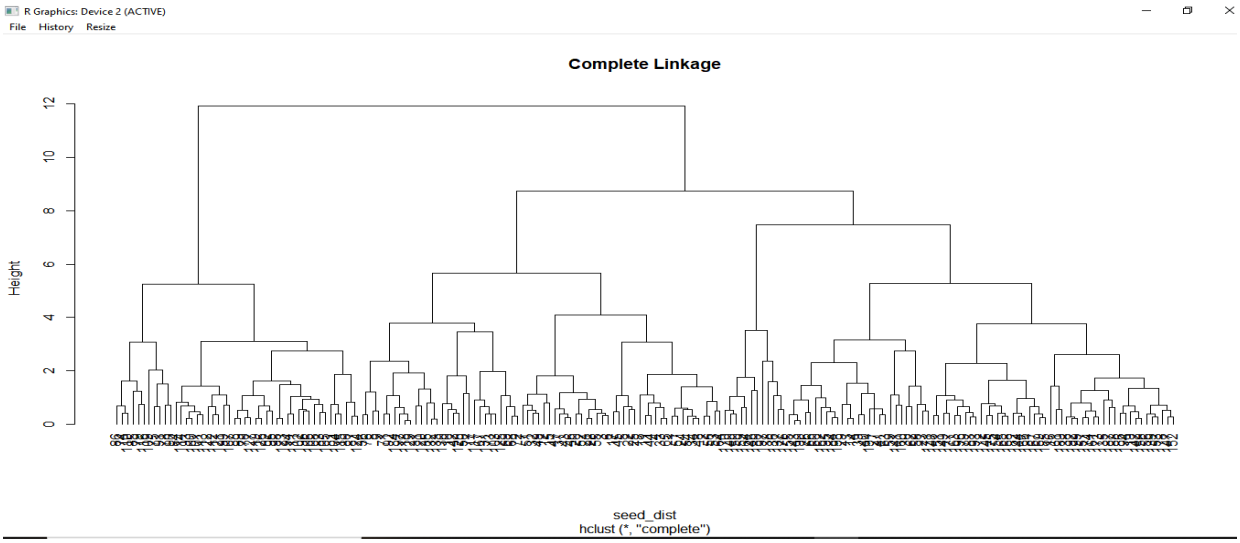
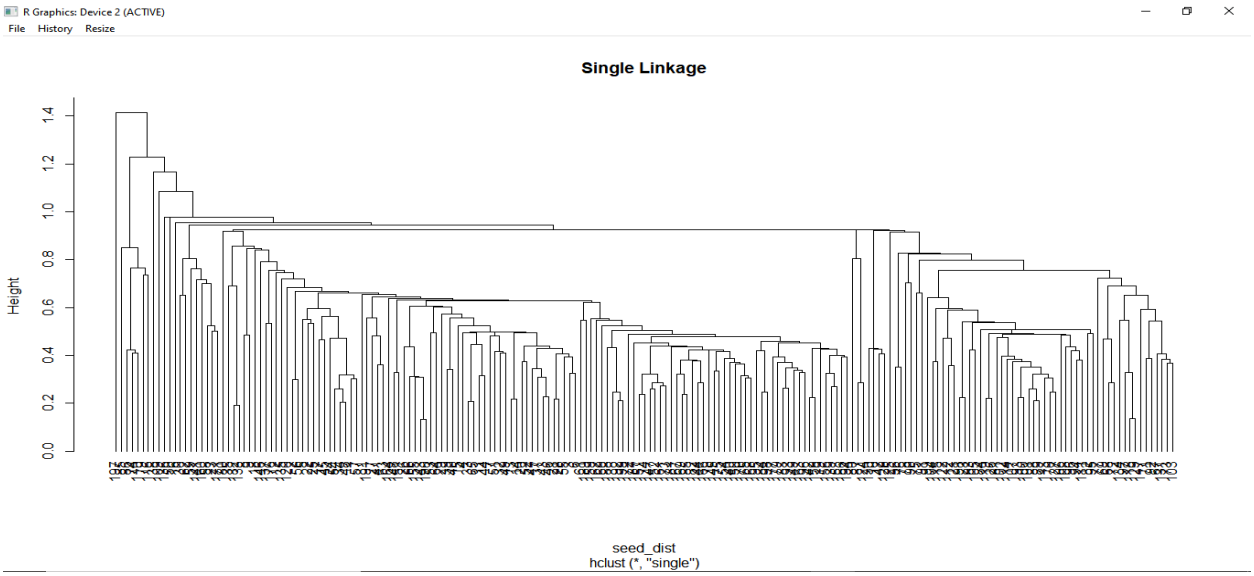
Question 3

(a) Complete linkage dendogram



Single Linkage Dendogram



Average Linkage Dendogram

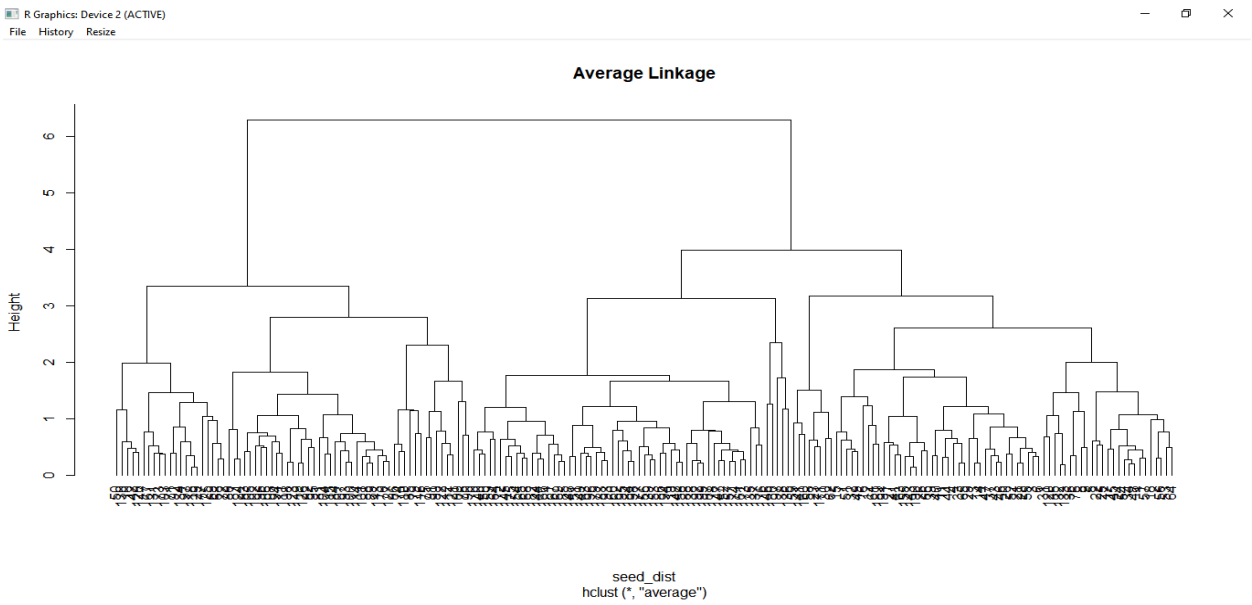


Table Plotting seed group and clusters for complete linkage with 3 clusters

```
> t_complete

complete_cut  A  B  C
           1 46 22  0
           2 20  0 65
           3  0 46  0
>
```

Table Plotting seed group and clusters for Single linkage with 3 clusters

```
> t_single

single_cut  A  B  C
          1 66 62 64
          2  0  6  0
          3  0  0  1
>
```

Table Plotting seed group and clusters for average linkage with 3 clusters

```
> t_average

average_cut  A  B  C
          1 60  4  8
          2  3 64  0
          3  3  0 57
>
```

- We can conclude from the table above that average linkage clustering is the best grouping method for clustering.

Calculating adjusted Rand indexes for the clustering methods, we get.

```
          3  3  0 57
> adj_complete = adj.rand.index(complete_cut,seeds$Seed.Group)
> adj_complete
[1] NaN
> adj_complete = adj.rand.index(complete_cut,as.numeric(seeds$Seed.Group))
> adj_complete
[1] 0.520021
> adj_single = adj.rand.index(complete_cut,as.numeric(seeds$Seed.Group))
> adj_single
[1] 0.520021
> adj_single = adj.rand.index(single_cut,as.numeric(seeds$Seed.Group))
> adj_single
[1] 0.001509422
> adj_average = adj.rand.index(average_cut,as.numeric(seeds$Seed.Group))
> adj_average
[1] 0.7482942
>
```

- From the rand index we can conclude that adjusted rand index of average linkage is greatest and adjusted rand index of single linkage is the worst
- Therefore , we can conclude that average linkage method performed the best and single linkage method performed the worst.

b) To find the cluster stability ,Using bootstrapping cluster stability we find that k=3

```
> k.select(seed, range = 2:6, B=50, r= 5, scheme_2=TRUE)
$profile
        2         3         4         5         6
0.9043121 0.8827857 0.6648202 0.4373951 0.4007511

$k
[1] 3

>
> k.select(seed, range = 2:6, B=50, r= 5, scheme_2=FALSE)
$profile
        2         3         4         5         6
0.8786189 0.8695701 0.6331820 0.3307543 0.3845723
```

Therefore for k means value of k=3.

```
> km = kmeans(seed,centers=3,nstart=20)
> km
K-means clustering with 3 clusters of sizes 67, 60, 72

Cluster means:
      Area Perimeter Compactness Length.Kernel Width.Kernel Asymmetry Length.Kernel.Grove
1 14.65731  14.47284   0.8782030      5.573627     3.275657  2.662525             5.192836
2 18.71967  16.29950   0.8847450      6.209883     3.721283  3.616267             6.063867
3 11.99458  13.29056   0.8523194      5.235569     2.876319  4.733042             5.096639

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 3 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 2 1 3 1 1 1 1 1 1 1 1 1
 [49] 1 1 1 1 1 1 1 1 1 1 3 3 3 1 1 1 1 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
 [97] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 1 1 2 1 1 1 3 3 3 3 3 3 3 3 3 3
[145] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 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 3 3 3 3
[193] 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 195.2309 183.4582 168.4169
 (between_SS / total_SS =  78.7 %)

Available components:

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

Gap statistics on K-means

```
> gap_kmeans
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = seed, FUNcluster = kmeans, K.max = 10, B = 100, nstart = 20)
B=100 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstSEmax', SE.factor=1): 3
          logW   E.logW       gap     SE.sim
 [1,] 5.391318 5.535044 0.1437255 0.02225016
 [2,] 4.909119 5.163324 0.2542055 0.02213710
 [3,] 4.638097 4.994389 0.3562918 0.02115669
 [4,] 4.519943 4.841904 0.3219602 0.02215943
 [5,] 4.420955 4.713757 0.2928024 0.02028086
 [6,] 4.335164 4.611284 0.2761197 0.01872010
 [7,] 4.271944 4.535915 0.2639704 0.01763565
 [8,] 4.196353 4.468693 0.2723396 0.01844727
 [9,] 4.141266 4.408379 0.2671127 0.01891271
[10,] 4.096168 4.352796 0.2566276 0.01875928
>
```

For K=3 , we have

```
> table(km$cluster,seeds$Seed.Group)

    A  B  C
  1 57  9  1
  2  8  0 64
  3  1 59  0
>
```

Adjusted rand index for

```
> adj_kmeans <- adj.rand.index(km$cluster,as.numeric(seeds$Seed.Group))
> adj_kmeans
[1] 0.7402708
>
```

Adjusted rand index for Average Linkage = 0.748294

Adjusted rand index for K-means = 0.7402708

Since Adjusted rand Index of Average linkage is greater than K-means . Therefore, performance of Average linkage is better for this data.