EXP 8 : CLUSTERING MODEL

Clustering in R refers to the assimilation of the same kind of data in groups or clusters to distinguish one group from the others(gathering of the same type of data). This can be represented in graphical format through R. We use the KMeans model in this process.

What is the K Means algorithm? K Means is a clustering algorithm that repeatedly assigns a group amongst k groups present to a data point according to the features of the point. It is a centroid-based clustering method.The number of clusters is decided, cluster centers are selected in random farthest from one another, the distance between each data point and center is calculated using Euclidean distance, the data point is assigned to the cluster whose center is nearest to that point. This process is repeated until the center of clusters does not change and data points remain in the same cluster.

Step 1: The Iris dataset which is an inbuilt dataset in R using the Cluster package is used. It has 5 columns namely – Sepal length, Sepal width, Petal Length, Petal Width, and Species. Iris is a flower and here in this dataset 3 of its species Setosa, Versicolor, Verginica are mentioned. We will cluster the flowers according to their species.

```
> data(iris)
```

```
> head(iris) #will show top 6 rows only
```

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

Step 2: The next step is to separate the 3rd and 4th columns into separate object x. As we are using the unsupervised learning method,we are removing labels so that the huge input of petal length and petal width columns will be used by the machine to perform clustering unsupervised.

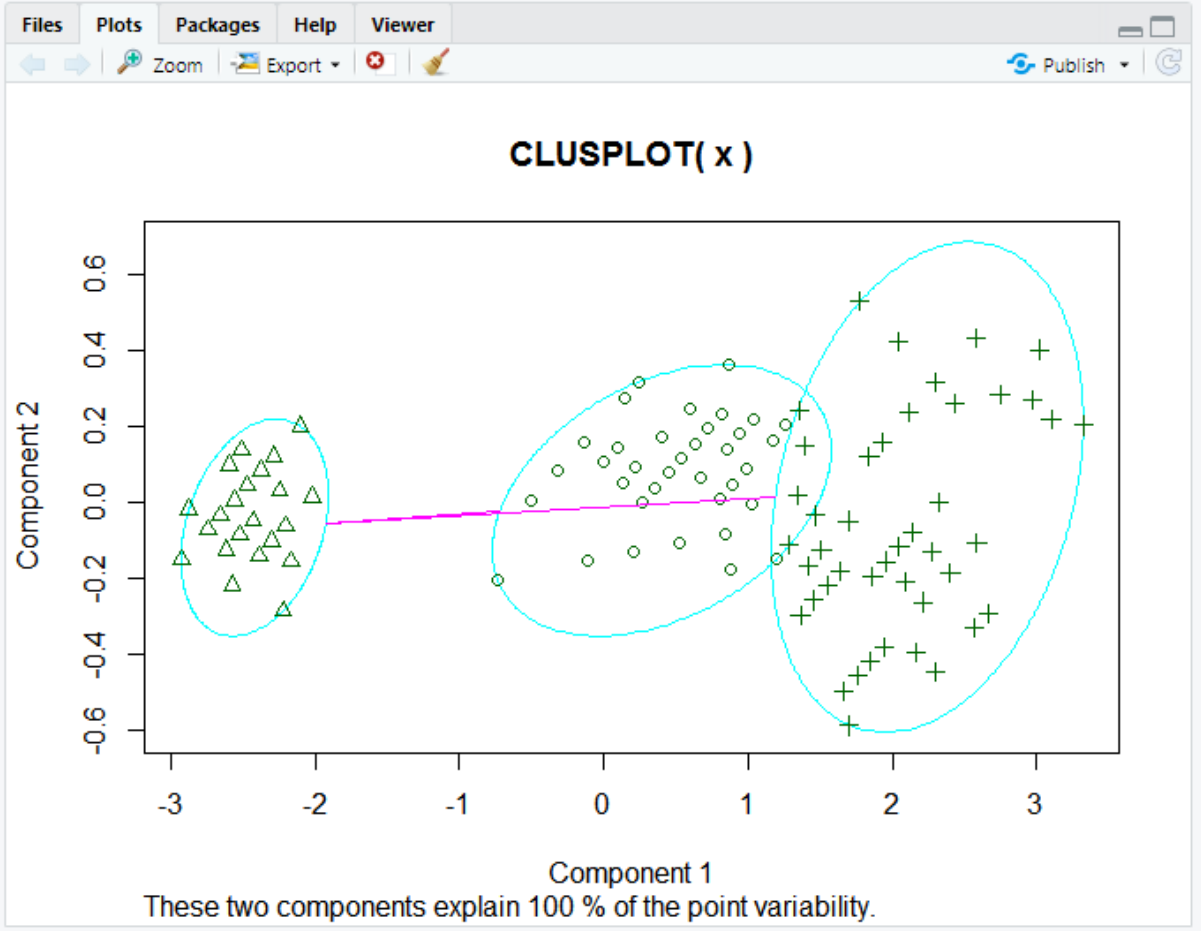
```
> x=iris[,3:4] #using only petal length and width columns
```

```
> head(x)

  Petal.Length Petal.Width
1          1.4         0.2
2          1.4         0.2
3          1.3         0.2
4          1.5         0.2
5          1.4         0.2
6          1.7         0.4

>
```

Step 3: The next step is to use the K Means algorithm. K Means is the method we use which has parameters (data, no. of clusters or groups). Here our data is the x object and we will have k=3 clusters as there are 3 species in the dataset.
Then the 'cluster' package is called. Clustering in R is done using this inbuilt package which will perform all the mathematics. Clusplot function creates a 2D graph of the clusters.

```
> model=kmeans(x,3)

> model

K-means clustering with 3 clusters of sizes 52, 50, 48

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

Clustering vector:
  [1] 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 2 2 2 2 2 2 2 2 2 2 2 2 2
 [30] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [59] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1
 [88] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[117] 3 3 3 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[146] 3 3 3 3 3

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

Available components:

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

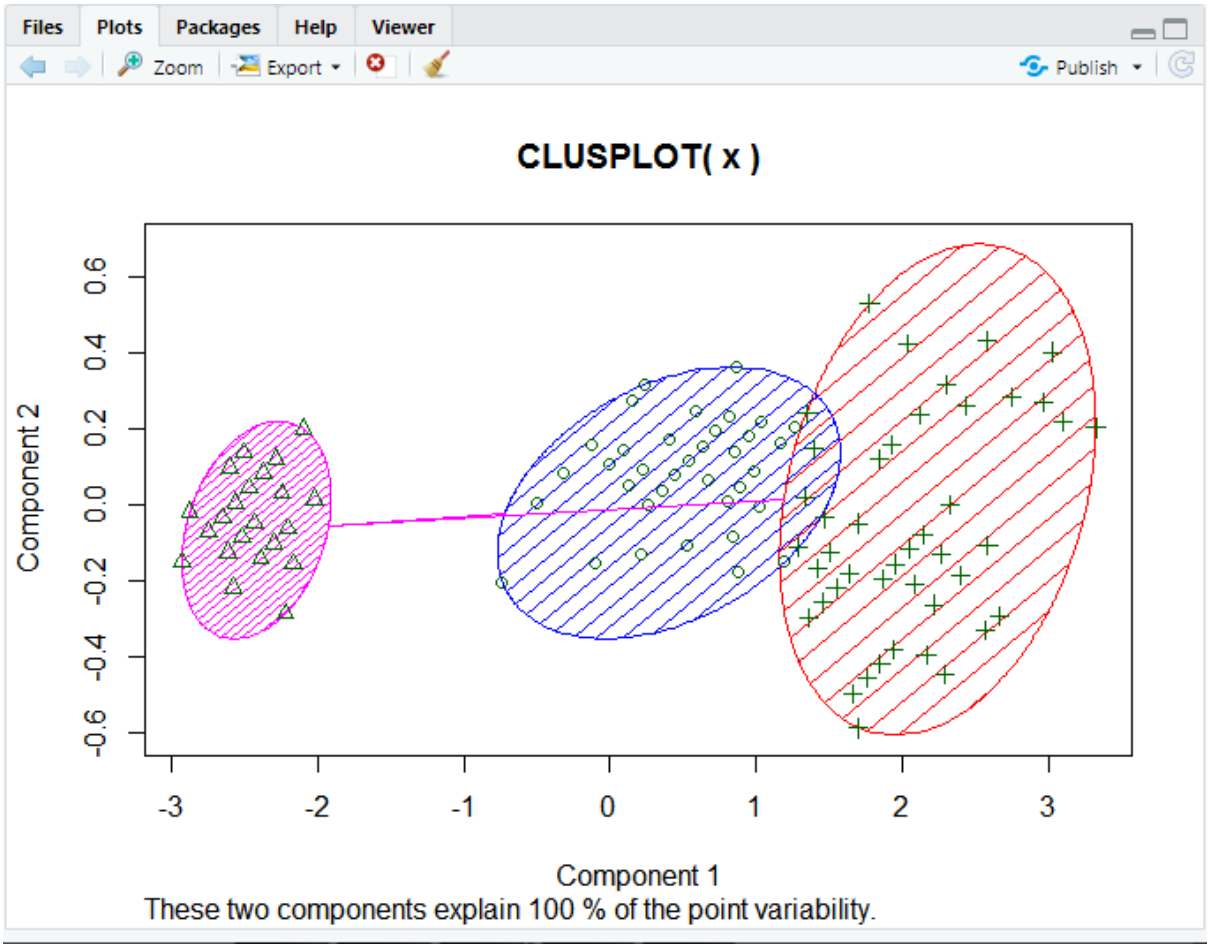
```
> library(cluster)
> clusplot(x,model$cluster)

>
```



Component 1 and Component 2 seen in the graph are the two components in PCA (Principal Component Analysis) which is basically a feature extraction method that uses the important components and removes the rest. It reduces the dimensionality of the data for easier KMeans application. All of this is done by the cluster package itself in R.These two components explain 100% variability in the output which means the data object x fed to PCA was precise enough to form clear clusters using KMeans and there is minimum (negligible) overlapping amongst them.

Step 4: The next step is to assign different colors to the clusters and shading them hence we use the color and shade parameters setting them to T which means true.

```
> clusplot(x,model$cluster,color=TRUE,shade=TRUE)
```

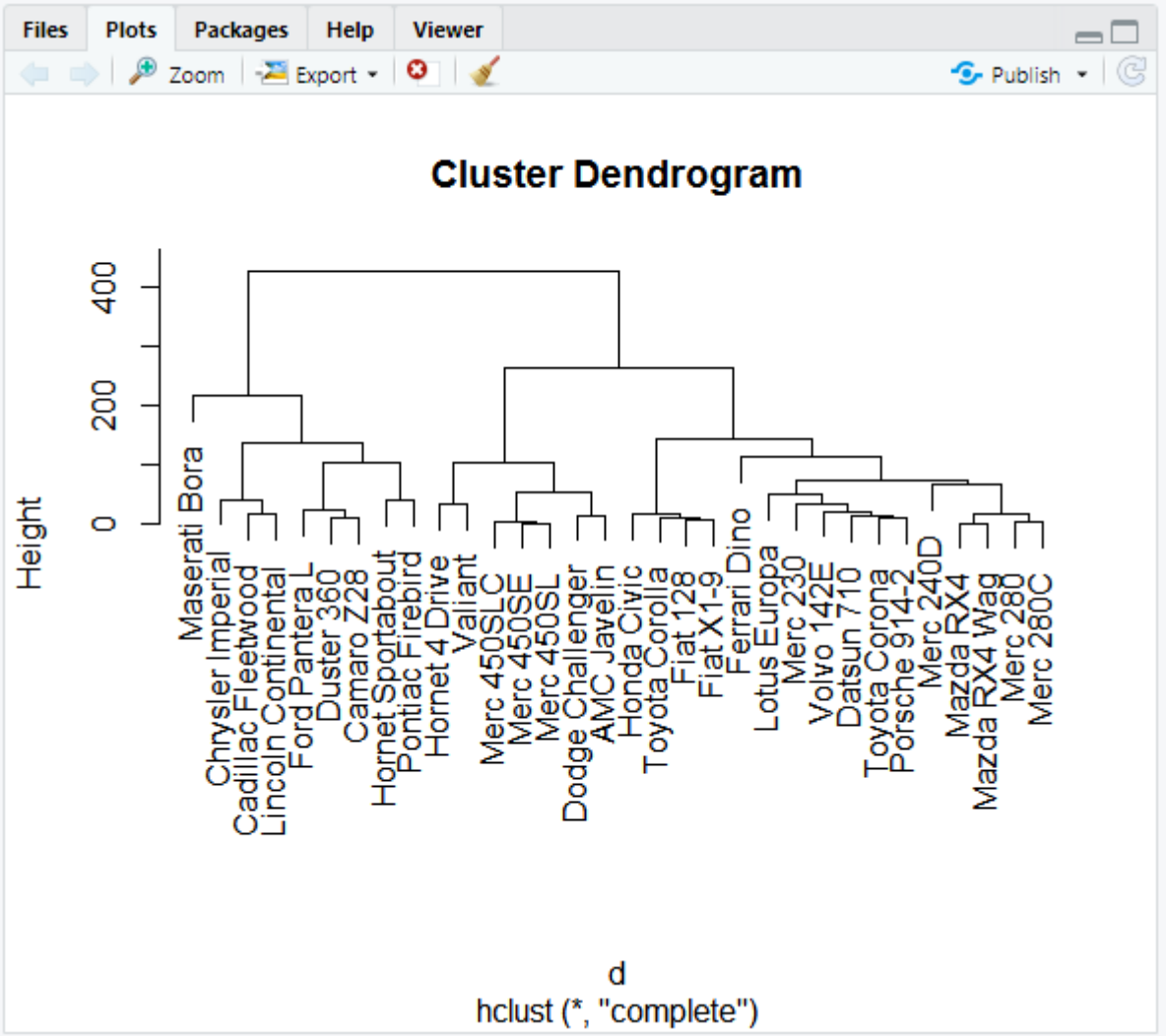


Hierarchical Clustering: R programming language provides a function hclust() for hierarchical clustering on a distance matrix. dist() function is used that generates the distance matrix.

> d <- dist(as.matrix(mtcars)) # find distance matrix

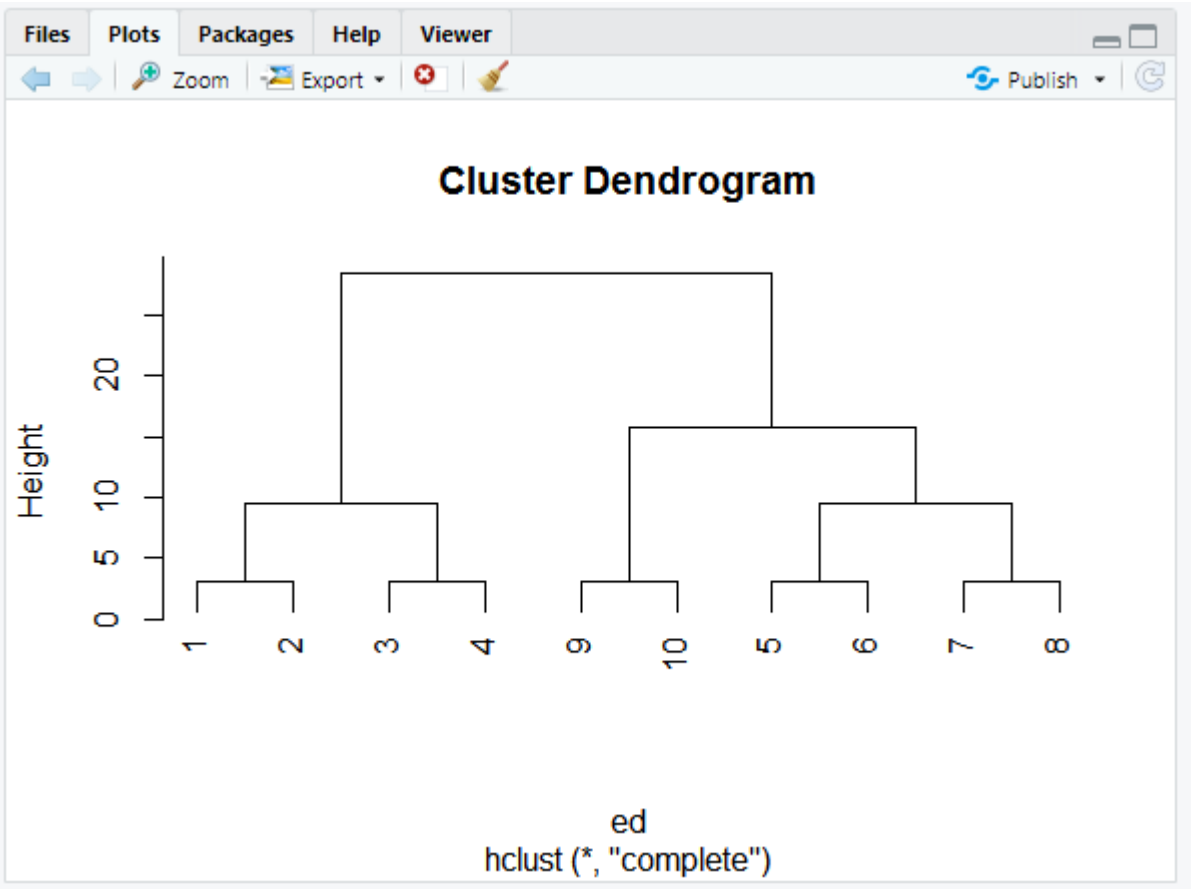
> hc <- hclust(d) # apply hierarchical clustering

> plot(hc) # plot the dendrogram




```
> mt<-Matrix(1:100,10,10)
Warning messages:
```

```
1: package 'RMySQL' was built under R version
3.6.1
2: package 'DBI' was built under R version 3.6.1
> mt
10 x 10 Matrix of class "dgeMatrix"
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
 [1,]    1   11   21   31   41   51   61   71
 [2,]    2   12   22   32   42   52   62   72
 [3,]    3   13   23   33   43   53   63   73
 [4,]    4   14   24   34   44   54   64   74
 [5,]    5   15   25   35   45   55   65   75
 [6,]    6   16   26   36   46   56   66   76
 [7,]    7   17   27   37   47   57   67   77
 [8,]    8   18   28   38   48   58   68   78
 [9,]    9   19   29   39   49   59   69   79
[10,]   10   20   30   40   50   60   70   80
      [,9] [,10]
 [1,]   81    91
 [2,]   82    92
 [3,]   83    93
 [4,]   84    94
 [5,]   85    95
 [6,]   86    96
 [7,]   87    97
 [8,]   88    98
 [9,]   89    99
[10,]   90   100

> ed<-dist(mt,method="eucledian")
Error in dist(mt, method = "eucledian") :
invalid distance method

> ed<-dist(mt,method="euclidean")

> h1<-hclust(ed)

> h1

Call:
hclust(d = ed)

Cluster method   : complete
Distance         : euclidean
Number of objects: 10

> plot(h1)

>
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Cluster Dendrogram
Height
0
5
10
20
1
2
3
4
9
10
5
6
7
8
ed
hclust (*, "complete")