PRACTICAL NO. 10

Aim: To perform classification and clustering using R-programming.

Classification:
Theory:
Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. It is mostly used in Machine Learning and Data Mining applications using R.
Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.
Examples of use of decision tress is − predicting an email as spam or not spam, predicting of a tumor is cancerous or predicting a loan as a good or bad credit risk based on the factors in each of these. Generally, a model is created with observed data also called training data. Then a set of validation data is used to verify and improve the model. R has packages which are used to create and visualize decision trees. For new set of predictor variable, we use this model to arrive at a decision on the category (yes/No, spam/not spam) of the data.
The R package "party" or "rpart" can be used to create decision trees.

Basic Commands:
1. Install Package:
    install.packages("<name of package")

2. To check installed or not
    library("<name of package")

3. The basic syntax for creating a decision tree in R is −

```
ctree(formula, data)
```

Following is the description of the parameters used −

- formula is a formula describing the predictor and response variables.
- data is the name of the data set used.

The package "party" has the function **ctree()** which is used to create and analyze decision tree.

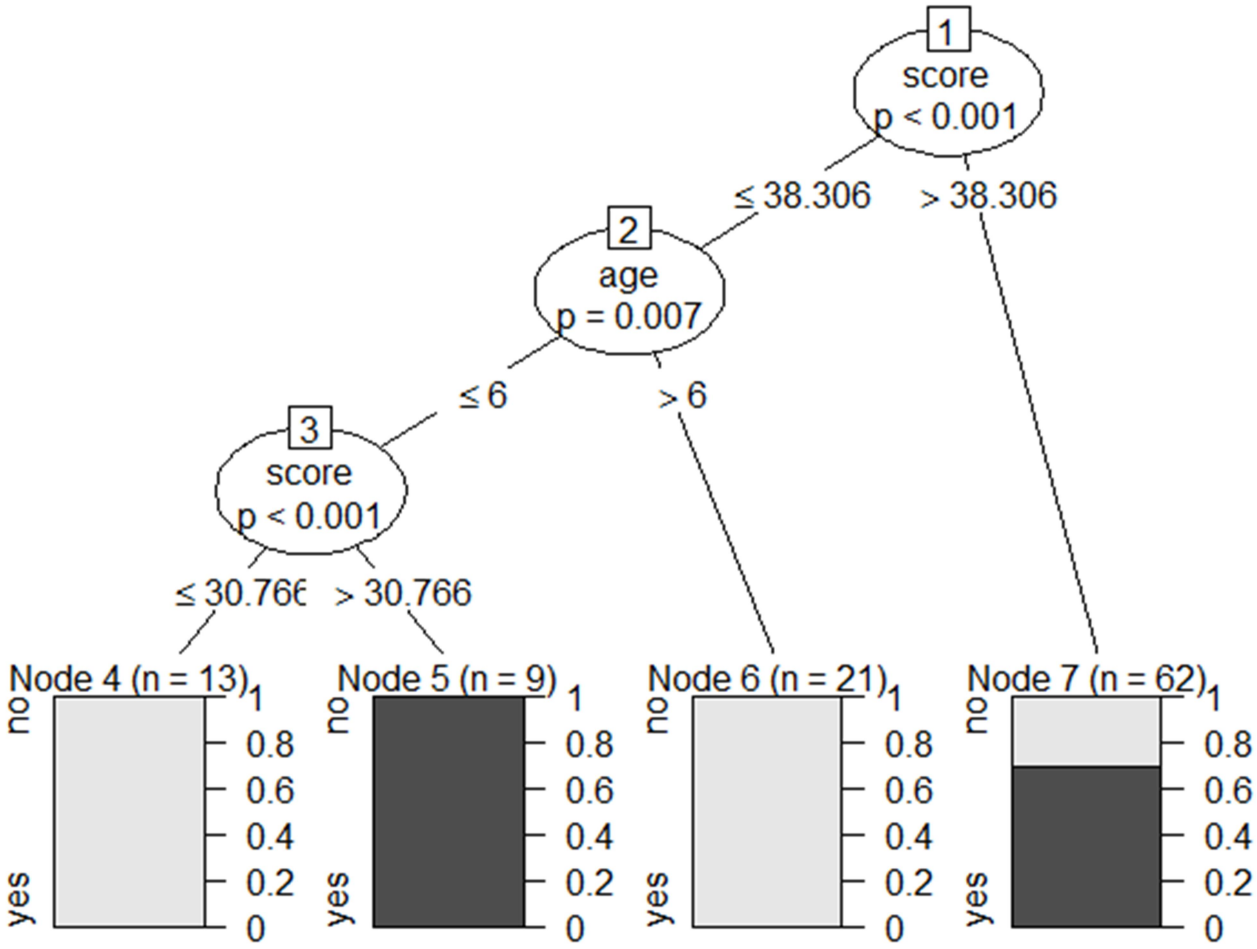
Questions:

Q1. Build Decision Tree as directed:

1. Use R in-built data set named **readingSkills** to create a decision tree. It describes the score of someone's readingSkills if we know the variables "age","shoesize","score" and whether the person is a native speaker or not.
2. Load the party package.
3. Print some records from data set readingSkills
    Print(head(readingSkills)

4. Use the **ctree()** function to create the decision tree and see its graph and make conclusion.

```
readingSkills

readingSkills[c(1:10),]
inputdata<-readingSkills[c(1:105),]
outtree<-ctree(nativeSpeaker~age+shoeSize+score, data=inputdata)
plot(outtree)
```



Conclusion: From the decision tree shown above we can conclude that anyone whose readingSkills score is less than 38.3 and age is more than 6 is not a native Speaker.

Q2. Build a decision tree for the **iris data with function ctree() in package party**

- Sepal.Length, Sepal.Width, Petal.Length and Petal.Width are used to predict the Species of flowers. In the package, function ctree() builds a decision tree, and predict() makes prediction for new data.
- Before modeling, the iris data is split below into two subsets: training (70%) and test (30%).
- The random seed is set to a fixed value below to make the results reproducible.
- Use myFormula to specify that Species is the target variable and all other variables are independent variables.
- check the prediction & plot.
- predict on test data

```
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
```

```
attributes(iris)
```

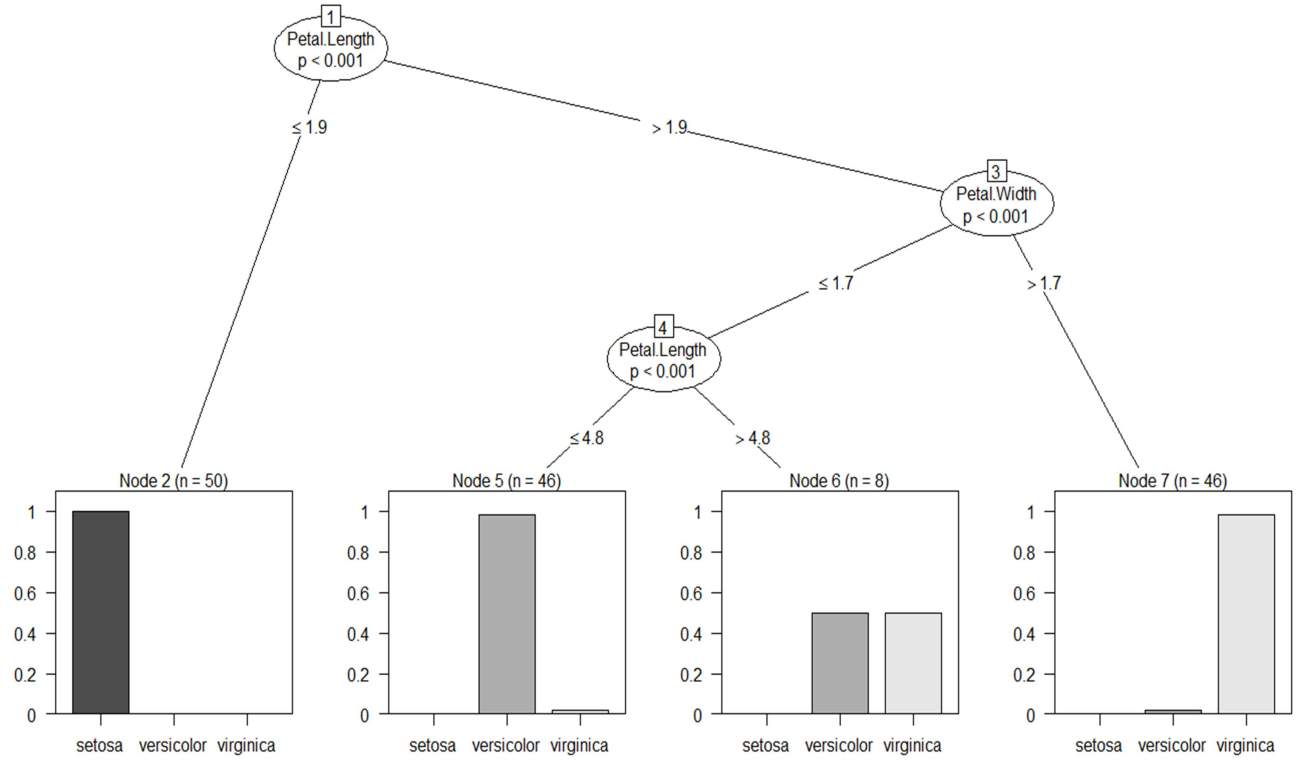
```
$`names`
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"

$class
[1] "data.frame"

$row.names
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
 [20]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
 [39]  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57
 [58]  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76
 [77]  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
 [96]  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
[115] 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133
[134] 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
```

```
inputiris<-iris
iristree<-ctree(Species~.,data=inputiris)
plot(iristree)
```

(instead of using plot function)

```
iristree
```

```
        Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  150

1) Petal.Length <= 1.9; criterion = 1, statistic = 140.264
  2)*  weights = 50
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 67.894
    4) Petal.Length <= 4.8; criterion = 0.999, statistic = 13.865
      5)*  weights = 46
    4) Petal.Length > 4.8
      6)*  weights = 8
  3) Petal.Width > 1.7
    7)*  weights = 46
```



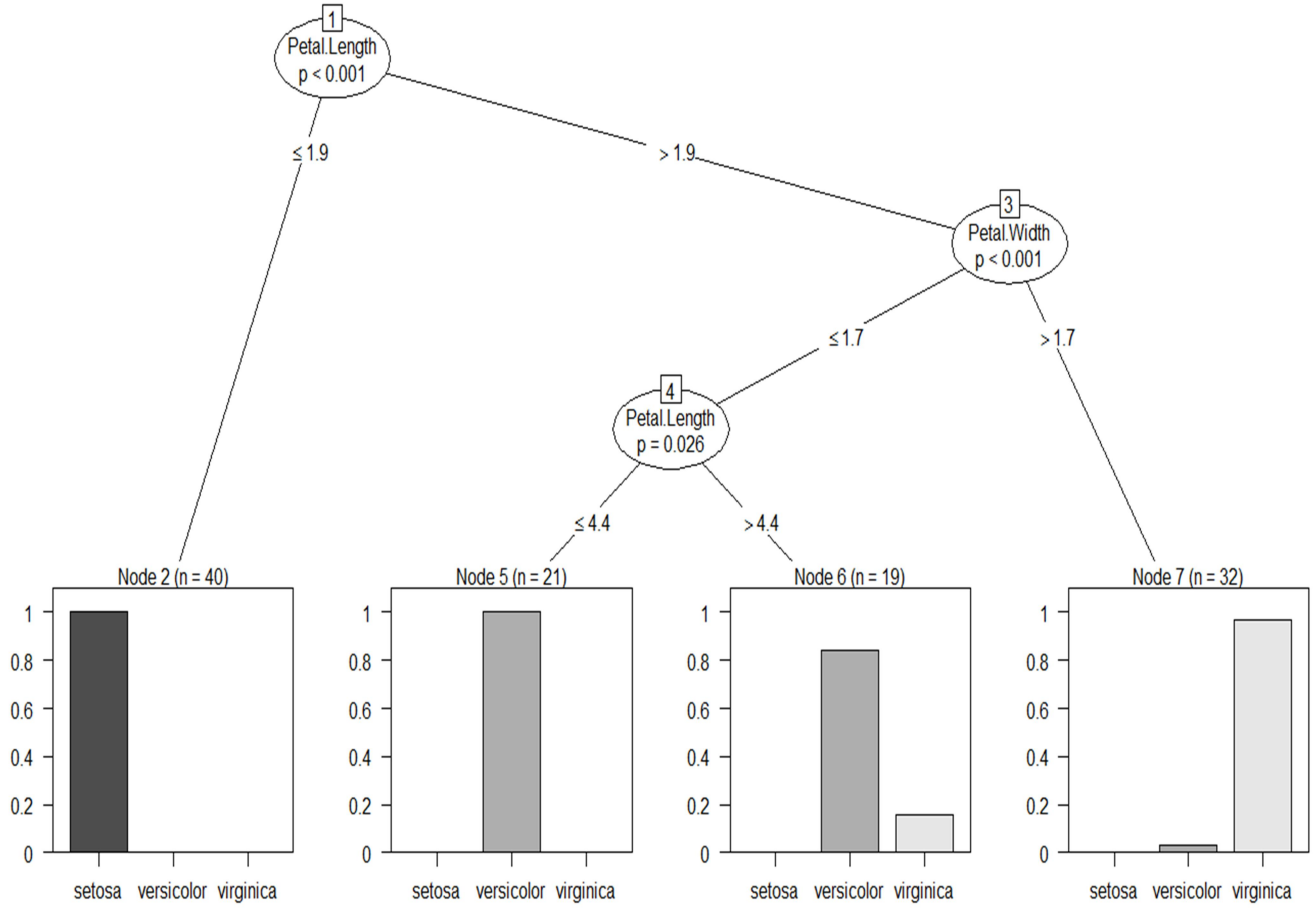

```
str(iris)
set.seed(1234)
ind<-sample(2,nrow(iris),replace=TRUE,prob=c(0.7,0.3))
trainData<-iris[ind==1,]
testData<-iris[ind==2,]
myFormula<-Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width
iris_ctree<-ctree(myFormula,data=trainData)
table(predict(iris_ctree),trainData$Species)
            ----This is the confusion matrix
```

```
            setosa versicolor virginica
  setosa         40          0         0
  versicolor      0         37         3
  virginica       0          1        31
```

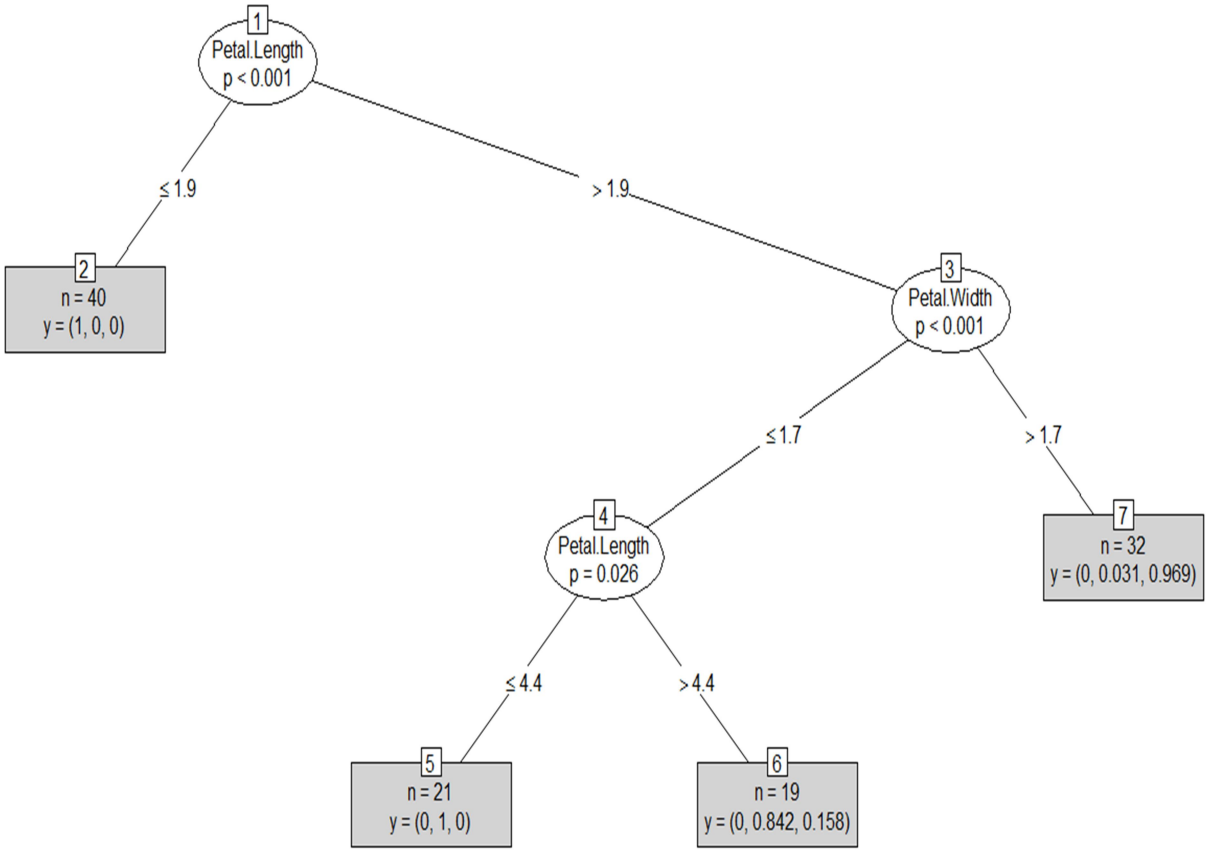
```
print(iris_ctree)
```

```

         Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  112

1) Petal.Length <= 1.9; criterion = 1, statistic = 104.643
  2)*  weights = 40
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 48.939
    4) Petal.Length <= 4.4; criterion = 0.974, statistic = 7.397
      5)*  weights = 21
    4) Petal.Length > 4.4
      6)*  weights = 19
  3) Petal.Width > 1.7
    7)*  weights = 32
```

```
plot(iris_ctree)
```




```
plot(iris_ctree,type="simple")
```




```
testPred<-predict(iris_ctree, newdata=testData)
table(testPred,testData$Species)
```

```
testPred     setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         12         2
  virginica       0          0        14
```

Clustering:
Theory:

Write Method for K-means and Hierarchical Clustering.
Differentiate between both the methods.

Syntaxes:

1. To check all the attributes of the dataset:

```
names(iris)
```

Part (A) Perform Clustering using K-Means in R using on iris dataset. Answer the Questions and write the commands corresponding to each:

1. Check for all attributes and name them.

"Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"

2. View the Complete iris dataset.

How many instances are there?

150

3. Use head command on iris dataset. What does it return?

->First 6 rows.

4. Assign 1st 4 columns to X , last column Species as class Variable to Y (assign the data from column 1-4 (features) to variable x, and the class to variable y)

```
> x = iris[,-5]          ///Do not consider 5th Column i.e. Species
> y=iris$Species
```

5. Create k-means model. Assume three Clusters as there are 3 possible classes.

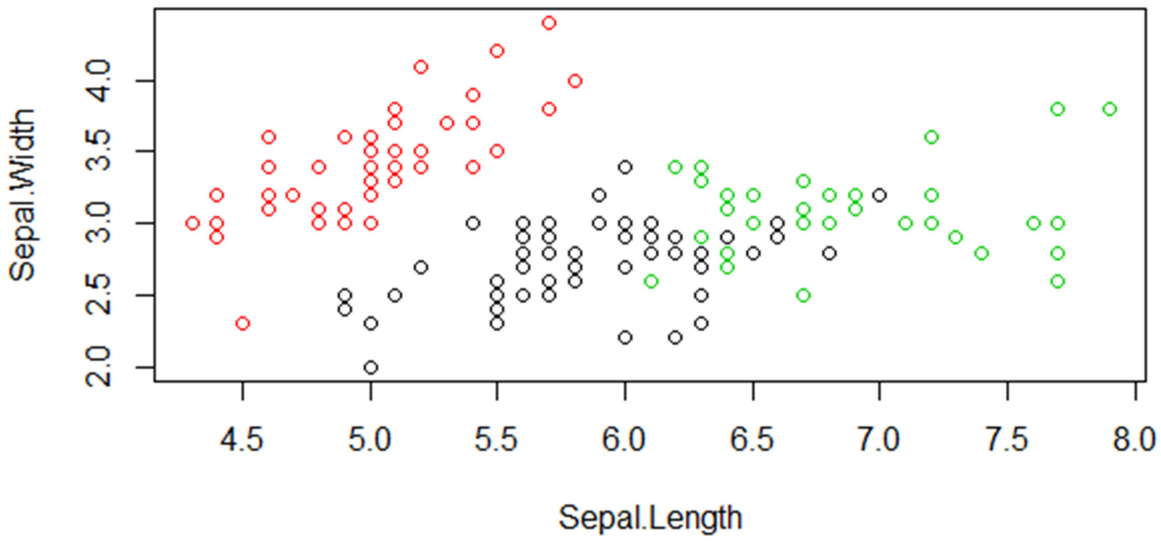
6. Show summary for the kmeans model. Give the analysis

```
kc<-kmeans(newiris,3)   //dataset, number of clusters
> kc
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
 [40] 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 3
 [79] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2
2 3 3
[118] 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 15.15100 39.82097 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

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

7. Draw the table to show the result. How many error and missing data? State the instances that belong to which cluster.

```
             1  2  3
setosa       0 50  0    //Setosa was all misclassified into category 2
versicolor 48  0  2
virginica  14  0 36
```

8. Draw the plot to visualize the cluster allotment.



9. Conclude about the Clustering on iris dataset.

Part (B) Perform Clustering using Hierarchical clustering in R using on iris dataset. Answer the Questions and write the commands corresponding to each:

1. Consider a sample of 40 records from the iris data, so that the Hierarchical clustering plot will not be over crowded.
2. Allot this to IrisSample variable

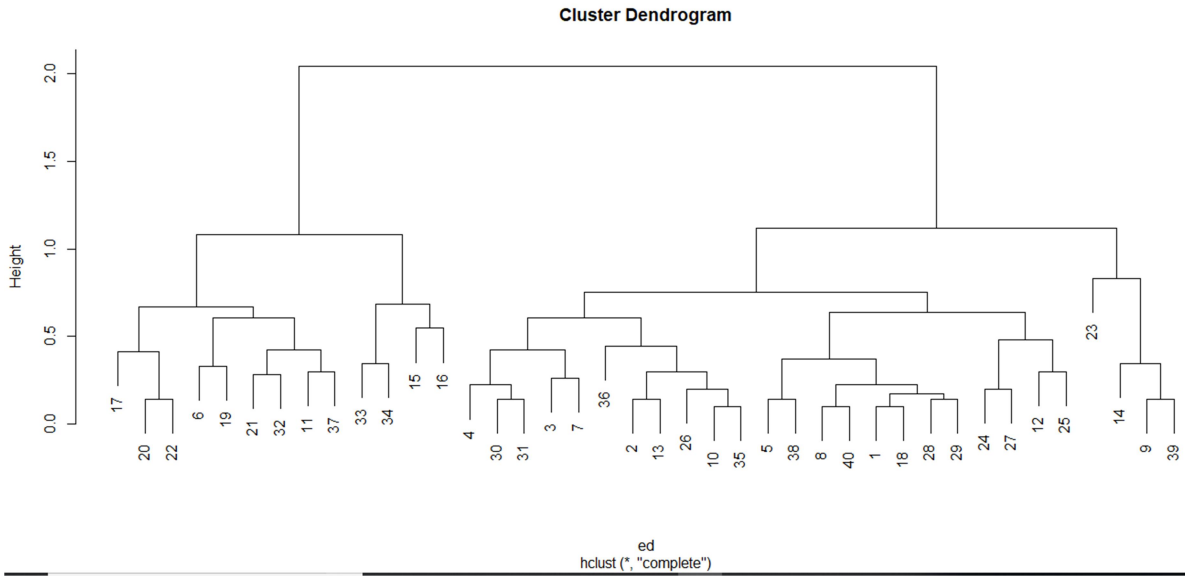
```
newiris<-iris
newiris<-newiris[1:40,]
```

3. Add species as null. Class variable will be found by the algorithm.

```
newiris$Species<-NULL
```

4. Perform Hierarchical clustering and view the dendrogram plot.

```
ed<-dist(newiris,method="euclidean")
hl<-hclust(ed,method="complete")
hplot(hl)
```



5. Cut tree into 3 clusters and show groups.

```
sub_grp <- cutree(hl, k = 3)
```

```
> table(sub_grp)
```

```
sub_grp
```

```
 1  2  3
```

```
23 13  4
```