M4_L2_RomilShah

Romil Shah

June 13, 2016

Read Data and additional packages

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
require(cluster)
```

```
## Loading required package: cluster
```

```
require(useful)
```

```
## Loading required package: useful
```

```
## Warning: package 'useful' was built under R version 3.2.4
```

```
require(energy)
```

```
## Loading required package: energy
```

```
## Warning: package 'energy' was built under R version 3.2.5
```

```
dataframe <-
read.table("C:/Users/rams1/Desktop/DSCS6030/Module_04/ecoli.data")
colnames(dataframe) <-
c("SeqNames","mcg","gvh","lip","chg","aac","alm1","alm2","Class")
ecoli <- dataframe[2:8]
head(ecoli)
```

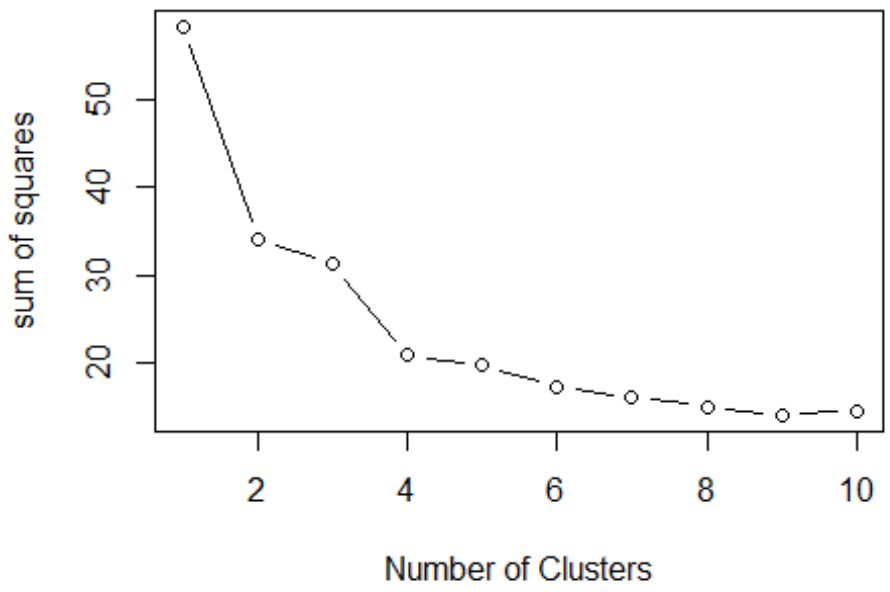
```
##    mcg  gvh  lip chg  aac alm1 alm2
## 1 0.49 0.29 0.48 0.5 0.56 0.24 0.35
## 2 0.07 0.40 0.48 0.5 0.54 0.35 0.44
## 3 0.56 0.40 0.48 0.5 0.49 0.37 0.46
## 4 0.59 0.49 0.48 0.5 0.52 0.45 0.36
## 5 0.23 0.32 0.48 0.5 0.55 0.25 0.35
## 6 0.67 0.39 0.48 0.5 0.36 0.38 0.46
```

Clustering

(1) K-means Clustering:

```
# Determining number of clusters
sos <- (nrow(ecoli)-1)*sum(apply(ecoli,2,var))
```

```r
for (i in 2:10) sos[i] <- sum(kmeans(ecoli, centers=i)$withinss)
plot(1:10, sos, type="b", xlab="Number of Clusters", ylab="sum of squares")
```




```r
# Hartigans's rule  FitKMean (similarity)
# require(useful)
best<-FitKMeans(ecoli,max.clusters=10, seed=111)
PlotHartigan(best)
```

Clustering by k = 4

```
k<-4
ecoli.4.clust<-kmeans(ecoli,k)
ecoli.4.clust
```

```
## K-means clustering with 4 clusters of sizes 64, 43, 82, 147
##
## Cluster means:
##         mcg       gvh       lip       chg       aac      alm1      alm2
## 1 0.7087500 0.4826563 0.4962500 0.5000000 0.5670313 0.7632812 0.7703125
## 2 0.3804651 0.4865116 0.4920930 0.5000000 0.5541860 0.7637209 0.7709302
## 3 0.6686585 0.6835366 0.5243902 0.5060976 0.5100000 0.4742683 0.3363415
## 4 0.3501361 0.4091156 0.4800000 0.5000000 0.4494558 0.3229932 0.3937415
##
## Clustering vector:
##   [1] 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [36] 4 4 3 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4
##  [71] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 3 4 4 4 4 4
## [106] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [141] 4 4 4 2 4 1 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 1 2 2 1 2 2 1 2 2 1 2 2
## [176] 2 1 1 1 2 1 1 1 4 1 2 1 1 2 1 2 1 2 2 1 1 2 2 1 2 1 2 2 2 2 1 2 2 2
```

```
1
## [211] 1 2 3 4 4 4 4 3 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
## [246] 2 1 1 1 1 1 2 1 1 1 1 1 4 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3
## [281] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3
1
## [316] 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 4 3 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 2.830589 2.405093 7.493306 7.768464
##  (between_SS / total_SS =  64.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

(2) PAM Clustering:

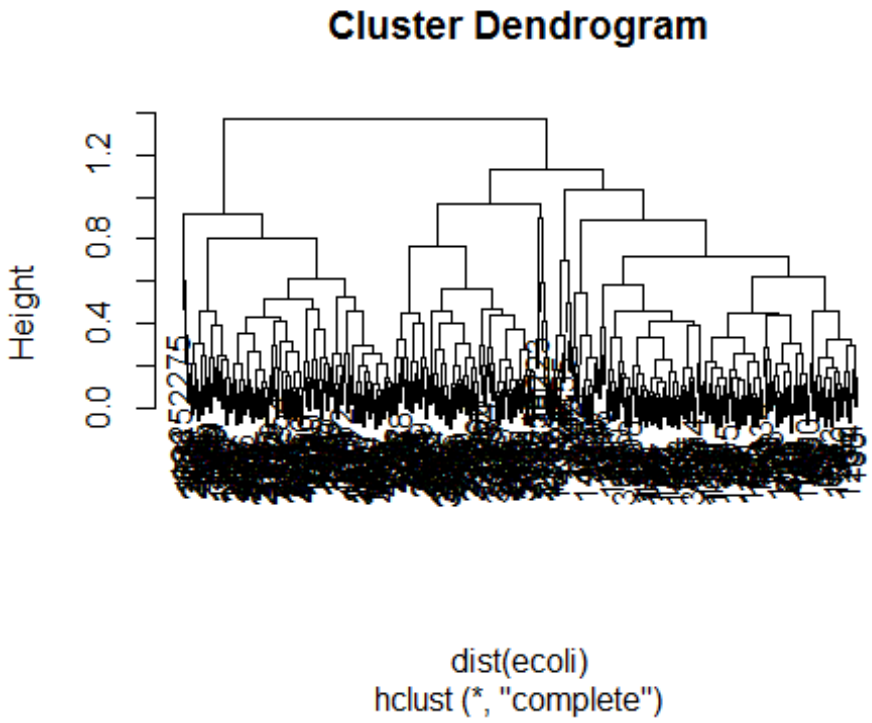
```
k<-4
ecoli.pam.4.clust<- pam(ecoli,k, keep.diss = TRUE, keep.data = TRUE)
ecoli.pam.4.clust
```

```
## Medoids:
##       ID  mcg  gvh  lip chg  aac alm1 alm2
## [1,] 127 0.28 0.33 0.48 0.5 0.45 0.22 0.33
## [2,] 130 0.37 0.44 0.48 0.5 0.42 0.39 0.47
## [3,] 291 0.67 0.70 0.48 0.5 0.46 0.45 0.33
## [4,] 222 0.63 0.49 0.48 0.5 0.54 0.76 0.79
## Clustering vector:
##   [1] 1 1 2 3 1 2 1 1 2 1 1 1 1 2 1 2 1 2 1 2 1 1 2 1 1 2 2 2 2 2 2 2 2
1
##  [36] 2 2 3 1 1 2 2 2 2 2 2 2 2 1 1 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 1 2 1
2
##  [71] 1 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 3 2 3 3 2 1 2 1 1
1
## [106] 2 2 2 2 1 1 2 1 1 1 1 2 1 2 2 1 1 1 1 2 2 1 2 1 2 2 2 1 1 2 2 1 1 1
1
## [141] 2 2 1 2 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4
## [176] 4 4 4 4 4 4 4 4 2 4 4 4 4 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4
4
## [211] 4 4 3 2 2 2 2 3 4 4 3 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4
## [246] 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3
## [281] 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
4
```

```
## [316] 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 2 3 3 3 3 3
## Objective function:
##     build      swap
## 0.2346583 0.2281333
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

(3) Hierarchical Clustering:

```
ecoli.h.clust<- hclust(d=dist(ecoli))
plot(ecoli.h.clust)
```



Answers:

A(1)

'k' for k-means was chosen based upon the hartigan's rule where the least sum of square was for k = 4.

A(2)

Cluster apporaches compare on the same data as follows: 1. The number of clusters i.e. 4 2. The further evaluation is done on basis of the confusion matrix 3. Cluster size and centers:

```
#size of cluster
ecoli.4.clust$size
```

```
## [1]  64  43  82 147
```

```
#centers of cluster
ecoli.4.clust$centers
```

```
##         mcg       gvh       lip       chg       aac      alm1      alm2
## 1 0.7087500 0.4826563 0.4962500 0.5000000 0.5670313 0.7632812 0.7703125
## 2 0.3804651 0.4865116 0.4920930 0.5000000 0.5541860 0.7637209 0.7709302
## 3 0.6686585 0.6835366 0.5243902 0.5060976 0.5100000 0.4742683 0.3363415
## 4 0.3501361 0.4091156 0.4800000 0.5000000 0.4494558 0.3229932 0.3937415
```

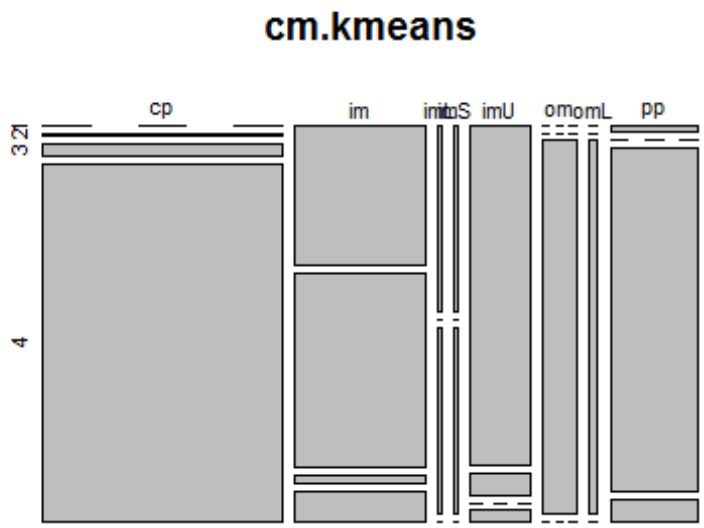
A(3)

1. Confusion matrix for k-means

```
cm.kmeans <-table(dataframe$Class,ecoli.4.clust$cluster)
cm.kmeans
```

```
##
##         1   2   3   4
##   cp    0   1   5 137
##   im   29  40   2   6
##   imL   1   0   1   0
##   imS   1   0   1   0
##   imU  32   2   0   1
##   om    0   0  20   0
##   omL   0   0   5   0
##   pp    1   0  48   3
```

```
plot(cm.kmeans)
```

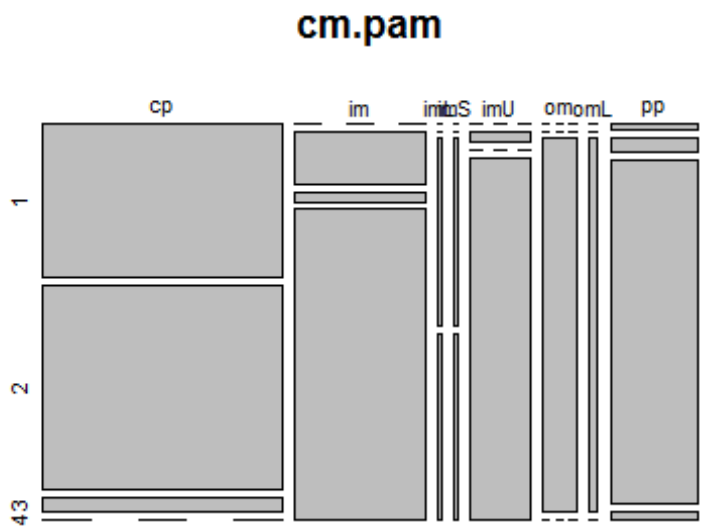


2. Confusion matrix for PAM

```
cm.pam<-table(dataframe$Class,ecoli.pam.4.clust$cluster)
cm.pam
```

```
##
##        1  2  3  4
##   cp  59 79  5  0
##   im   0 11  2 64
##   imL  0  0  1  1
##   imS  0  0  1  1
##   imU  0  1  0 34
##   om   0  0 20  0
##   omL  0  0  5  0
##   pp   1  2 48  1
```

```
plot(cm.pam)
```

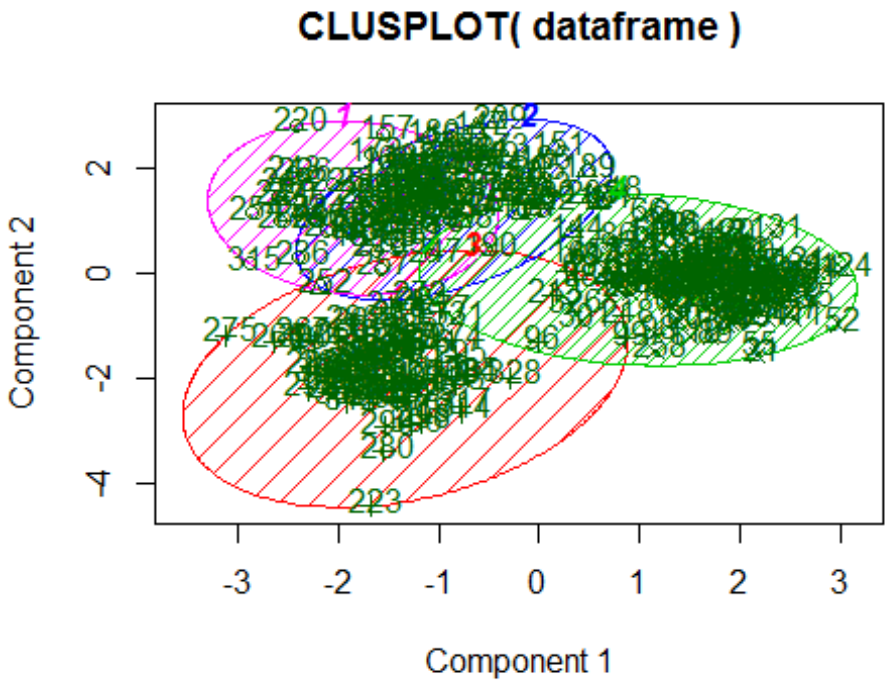


It is clearly seen that there are errors for 'cp', 'im' and 'imU' classes with respect to k-means and pam confusion plot. Other classes are quite similar and thus the approach matches in those classes.

A(4)

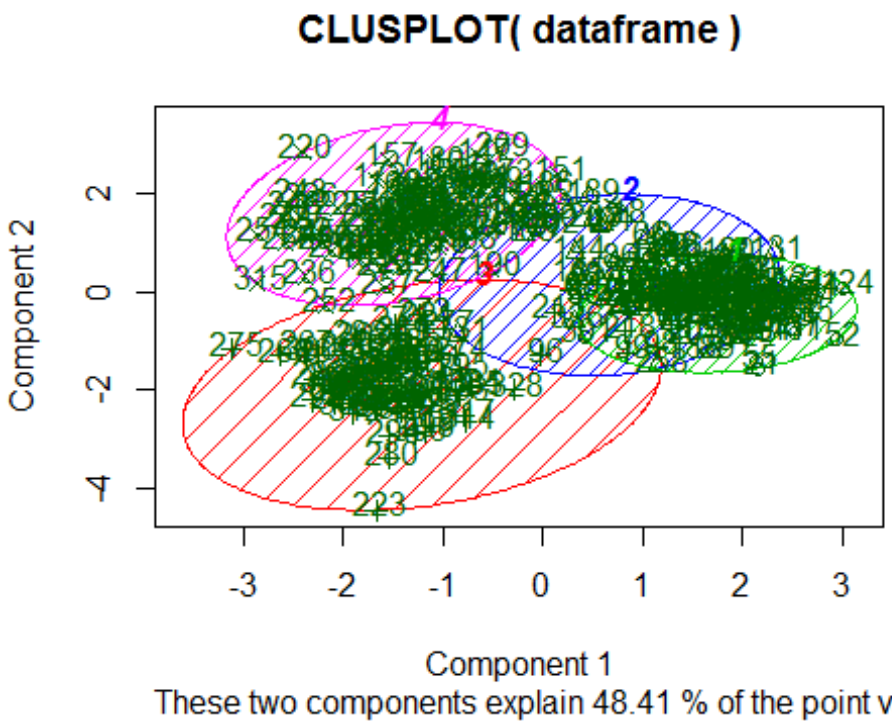
1.Centroid plots for k-means

```
clusplot(dataframe, ecoli.4.clust$cluster, color=TRUE, shade=TRUE, labels=2,
lines=0)
```



2. Centroid plots for PAM

```
clusplot(dataframe, ecoli.pam.4.clust$cluster, color=TRUE, shade=TRUE,
labels=2, lines=0)
```



From the centroid plots, it is seen that 'pink', 'red' and 'green' clusters have nearly similar centers but 'blue' has a different center. From that, 'pink' and 'red' also have nearly similar size and orientation compared to the other two clusters. Thus PAM and k-means have very similar approach in some sections of the data and not similar in other sections.

A(5)

Silhoutte plot for PAM

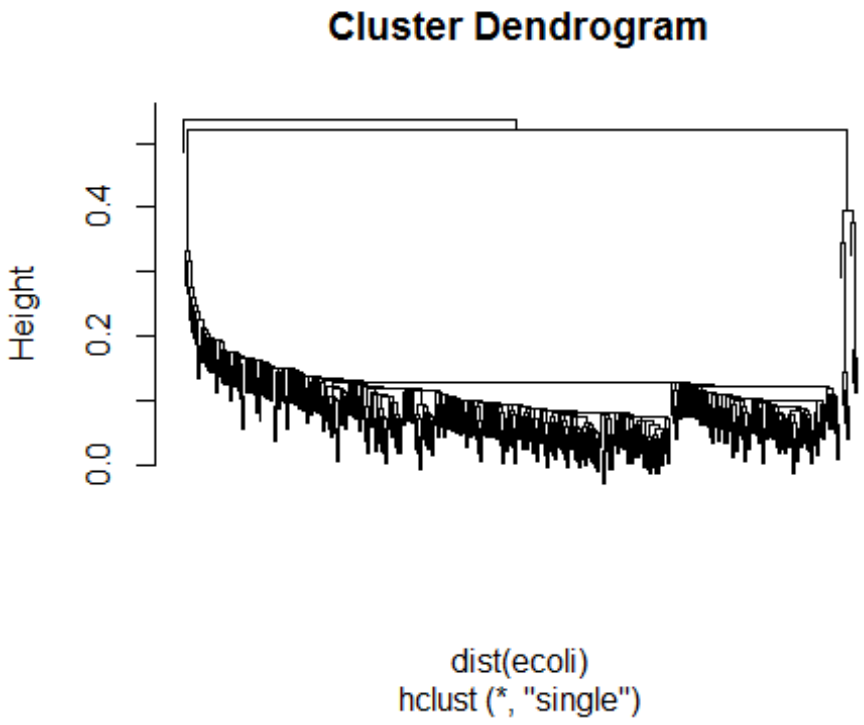
```
plot(ecoli.pam.4.clust, which.plots = 3)
```

The similarity between the for each cluster within every cluster is quite less for all the 4 clusters. Cluster 1 has about 0.32, Cluster 2 has about 0.11, Cluster 3 has about 0.28 and CLuster 4 has the highest among them at 0.46

A(6)

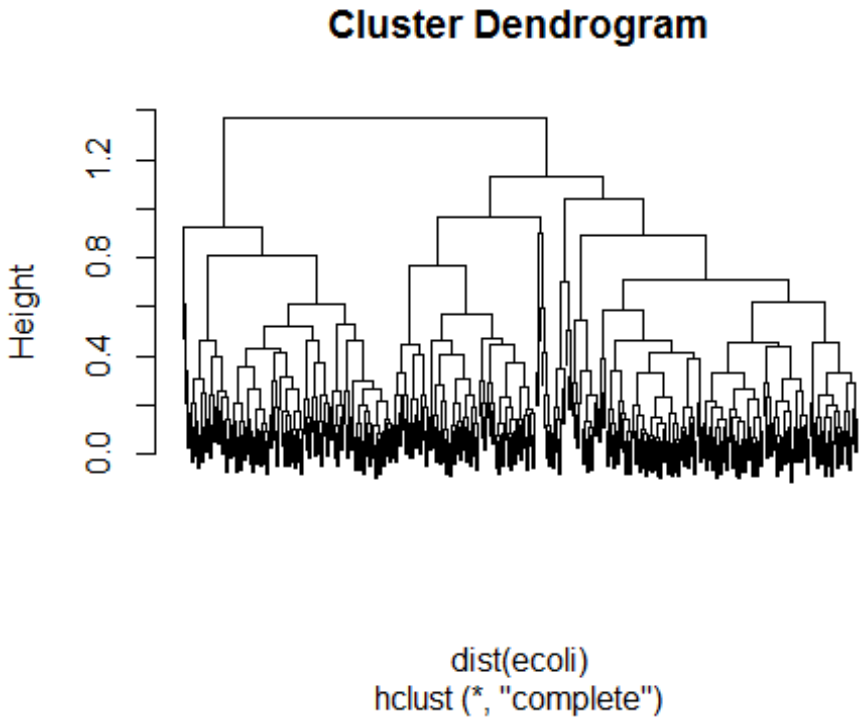
Hierarchical Clustering methods 1. Single Link

```
ecoli.h.clust.single<- hclust(dist(ecoli), method = "single")
plot(ecoli.h.clust.single, labels = FALSE)
```



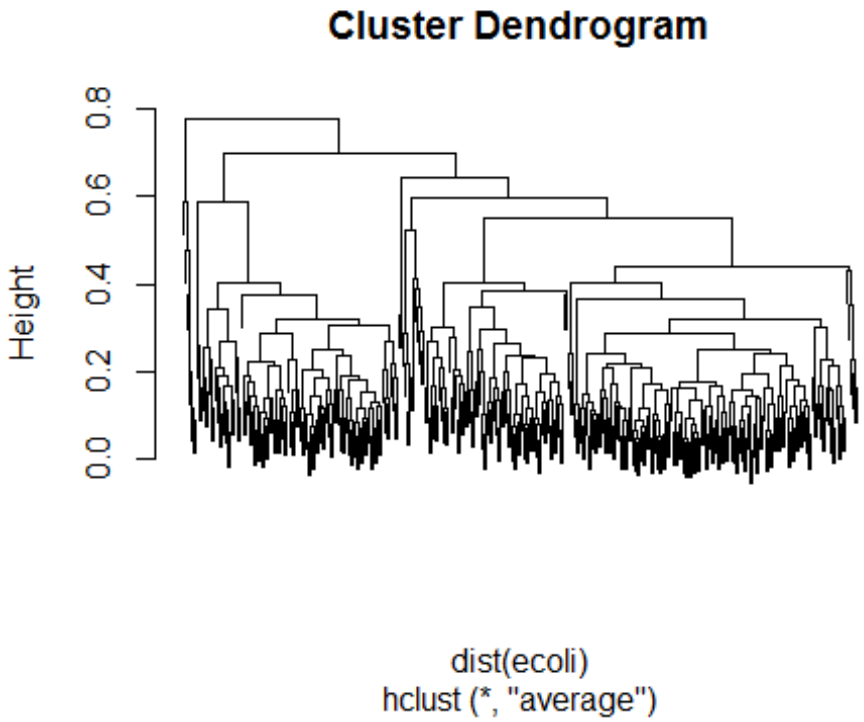
2. Complete Link

```
ecoli.h.clust.complete<- hclust(dist(ecoli), method = "complete")
plot(ecoli.h.clust.complete, labels = FALSE)
```



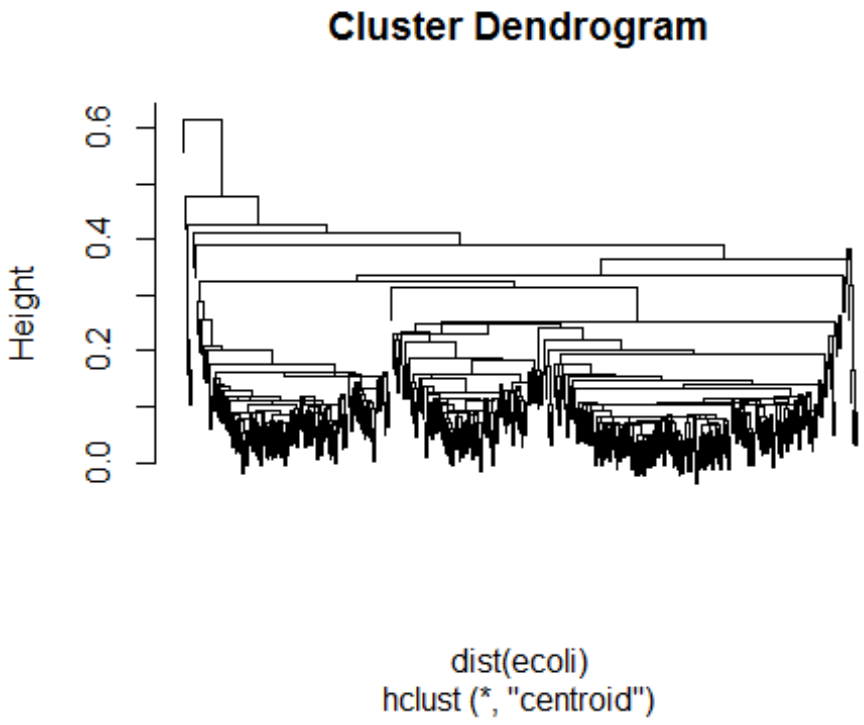
3. Average Link

```
ecoli.h.clust.average<- hclust(dist(ecoli), method = "average")
plot(ecoli.h.clust.average, labels = FALSE)
```



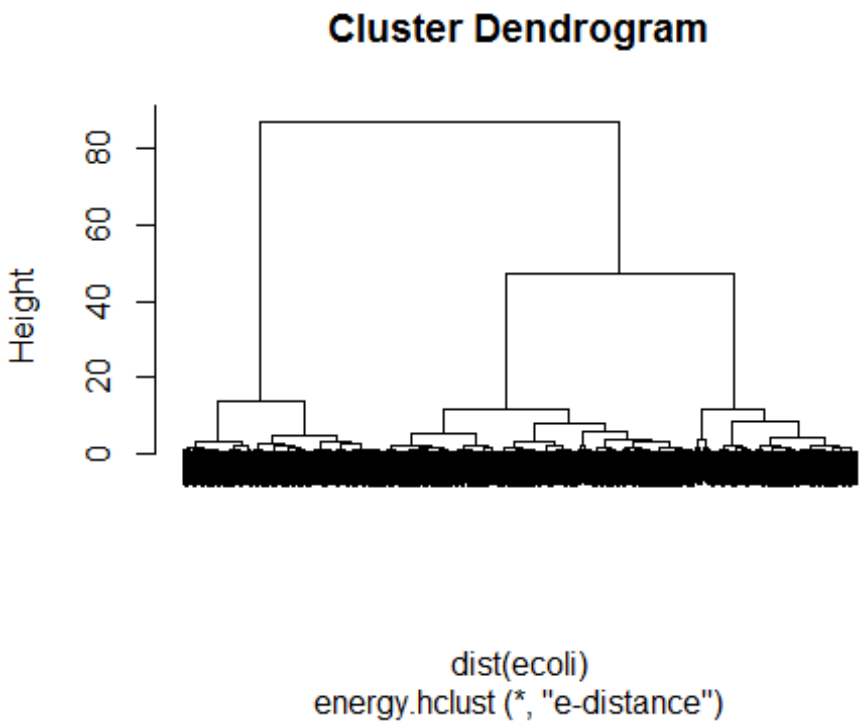
4. Centroid

```
ecoli.h.clust.centroid<- hclust(dist(ecoli), method = "centroid")
plot(ecoli.h.clust.centroid, labels = FALSE)
```



5. Minimum Energy

```
plot(energy.hclust(dist(ecoli)),labels = FALSE)
```



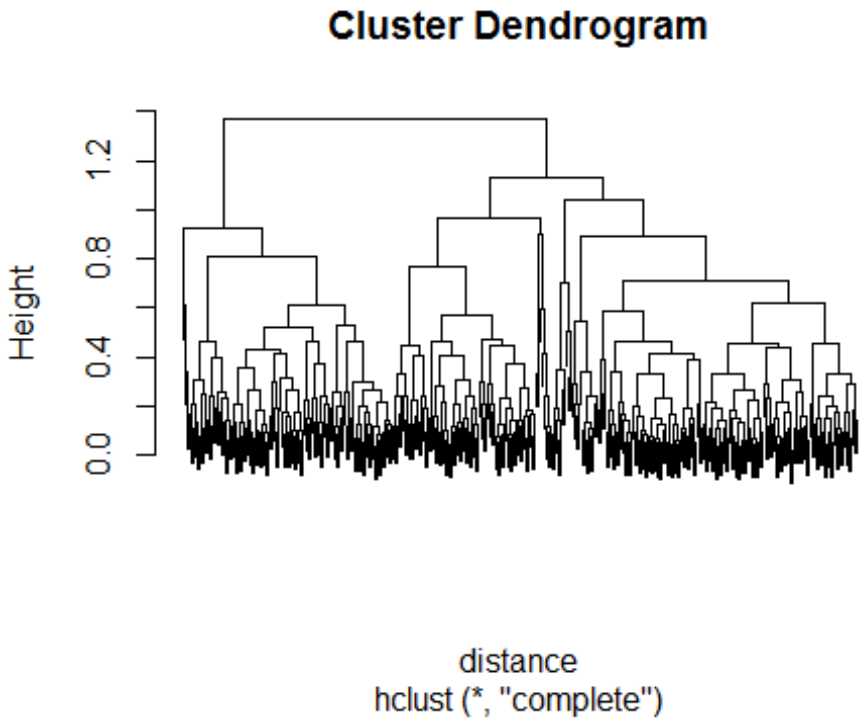
Complete and Average have similar type of clustering. Single has less heirarchical structure. Centroid is

more heirarchical towards the end. Energy clustering shows the most structured clustering along with high clustering towards the end.

A(7)

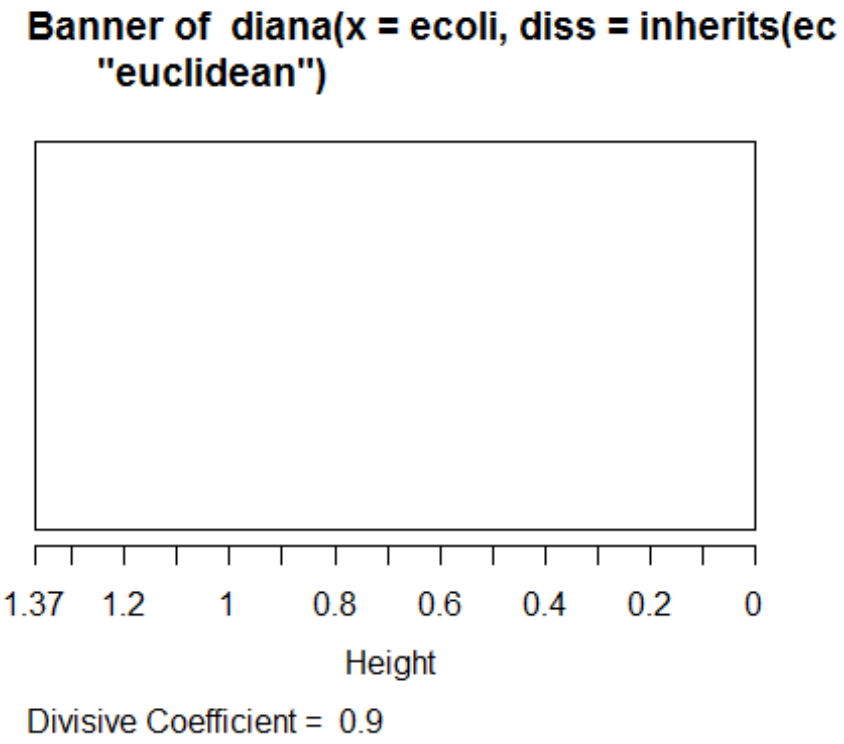
Hierarchical Clustering methods 1. Agglomerative

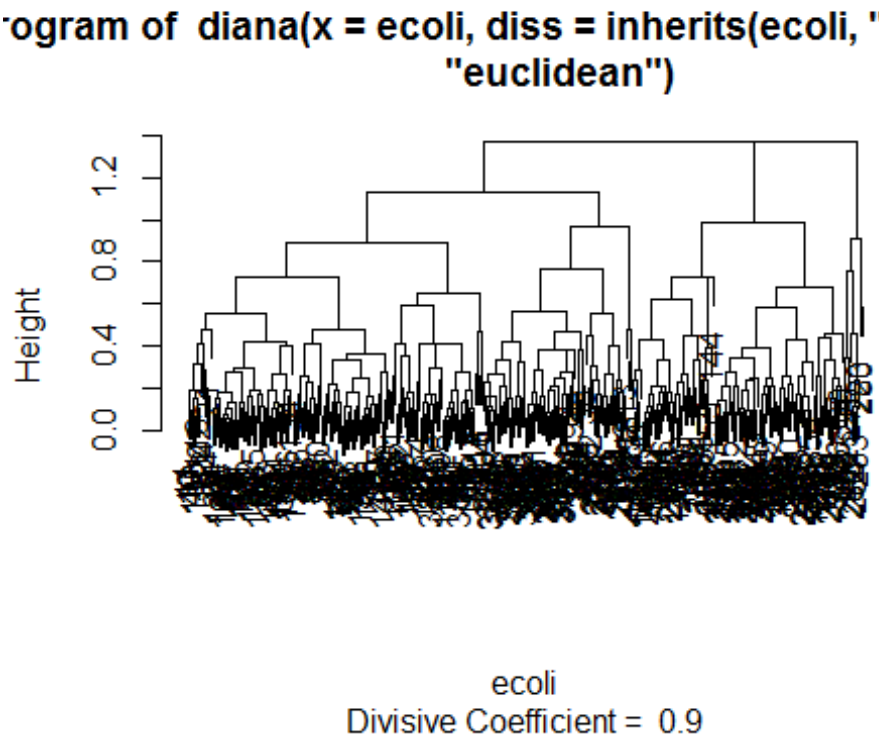
```
distance<- dist(ecoli,method="euclidean")
ecoli_agglo<-hclust(distance, method="complete")
plot(ecoli_agglo,labels=FALSE)
```



2. Divisive

```
ecoli_divi<-diana(ecoli, diss=inherits(ecoli, "dist"), metric="euclidean")
plot(ecoli_divi)
```



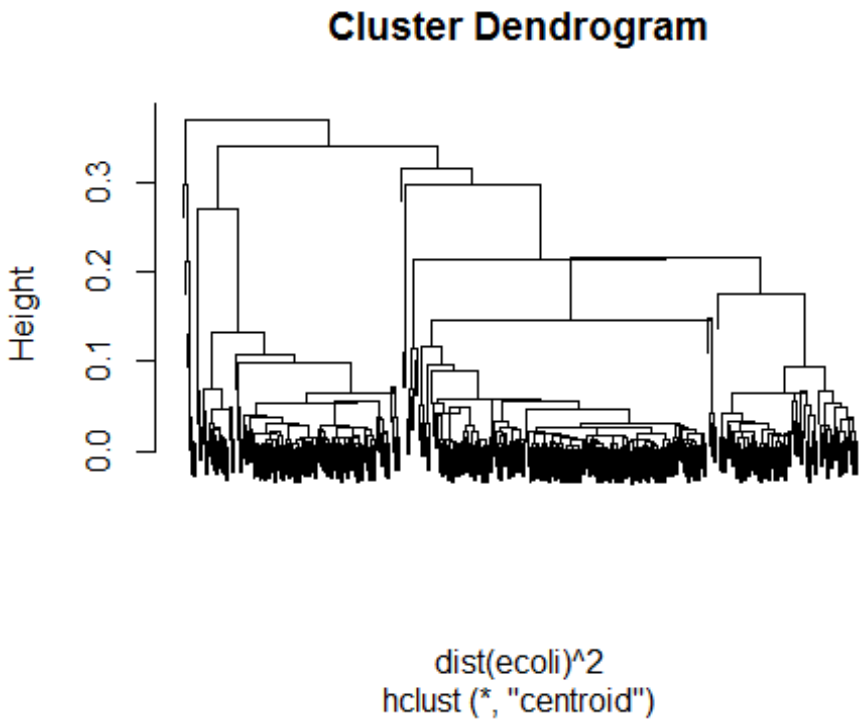


The cluster dendogram is quite similar for agglomerative and divisive clustering. The divisive clustering also gives a banner but as the correlation for in-between cluster data is less, the banner does not show lines.

A(8)

Hierarchical Clustering methods Centroid clustering with squared euclidean distance

```
ecoli_c_e<- hclust(dist(ecoli)^2, "centroid")
plot(ecoli_c_e,labels=FALSE)
```



The resulting dendogram is very much similar to agglomerative, centroid and minimum energy clustering.