PRACTICAL 4: Practical of Clustering

1) k-means clustering

```
> data("iris")
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> new_data<-subset(iris,select=c(-Species))
> new_data
    Sepal.Length Sepal.Width Petal.Length Petal.Width
1            5.1         3.5          1.4         0.2
2            4.9         3.0          1.4         0.2
3            4.7         3.2          1.3         0.2
4            4.6         3.1          1.5         0.2
5            5.0         3.6          1.4         0.2
6            5.4         3.9          1.7         0.4
7            4.6         3.4          1.4         0.3
8            5.0         3.4          1.5         0.2
9            4.4         2.9          1.4         0.2
10           4.9         3.1          1.5         0.1
11           5.4         3.7          1.5         0.2
12           4.8         3.4          1.6         0.2
13           4.8         3.0          1.4         0.1
14           4.3         3.0          1.1         0.1
15           5.8         4.0          1.2         0.2
16           5.7         4.4          1.5         0.4
17           5.4         3.9          1.3         0.4
18           5.1         3.5          1.4         0.3
19           5.7         3.8          1.7         0.3
20           5.1         3.8          1.5         0.3
21           5.4         3.4          1.7         0.2
22           5.1         3.7          1.5         0.4
23           4.6         3.6          1.0         0.2
24           5.1         3.3          1.7         0.5
25           4.8         3.4          1.9         0.2
26           5.0         3.0          1.6         0.2
27           5.0         3.4          1.6         0.4
28           5.2         3.5          1.5         0.2
29           5.2         3.4          1.4         0.2
30           4.7         3.2          1.6         0.2
31           4.8         3.1          1.6         0.2
32           5.4         3.4          1.5         0.4
33           5.2         4.1          1.5         0.1
34           5.5         4.2          1.4         0.2
35           4.9         3.1          1.5         0.2
36           5.0         3.2          1.2         0.2
37           5.5         3.5          1.3         0.2
38           4.9         3.6          1.4         0.1
39           4.4         3.0          1.3         0.2
40           5.1         3.4          1.5         0.2
41           5.0         3.5          1.3         0.3
42           4.5         2.3          1.3         0.3
43           4.4         3.2          1.3         0.2
44           5.0         3.5          1.6         0.6
45           5.1         3.8          1.9         0.4
46           4.8         3.0          1.4         0.3
```

- Here, we create a subset of the iris data in new_data where we remove Species variable as clustering can only be performed on numerical data and not categorical data and species is categorical data.

- We use kmeans function that takes as parameter the dataset and k i.e. the number of clusters we want to form. We start with 3 clusters. The clusters so formed have 50, 38 and 62 data points respectively.

- As we can see:
    - **cluster means** gives means for all the attributes for all three clusters.
    - **clustering vector** tells us in which cluster does every data point belong to among 150 rows.
    - **within cluster sum of squares by clusters gives** variance in all 3 clusters. Lesser the variance higher the efficiency of the cluster formed. 1^{st} cluster has lowest variance so efficiency is highest.

```
> cl<-kmeans(new_data,3)
> cl
K-means clustering with 3 clusters of sizes 50, 38, 62

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

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

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

Available components:

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

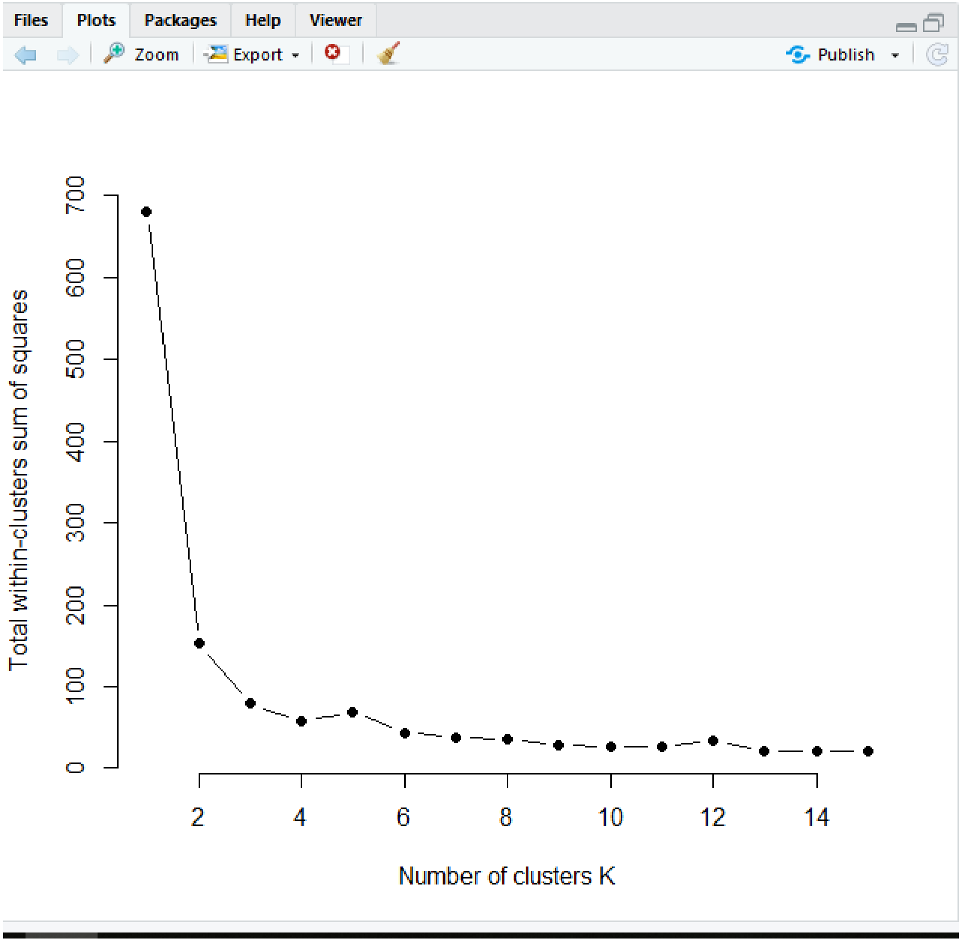
- We use sapply function that applies the inner function kmeans 15 times to the data set to form multiple clusters. wss contains a list of "total within sum of squares distance" i.e. $tot.withinss for every cluster.

```
> data<-new_data
>
> wss<-sapply(1:15,
+             function(k){kmeans(data,k)$tot.withinss})
> wss
 [1] 681.37060 152.34795  78.85144  57.26562  69.24240  43.68323  38.11226
 [8]  35.92851  27.78609  26.76674  26.35376  34.11821  21.55400  21.23139
[15]  20.59309
>
> plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab="Number of clusters K",
+      ylab="Total within-clusters sum of squares")
```

Then we plot with:

- no of clusers(1:15) in x axis
- wss in y axis
- type="b" means we want both lines and points.
- pch means plotting character symbol. Number 19 means solid circle.
- xlab and ylab are labels of x and y axes respectively.

Output of plot:



Analysis:
x axis shows number of clusters. Here there are 15 clusters 1:15 as we provided in the function. Within clusters sum of squares is represented on the y axis. We observe that as we move from 1^{st} to 2^{nd} cluster the wss drastically decreases and 2 onwards there is not much change. Hence we can chose 2 as the appropriate number of clusters.

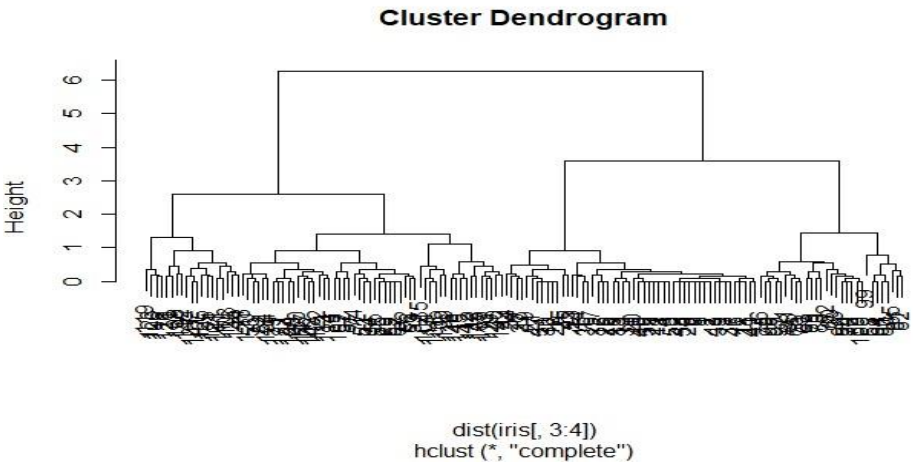
hierarchical clustering is of 3 type: single linkage, complete linkage and average linkage.

Q.2) Complete agglomerative clustering
in complete HAC we take max of distance between 2 data points for clustering

i) cluster

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

Output:-



Analysis: we form the clusters using hclust method. It shows a dendogram which is a graphical representation of hierarchical agglomerative clustering.

ii) clustercut

Output:-

```
> clustercut<-cutree(clusters,3)
> clustercut
  [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 1 1
 [43] 1 1 1 1 1 1 1 1 2 2 2 3 2 2 2 3 2 3 3 3 3 2 3 3 2 3 2 3 2 3 2 2 3 3 2 2 2 3 3 3 3 2
 [85] 2 2 2 3 3 3 3 2 3 3 3 3 3 3 3 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 2
[127] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Analysis: We cute the tree into 3 groups

iii) table(Create table of specified data set)

```
table(clustercut,iris$Species)
```

Output:-

```
> table(clustercut,iris$Species)
```

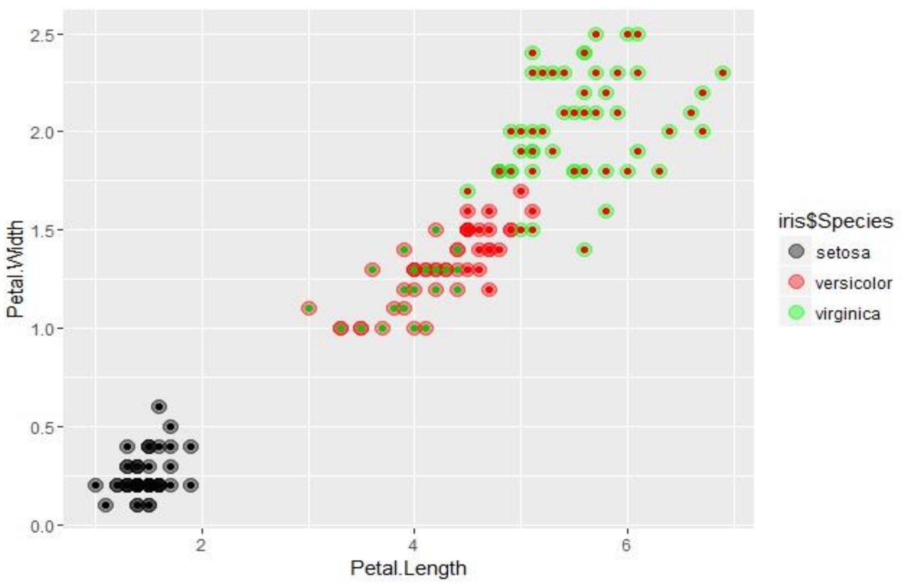
clustercut	setosa	versicolor	virginica
1	50	0	0
2	0	21	50
3	0	29	0

Analysis: We see that data points with setosa completely fall in cluster 1,with versicolor fall in 2 and 3 and virginica falls in cluster 2

iv)ggplot in average agglomerative cluster

```
library(ggplot2)
ggplot(iris,aes(Petal.Length,
Petal.Width,color=iris$Species))+geom_point(alpha=0.4,size=3.5
)+geom_p oint(col=clustercut)+scale_color_manual(values =
c('black','red','green'))
```

Output:-



Analysis: As we can see, the clustering in case of setosa is very clear and appropriate. Red representing versicolor is also clustered properly to some extent. However there is no distinct clustering for virginica (green).

Q.3) Average agglomerative clustering

in average agglomerative clustering we take average of data points while clustering

i) Cluster

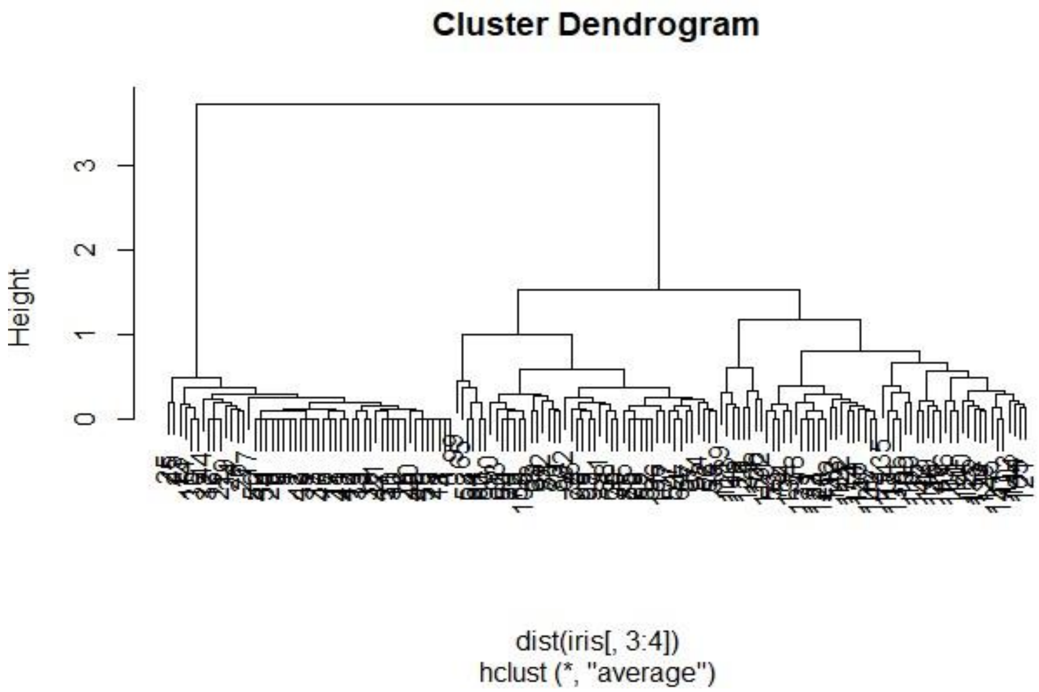
Output:-

```
> plot(clusters)
> clusters1<-hclust(dist(iris[,3:4]),method = 'average')
> clusters1

Call:
hclust(d = dist(iris[, 3:4]), method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 150
```

Output:



ii) Clustercut

Output:-

```
> clustercut<-cutree(clusters,3)
> clustercut
  [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 1 1
 [43] 1 1 1 1 1 1 1 1 2 2 2 3 2 2 2 3 2 3 3 3 3 2 3 3 2 3 2 3 2 3 2 3 2 2 3 3 2 2 2 3 3 3 3 2
 [85] 2 2 2 3 3 3 3 2 3 3 3 3 3 3 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 2 2
[127] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

iii) table(Create table of specify data set)

table(clustercut,iris$Species)

Output:-

```
> table(clustercut,iris$Species)

clustercut setosa versicolor virginica
         1     50          0         0
         2      0         21        50
         3      0         29         0
```

iv)ggplot in average agglomerative cluster

```
 library(ggplot2)
ggplot(iris,aes(Petal.Length,
Petal.Width,color=iris$Species))+geom_point(alpha=0.4,size=3.5)+geo
m_p oint(col=clustercut)+scale_color_manual(values =
c('black','red','green'))
```

Output:-