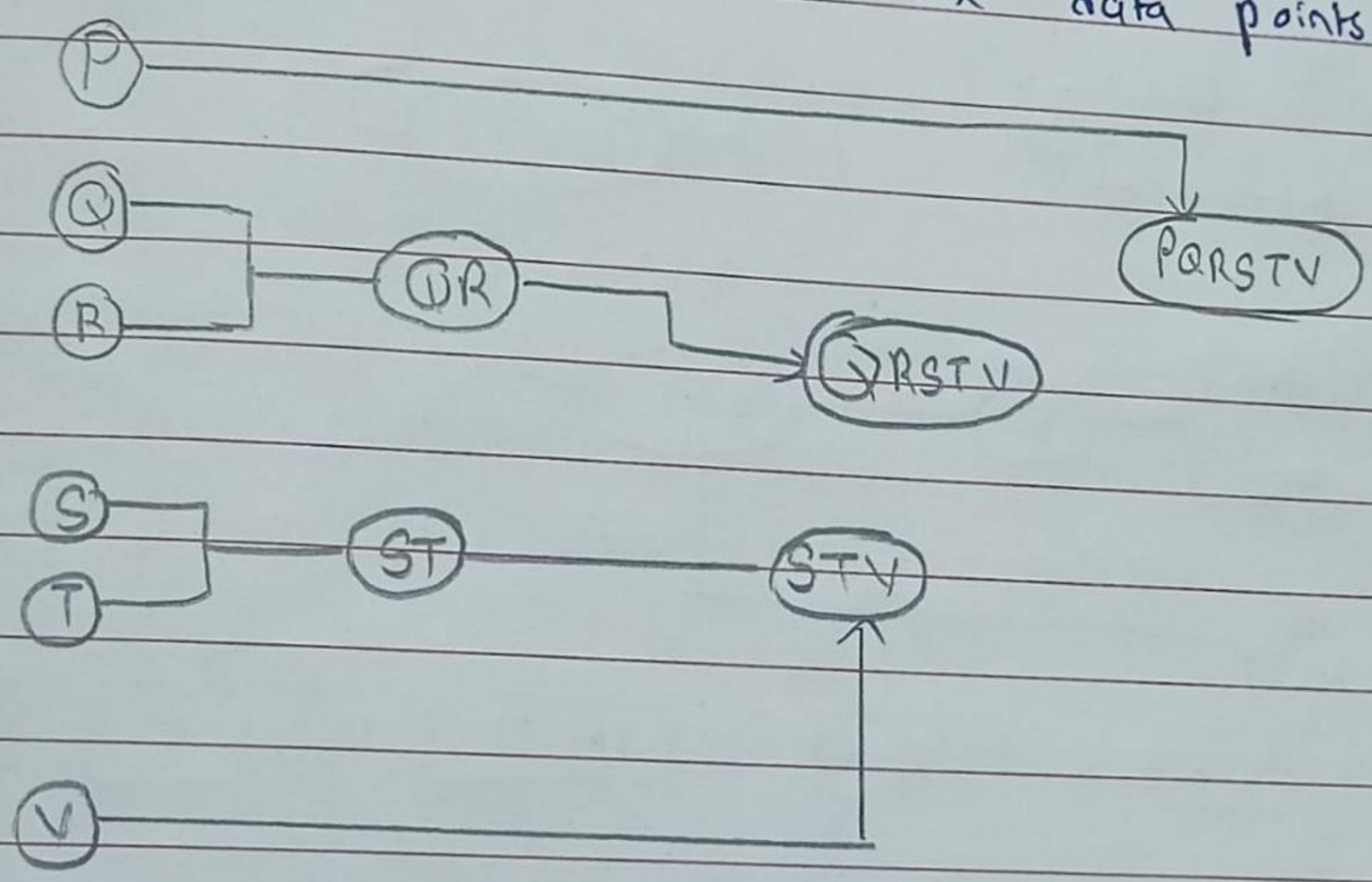


Scanned by TapScanner

## Agglomerative clustering:-

Agglomerative clustering is one of the most common types of hirarchical clustering used to group similar objects in clusters. Agglomerative clustering is also known as AGNES (Agglomerative Nesting). In agglomerative clustering, each data point act as an individual cluster and at each step, data objects are grouped in a bottom-up method. Initially, each data object is in its cluster. At each iteration, the clusters are combined with different clusters until one cluster is formed.

Let's suppose we have six data points P, Q, R, S, T, V.



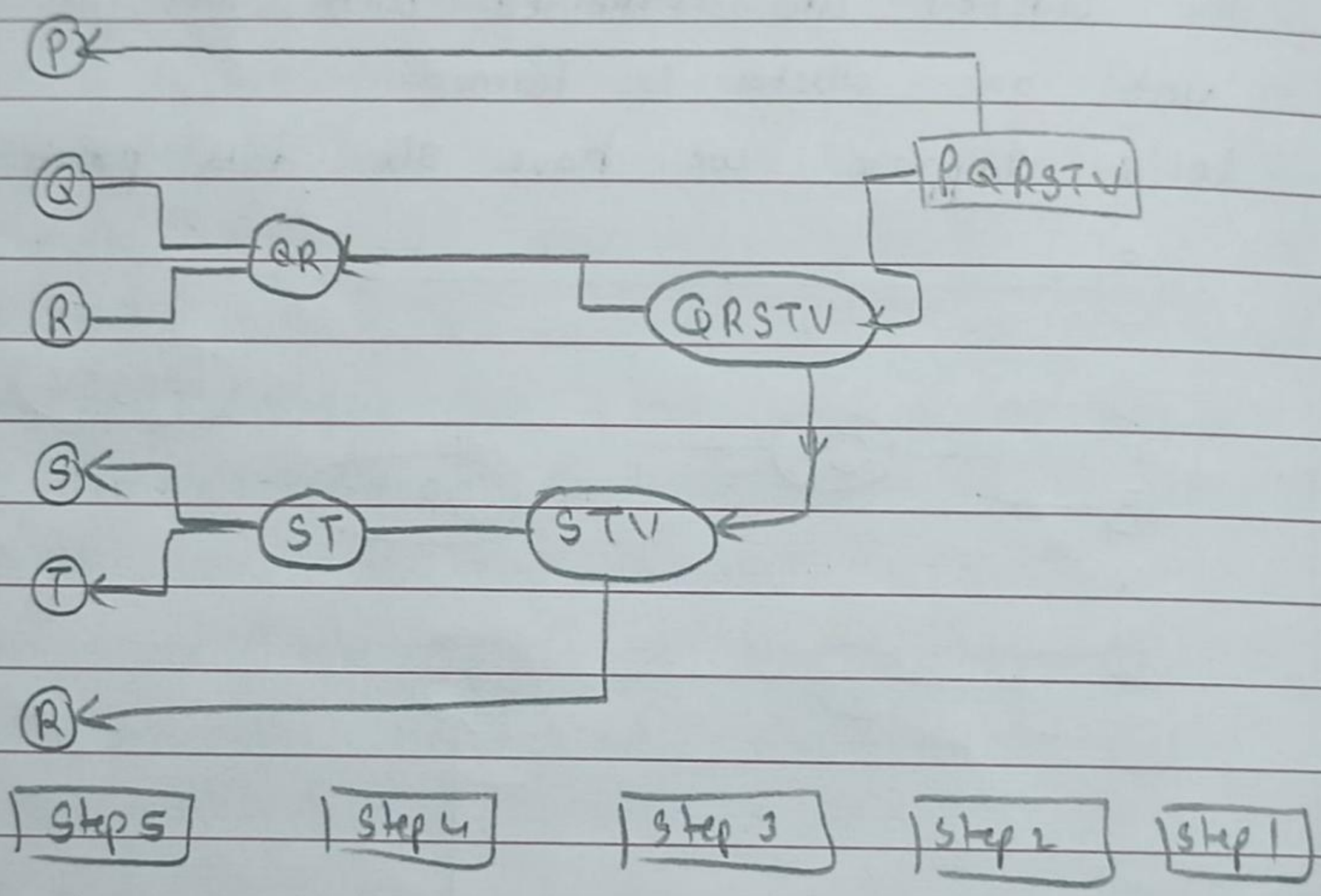

Step 1 | Step 2 | Step 3 | Step 4 | Step 5



Scanned by TapScanner

Divisive hirarchical clustering:-

Divisive hirarchical clustering is exactly the opposite of Agglomerative Hirarchical clustering. In Divisive Hirarchical clustering, all the data points are considered an individual cluster and in every iteration the data points that are not similar are seperated from the cluster. The seperated data points are treated as an individual cluster. Finally, we are left with N clusters.






Scanned by TapScanner

2) Explain about R lists, Vectors and Arrays?

A) R-lists:-

A list in R is a generic object consisting of an ordered collection of objects. Lists are one-dimensional, heterogeneous data structures. The list can be a list of vectors, a list of matrices, a list of functions and so on.

Creating a list:-

To create a list in R we need to use the function called "list()."

Ex:

```
EmpId = c(1,2,3,4,5)
EmpName = c("Kaushik", "Bhanu", "Vamshi", "Vinay",
            "lokesh")
number of Emp = 5
Emplist = list(EmpId, EmpName, number of Emp)
print (Emplist)
```

output:

```
[[1]]
[1] 1 2 3 4 5
[[2]]
[2] "Kaushik" "bhanu" "Vamshi" "Vinay" "lokesh"
[[3]]
[1] 5
```



Scanned by TapScanner

Vectors:-

Vectors in R are same as arrays in c language which are used to hold multiple data values of same type. one major key point is that in R the indexing of the vector will start from '1' not from '0'. we can create numeric vectors and characters vectors as well.

Ex: $v_1 \leftarrow c(4,5,6,7)$

type of ($v_1$)

$v_2 \leftarrow c(1L, 4L, 2L, 5L)$

type of ($v_2$)

Output:

[1] "integer"

[1] "double"

Arrays:-

An array in R can be created with the use of array() function. List of Element is passed to array() functions along with the dimensions as required.

Syntax: array(data, dim(nrow, ncol, nmat), dimnames = names)

Ex: vec1 ← c(1,2,3,4,5,6,7,8,9)

print(vec1)

cat("length of vector :", length(vec1))

Output: [1] 1 2 3 4 5 6 7 8 9

length of vector : 9



CMRIT

Scanned by TapScanner

4) Explain about ANOVA in R with example.

A) **ANOVA:**

Analysis of variance (ANOVA) is a statistical technique, commonly used to studying differences b/w two or more group means. ANOVA test is centered on the different sources of variation in a typical variable. ANOVA in R primarily provides evidence of the existence of the mean equality b/w the groups. This statistical method is an extension of the t-test. It is used in a situation where the factor variable has more than one group.

ANOVA test involves setting up:

- Null Hypothesis: All population means are equal.
- Alternate Hypothesis: Atleast one population mean is different from other.

ANOVA test are of two types:

- One way test: It takes one categorical group into consideration.
- Two way test: It takes two categorical group into consideration.

To get started with ANOVA, we need to install and load the dplyr package.



Scanned by TapScanner

Ex: One way ANOVA

```
install. packages ("dplyr")
library (dplyr)
boxplot (mtcars$ disp ~factor (mtcars$ gear),
        xlab = "gear", ylab = "disp")
mtcars_aov ← aov (mtcars$ disp ~factor (mtcars$gear))
summary (mtcars_aov)
```

output:

| | Df | sum sq | mean sq | Fvalue | Pr(>F) |
|---|---|---|---|---|---|
| Factor | 2 | 28021 | 14011 | 2.073 | 2.56 -06*** |
| Residuals | 29 | 195964 | 6757 | | |

---

Sign if. Codes : 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '' 1

Two - way test:-

Ex: install. packages ("dplyr")

```
library (dplyr)
boxplot (mtcars$ disp ~ mtcars$ gear, subset = (mtcars$am==0),
        xlab = "gear", ylab = "disp", main = "Automatic")
boxplot (mtcars$ disp ~mtcars$ gear, Subset = (mtcars$ am ==1),
        xlab = "gear", ylab = "disp", main = "manual")
mtcars_aov2 ← aov (mtcars$ disp ~ factor (mtcars$ gear) *
                factor (mtcars$am))
summary (mtcars_aov2)
```



CMRIT

Scanned by TapScanner

output:

| | Df | Sum sq | mean sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| factor(mtcars$gear) | 2 | 280221 | 140110 | 20.695 | $3.03e^{-06}$ |
| factor(mtcars$am) | 1 | 6399 | 6399 | 0.945 | 0.339 |
| Residuals | 28 | 189565 | 6770 | | |

---

signif codes: 0 0.001 0.01 0.05 0.1 1

5) Define Regression? Explain multiple Regression in R with Example?

A) Regression:

Regression is a supervised learning technique which helps in finding the correlation b/w variables and enables us to predict the continous output variable based on one or more predictor variables.

Multiple linear regression:-

Multiple linear regression is used to estimate relationship b/w two or more independent and one dependent variable.

* It is one of the important regression algorithms which models the linear relationship b/w a single dependent continous variable and more than one independent variable.



Scanned by TapScanner

Use of multiple regression:-

* To know the how strong the relationship is b/w two or more independent variables and one dependent variables.
* To know the value of dependent variable at a certain value of the independent variables

Formula for multiple linear regression:

$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon$

$\beta_1 x_1$ = regression coefficient of first independent variable ($x_1$)

$\beta_n x_n$ = regression coefficient of the last independent variable ($x_n$).

Ex:

```
input <- mtcars[c("mpg", "disp", "hp", "wt")]
model <- lm(mpg~disp + hp+ wt, data = input)
print (model)
a <- coef (model)
print (a)
xdisp <- coef (model)
xhp <- coef (model)
xwt <- coef (model)
print (xdisp)
print (xhp)
print (xwt)
```



CMRIT

Scanned by TapScanner

output:

```
Call
lm (formula = mpg ~ disp + hp + wt, data = input)
Coefficients
(Intercept)      disp        hp        wt
37.105505   -0.000937   -0.031157   -3.800891
(Intercept)
  37.10551
      disp
 -0.000937009 1
        hp
 -0.031156 55
        wt
 -3.800891
```

3) Illustrate the use of R Programming in statistics?

A) R - statistics:-

Statistics is a form of mathematical analysis that concerns the collection, organization, analysis, interpretation and presentation of data. The statistical analysis helps to make the best usage of the vast data available and improves the efficiency of solutions.



CMRIT

Scanned by TapScanner

R is programming language and is used for Environment statistical computing and graphics. The following is an introduction to basic Statistical concepts like plotting graphs such as bar charts, pie charts, Histograms and box plots.

→ The following software is required to learn and implement statistics in R:

- R software
- R Studio IDE

Functions for plotting graphs in Statistics in R Programming language:

1) plot() function: This function is used to Draw a Scatter plot with axes and titles.

Syntax: plot (x, y = NULL, ylim = NULL, xlim = NULL, type = "b"...)

2) data() function():- This function is used to load specified data sets.

Syntax: data (list = character(), lib.loc = NULL, package = NULL....)



Scanned by TapScanner

# I - ASSIGNMENT

(Start Writing From Here)

3) table() function: The table function is used to build a contingency table of the counts at each combination of factor levels.

Syntax: table(x, row.names = NULL, ....)

4) barplot() function: It creates a bar plot with vertical/horizontal bars.

Syntax: barplot(height, width = 1, name.arg = NULL, space = NULL....)

5) Pie() function: this function is used to create a pie chart.

Syntax: pie(x, labels = names(x), radius = 0.6, edges = 100, clockwise = TRUE....)

6) hist() function: The function hist() creates a histrogram of the given data values.

Syntax: hist(x, breaks = "Sturges", probability = ! freq, freq = NULL, ....)



CMRIT

Scanned by TapScanner

Set-4

1). a). Illustrate the k-Means Clustering Algorithm with an example.

* The k Means clustering algorithm divides a set n observations into k clusters. Use k means clustering when you don't have existing group labels and want to assign similar data points to the number of groups you specify (k).

* In general, clustering is a method of assigning comparable data points to groups using data patterns. Clustering algorithms find similar data points and allocate them to the same set. k means clustering is one such algorithm. A cluster is a group or set of similar data points, and I'll use those terms synonymously.

"k-Means" refers to the following:-

* The number of clusters you specify (k).
* The process of assigning observations to the cluster with the nearest centre (mean).

* Use k means clustering to generate groups comprised of observations with similar characteristics. For example, if you have customer data, you might want to create sets of similar customers and then target each group with different types of marketing.

* K means clustering is a popular machine learning algorithm. It's an unsupervised method because it starts without labels and then forms and label groups itself. K means clustering is not a supervised learning method because it does not attempt to predict existing or known group labels.

* k means clustering usage has grown recently thanks to machine learning taking off. But its started as a statistical grouping method for signal processing in 1957.

## K- Means Clustering Algorithm Works As:-

The k-Means Clustering algorithm finds observation in a dataset that are like each other and places them in a set. The process starts by randomly assigned each data points to an initial group.



CMRIT

and calculating the centroid for each one. A centroid is the center of the group.

Then the assigned continues as follows:-

1. It evaluates each observation, assigning it to the closest cluster. The definition of "closest" is that the Euclidean distance between a data point and a group's centroid is shorter than the distances to the other centroid's.

2. When a cluster gains or loses a data point, the k means clustering algorithm recalculates its centroid.

3. The algorithm repeats until it can no longer assign data points to a closer set.

## K- MEANS CLUSTERING EXAMPLE:-

Imagine you're studying businesses in a specific industry and documentating their information. Specifically you record the variables shown in the data set snippet below.

| Company | Clients | Rate of Return | Sales | Years |
|---|---|---|---|---|
| A | 150 | 15.4 | 50400300 | 18 |
| B | 140 | 11.3 | 42100650 | 15 |
| C | 120 | 9.9 | 39440420 | 12 |
| D | 110 | 18.5 | 36500520 | 16 |
| E | 100 | 9.7 | 40650005 | 16 |
| F | 99 | 15.2 | 45685230 | 12 |
| G | 56 | 9.2 | 25978080 | 8 |

Interpreting the Results:-

The downloadable dataset contains the k mean clustering assignments for each business. We'll look at some of the groups

k-Means Cluster Analysis : Clients, Rate of Return, Sales, years.
Number of clusters: 3.

The statistical output shows that k means clustering has created the following three sets with the indicated

* cluster1: 6
* cluster: 6
* Cluster2: 10

| | Number of Observations | Within cluster sum of squares | Avg. dist of centroid | Max distance from centroid |
|---|---|---|---|---|
| Cluster 1 | 6 | 1.47960E+15 | 1310396.667 | 2715650.000 |
| Cluster 2 | 10 | 1.197954E+14 | 2900172.501 | 5164312.000 |
| Cluster 3 | 6 | 1.01890E+14 | 3403733.334 | 4721505.000 |

Cluster Centroids:-

| Variable | Cluster 1 | Cluster 2 | Cluster 3 | Grand centroid |
|---|---|---|---|---|
| Clients | 132.1667 | 106.3000 | 62.0000 | 101.2727 |
| Rate of Return | 14.0000 | 11.0400 | 9.6167 | 11.4591 |
| Sales | 4.76846E+07 | 3.16302E+07 | 2.22905E+07 | 3.61887E+07 |
| Years | 14.6667 | 11.1000 | 6.8333 | 10.9091. |

b) Compare Agglomerative versus Divisive Hierarchial clustering approaches.

* Divisive clustering is more complex as compared to agglomerative clustering, as in the case of divisive clustering we need a flat clustering method as "sub-routine" to split each cluster until we have each data having its own singleton cluster.

* Divisive clustering is more efficient if we do not


CMRIT

generate a complete hierarchy all the way down to individual data leaves. The time complexity of a naive agglomerative clustering is $O(n^3)$ because we exhaustively scan the $N \times N$ matrix dist-mat for lowest distance in each of $N-1$ iterations. Using priority queue data structure we can reduce this complexity to $O(n^2 \log n)$. By using some more optimizations it can be brought down to $O(n^2)$.

* A divisive algorithm is also more accurate. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without clustering the local patterns or neighbor global distribution of data. These early decisions cannot be undone.

2). Define Hypothesis? Explain the Null and Alternative Hypothesis with an Example.

Hypothesis testing is a technique used to determine whether an assumption about the population is true. ~~Null~~ There are two types of Hypothesis:-

1) Null Hypothesis
2) Alternate Hypothesis

Having a good understanding about null and alternate hypothesis will help you better design good hypothesis tests and understand their results in a nice manner. It is very important for data scientists to be able to distinguish between null and alternate hypothesis and design hypothesis tests. ~~In this~~

**Null Hypothesis:-** Null Hypothesis represent the claim which is well-established or true justified belief. In addition to well-established beliefs, it represents the default state of the claim in the real world. For example, null hypothesis can be that a 1-kG packet of salt contains of 1kG of salt. Another example is there is no relationship between pranayama and LDL Cholesterol as stated in the previous section.

→ Null hypothesis is typically denoted by H0.

**Alternate hypothesis:-**

→ Alternate hypothesis represents the new chain that needs to be established. For example, a 1-kG packet of salt contains of 900gm of salt. Or else there is an impact of pranayama on reducing LDL



CMRIT

cholesterol.

* Alternate Hypothesis is denoted by $H_1$.

What are some examples of Null And Alternate Hypothesis :-

* Investing in Stock Market :- Buying stocks during down market will earn higher returns.

Null Hypothesis :- It is not true that buying stocks during down market will return higher return.

Alternate Hypothesis :- Buying stocks during down market earns higher returns.

* New heart medicience :- New medicine reduces the risk of heart attack by 60%.

Null hypothesis :- It is not true that new medicine reduces the risk of heart attack by 60%.

Alternate Hypothesis :- New medicine reduces the risk of heart attack by at least 60%.

3). List out the types of variables in R.

Basic Syntax in R:-

There are two ways to write code in RStudio:

* First, in the command prompt and, second, in the R script file. We will go through both one by one.

Hello World Program in R:-

* Launch RStudio

* In the command prompt window, type the following and press Enter.

```
> print ("Hello World!")
```

Output:-

```
[1] "Hello world!"
```

Variables in R programming:-

A variable is a name given to a memory location, which is used to store values in a computer program. Variables in R programming can be used to store numbers, words, matrices, and even tables, R is a dynamatically programmed language which means that unlike other programming languages, we do not have to declare the data type



CMRIT

of a variable before we can use it is in our program.

For a ~~prog~~ variable to be valid, it should follow these rules:-

* It should contain letters, numbers, and only dot or underscore characters.
* It should not start with a number (eg:- 2iota)
* It should not start with a dot followed by a number (Eg:- .2iota)
* It should not start with an underscore Eg:- (_iota)
* It should not be a reserved keyword.

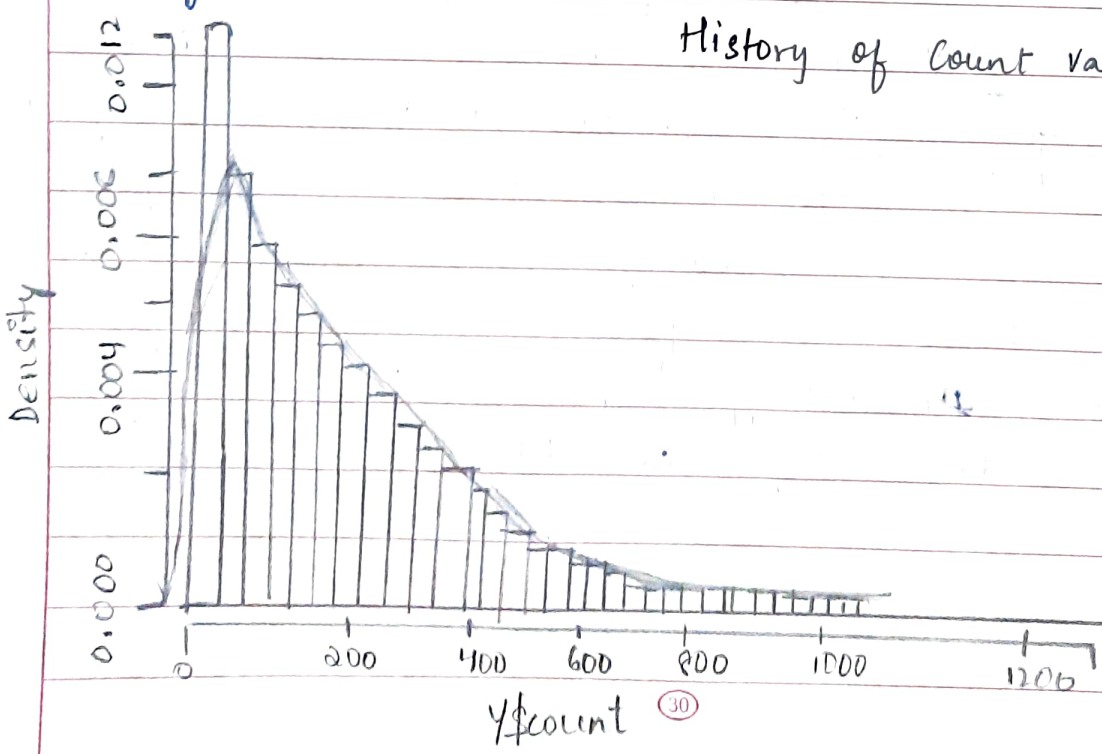
Each variable in R has an associated datatype. Each datatype requires different amounts of memory and has some specific operations which can be performed over it. R Programming Language has the following basic data types and the following table shows the datatype and the values that each data type can take

Datatypes in R Programming Language:-

| Basic Datatypes | Values |
|---|---|
| 1) Numeric | Set of all real Numbers |
| 2) Integer | Set of all integers, Z |
| 3) Logical | TRUE or FALSE |
| 4) complex | Set of complex Numbers |
| 5) Character | "a", "b", "c", ..., "#", "$", ..., "1", "2" etc |

4). Explain Maximum likelihood test in R, with example.

Let us say we want to predict the sale of tickets for an event. The data has the following histogram and density.

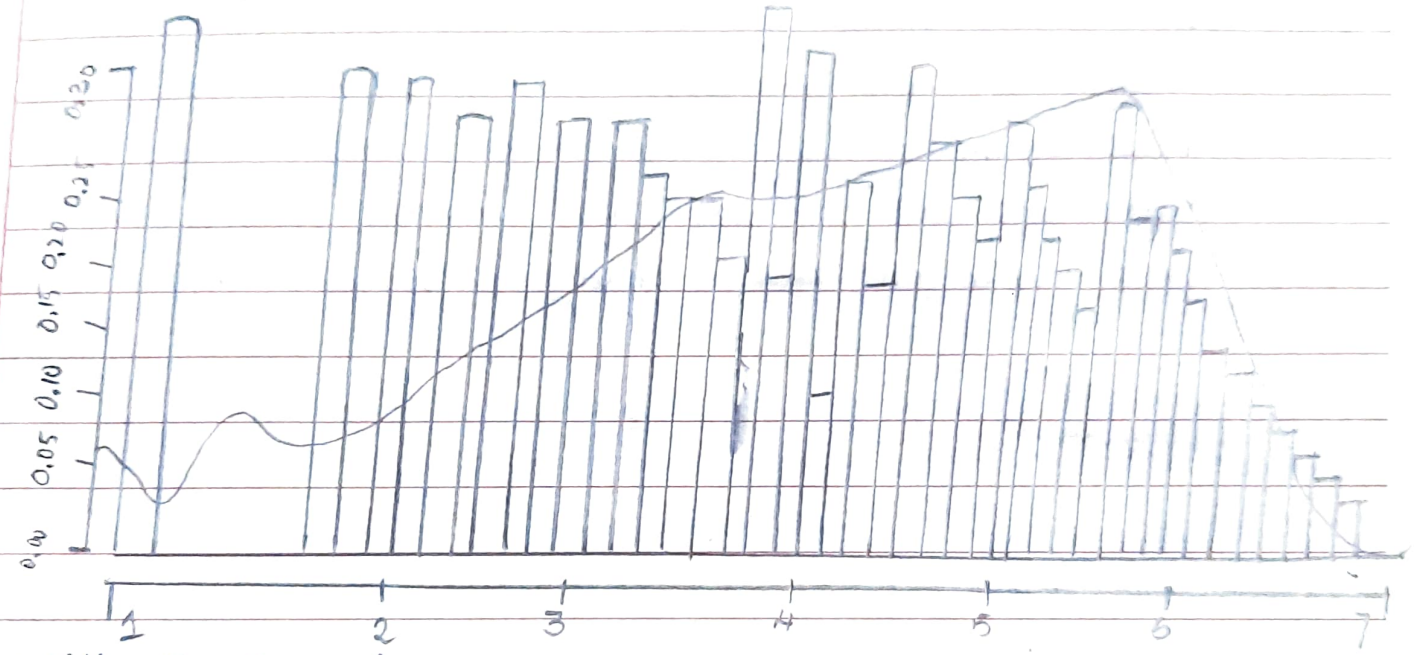




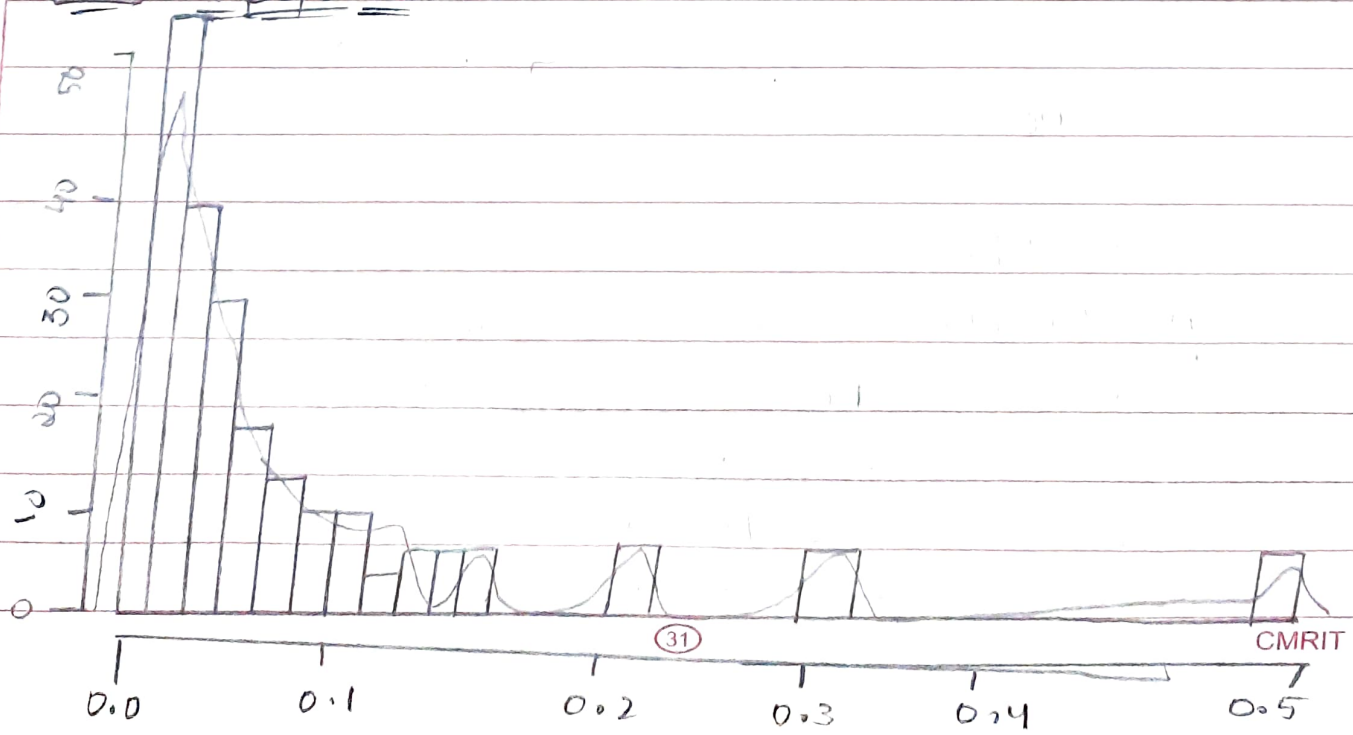

CMRIT

The variable is not normally distributed and is asymetric and hence it violates the assumptions of linear regression.

With log transformation:-




With Reciprocal:-

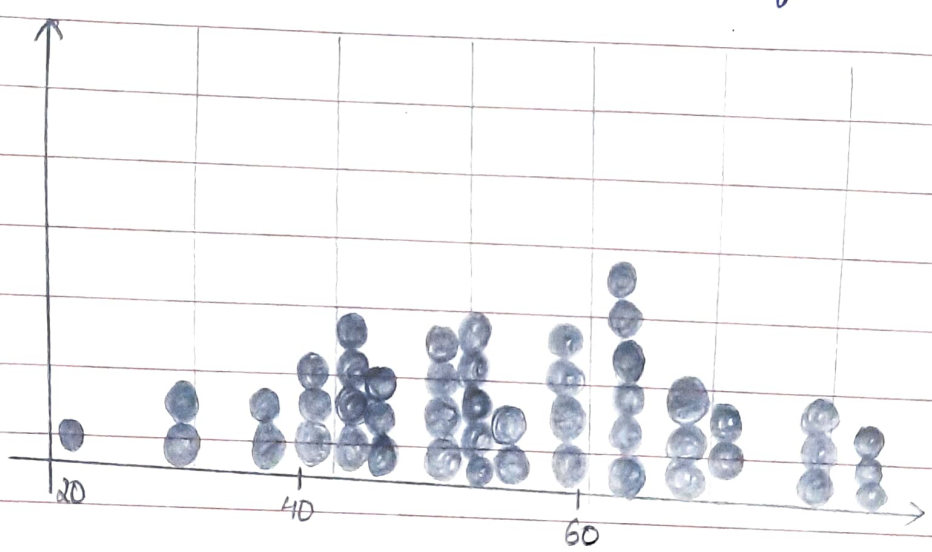




CMRIT

## Understanding MLE with an Example:-

While studying stats and probability, you must have come across problem like - What is the probability of $x > 100$, given follows a normal distrubution with mean 50 and standard deviation (sd) 10. In such problems, we already know the distrubution and its parameters but in real life problems these quantities are unknown and must be estimated from the data.




Method likelihood Estimation can be defined as a method for estimating population parameters from sample data such that the probability of obtaining the observed data is maximized.

1. Mean = 100, SD = 10
2. Mean = 50, SD = 10



CMRIT

5). Define regression? Poisson regression in R with example.

Regression refers to a mining technique that is used to predict the numeric values in a group given data set. For example, regression right might be used to predict the product or service cost or other variables. It is also used in various industries for business and marketing behavior, trend analysis, and financial forecast.

Regression can predict all the ~~reg~~ dependent data sets, expressed in the expression of independent variables, and the trend is available for a finite period Regression can predict all the dependent data sets, expressed in the regression of indepent variables, and the trend is available for a finite period. Regression provides a good way to predict variables, but there are also correleated. The marginal distributions of A and B need to be derived and used. Before applying regression analysis, the data needs to be studied carefully and perform certain preliminary tests to ensure the regression is applicable. There are non-parametric tests that are available in such cases.

## Poisson Regression :-

Poission Regression involves regression models in which the response-variable is in the form of counts and not fractional numbers. For example, the count of number of births or number of wins in a football match series Also the values of the response variables follow a Poisson distrubution.

The general mathematical equation for Poisson regression is :-

$$\boxed{\log(y) = a + b1x1 + b2x2 + __ + bnxn}$$

### Syntax :-

The basic syntax of glm() function in Poission regression is :-

glm(formula, data, family)

Following is the description of the parameters used :-

* y is the response variable
* a and b :- are the numeric coefficents
* x is the predictor variables.

* formula :- is the symbol presenting the relationship between the variables.

* Data:- is the data set giving the values is "poisson" for Logistic Regression.

Example:-

Input data :-

```
input <- Warpbreaks
print (head (input))
```

When we execute the above code, it produces the following result :-

| | breaks | Wool | tension |
|---|---|---|---|
| 1 | 26 | A | L |
| 2 | 30 | A | L |
| 3 | 54 | A | L |
| 4 | 25 | A | L |
| 5 | 70 | A | L |
| 6 | 52 | A | L |

Create Regression Model :-