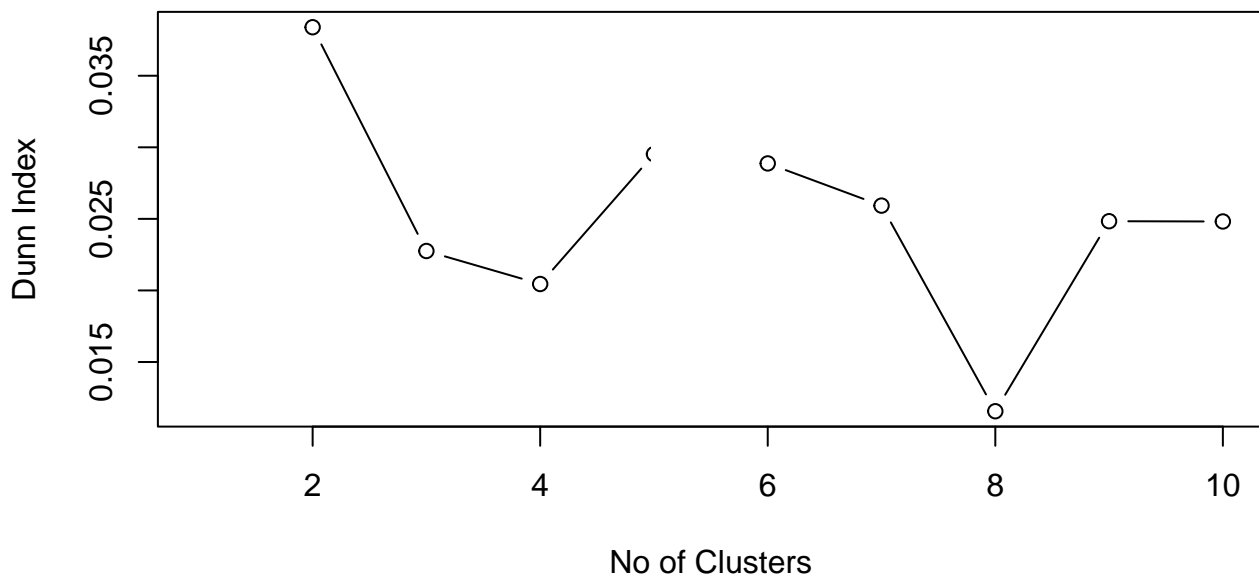


```
mydata=read.csv("C:\\Users\\zeeda\\Downloads\\anuran+calls+mfccs\\Frogs_MFCCs.csv")
data=mydata[,-c(23:26)]
```

i. *k*-means algorithm

```
dun_index=c()
for(i in 2:10){
  obj=kmeans(x=data,i)
  dun_index[i]=dunn(dist(data),as.vector(obj$cluster))
}
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```



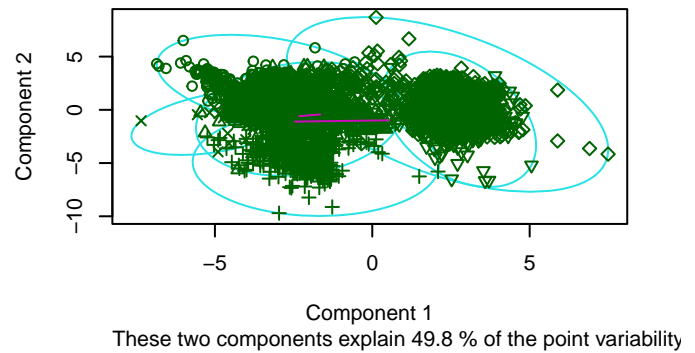
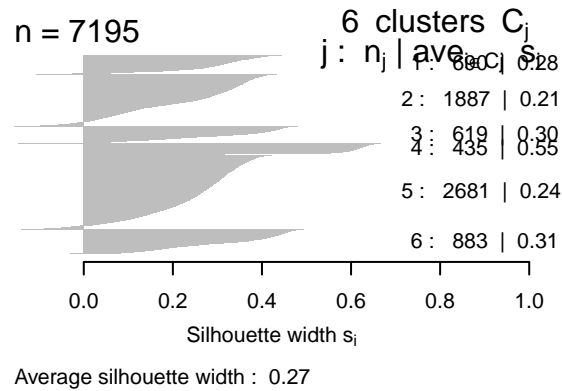
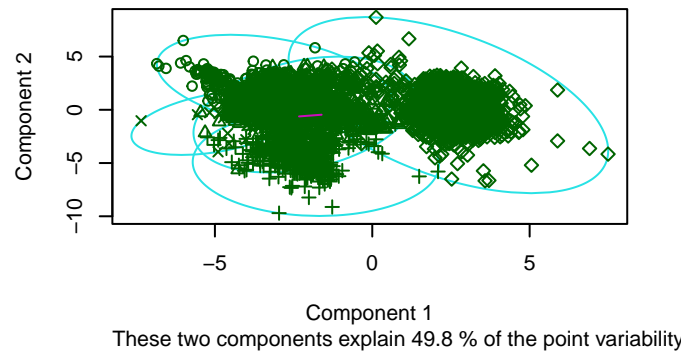
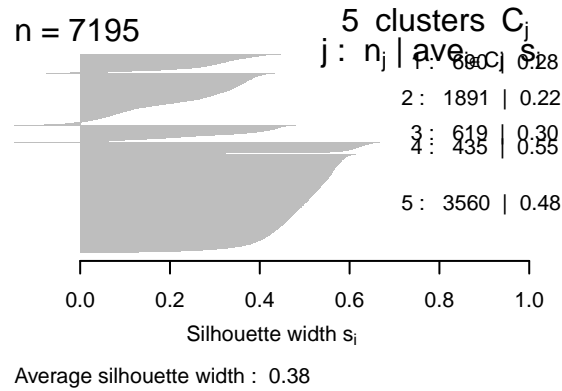
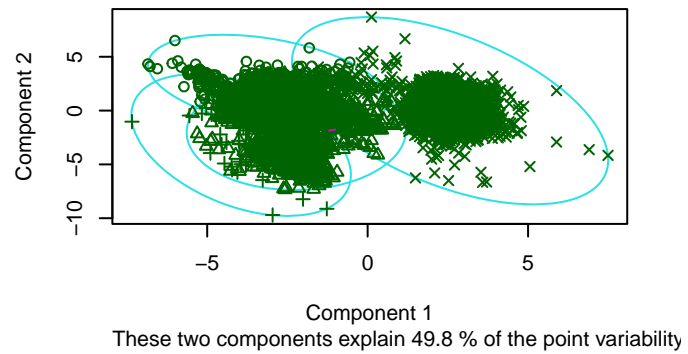
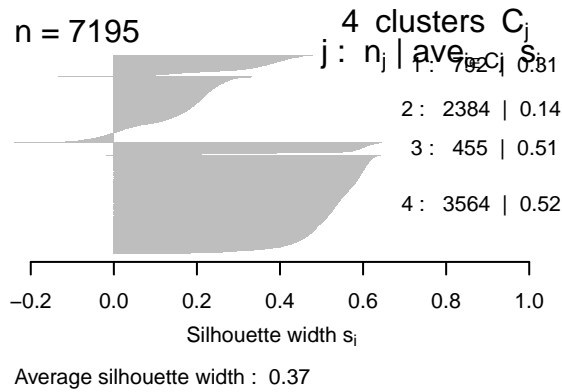
```
which.max(dun_index)
[1] 2
```

- From plot of Dunn Index the suitable no. of Cluster is 2 as its the least no of clusters that maximises the index.

ii. *k*-medoids algorithm

```
par(mfrow=c(3,2))
obj2=pam(data,4)
plot(obj2,which=2,main="")
plot(obj2,which=1,main="")
obj2=pam(data,5)
plot(obj2,which=2,main="")
plot(obj2,which=1,main="")
```

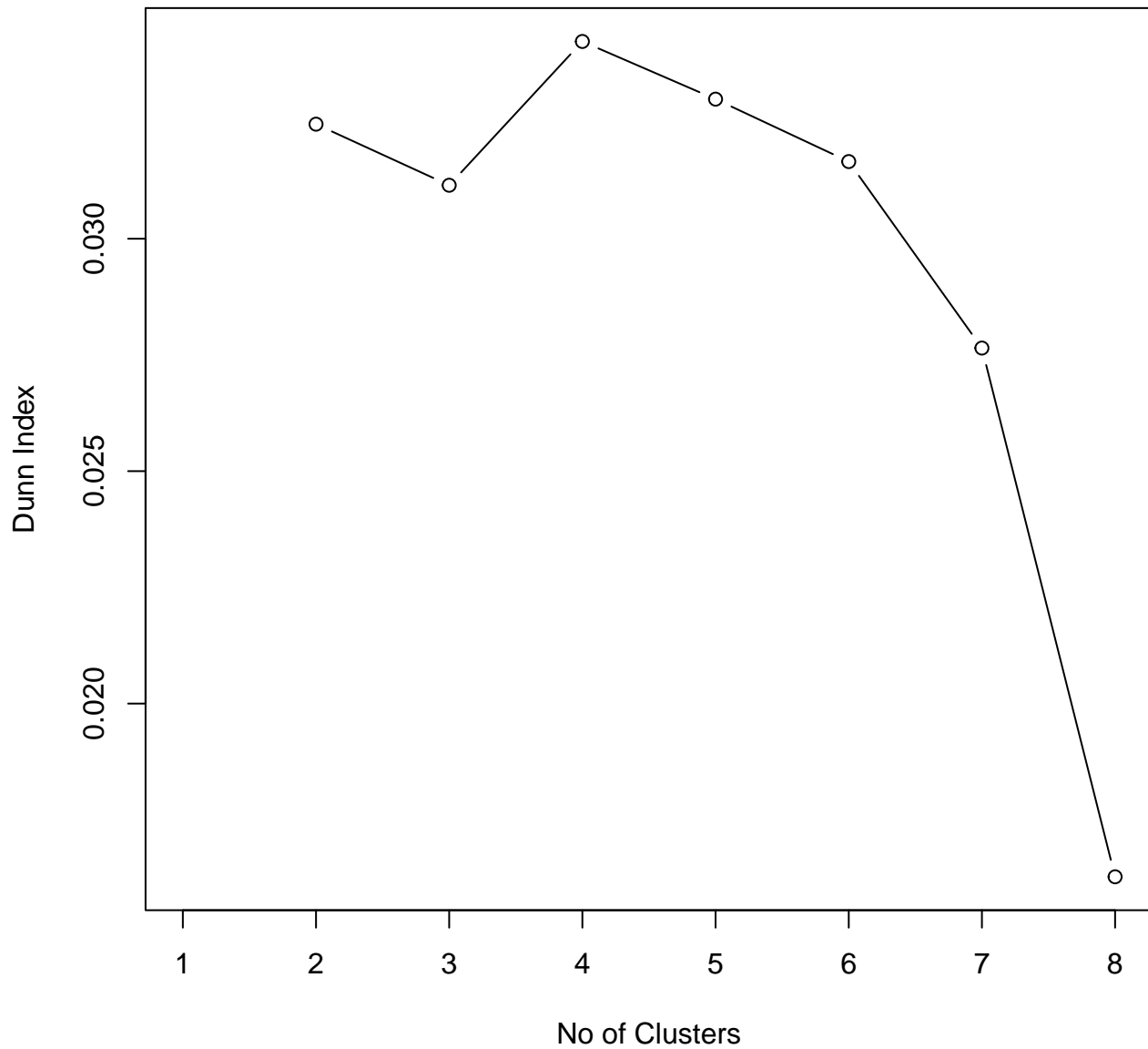
```
obj2=pam(data,6)
plot(obj2,which=2,main="")
plot(obj2,which=1,main="")
```



```
dun_index=c()
for(i in 2:8){
  obj2=pam(data,i)
  dun_index[i]=dunn(dist(data),obj2$clustering) }
which.max(dun_index)
```

```
[1] 4
```

```
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```

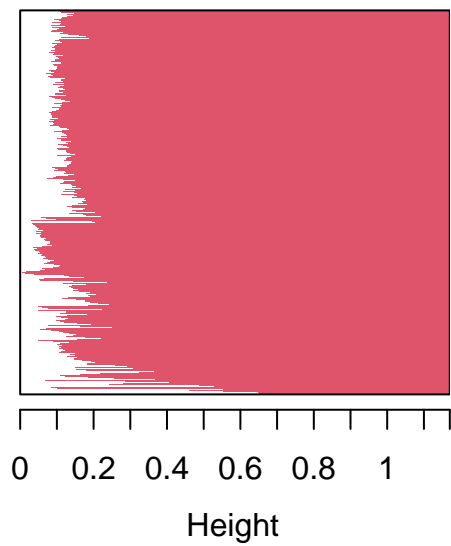


- From plot of Dunn Index the suitable no. of Cluster is 4 as its the least no of clusters that maximises the index.

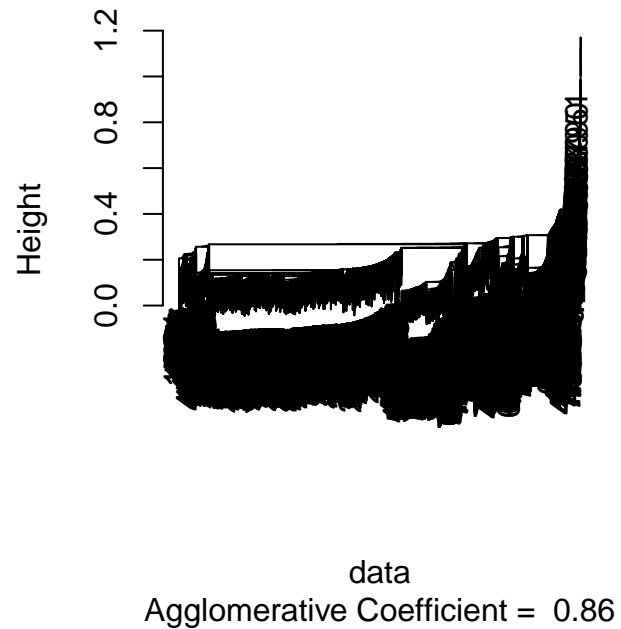
iii. single linkage algorithm

```
par(mfrow=c(1,2))  
obj3=agnes(data,method="single")  
plot(obj3,which=1,main="Banner plot of single linkage")  
plot(obj3,which=2,main="Dendogram of single linkage")
```

**Banner plot of single linkage**



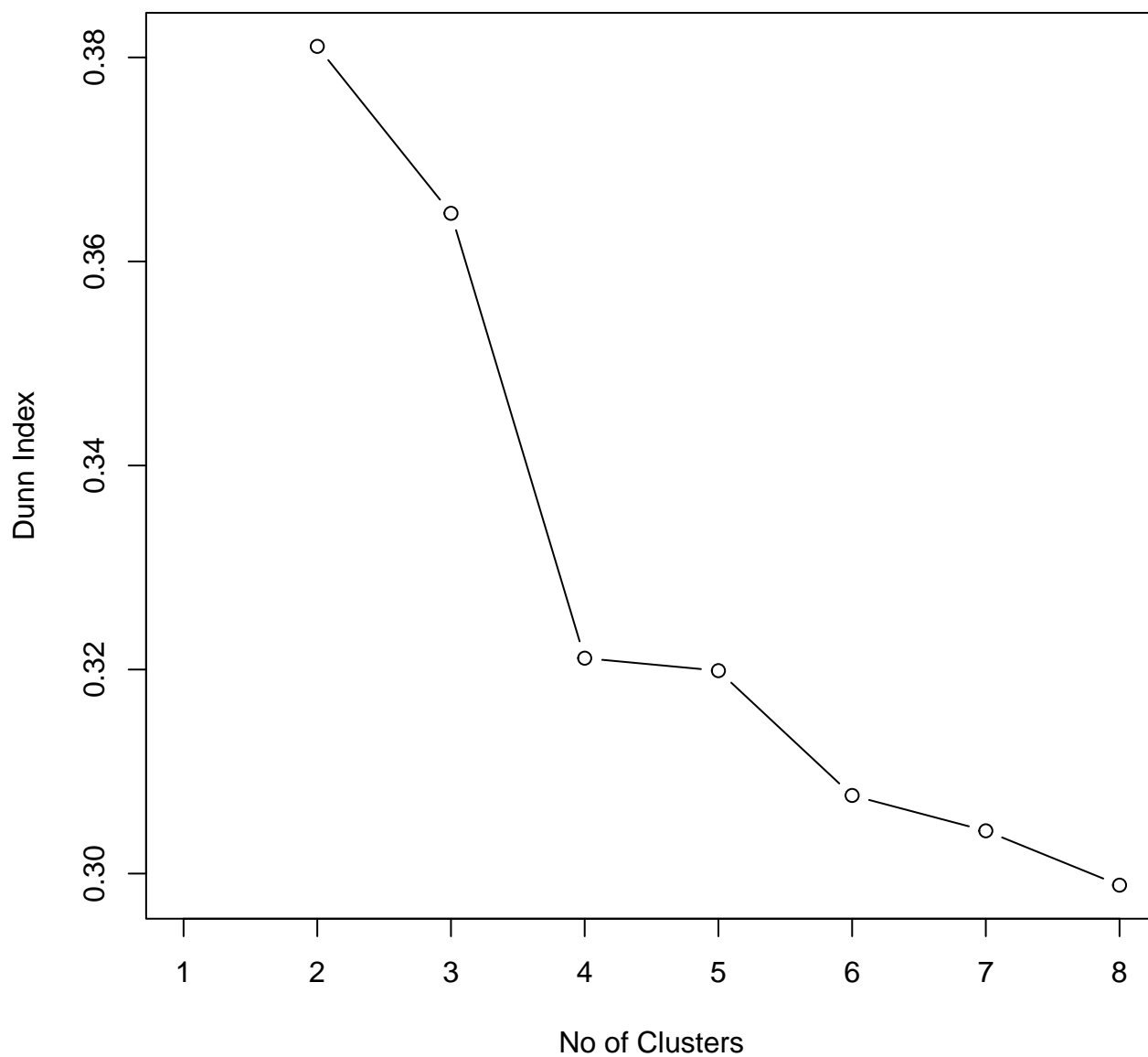
**Dendrogram of single linkage**



```
dun_index=c()
for(i in 2:8){
  clust=cutree(obj3,i)
  dun_index[i]=dunn(dist(data),clust) }
which.max(dun_index)

[1] 2
```

```
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```

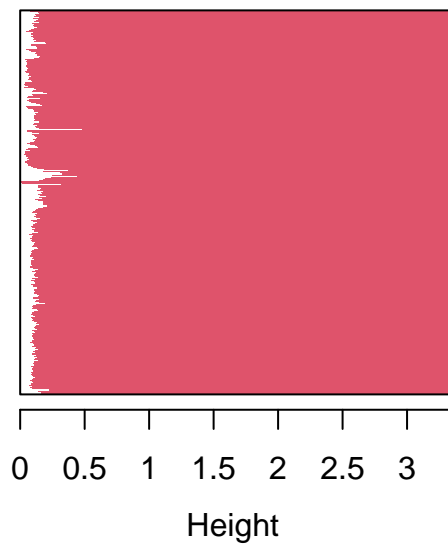


- From plot of Dunn Index the suitable no. of Cluster is 2 as its the least no of clusters that maximises the index.

iv. complete linkage algorithm

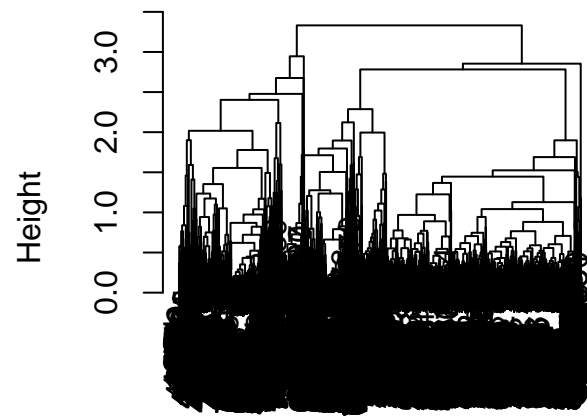
```
par(mfrow=c(1,2))
obj4=agnes(data,method="complete")
plot(obj4,which=1,,main="Banner plot of complete linkage ")
plot(obj4,which=2,,main="Dendrogram of complete linkage ")
```

**Banner plot of complete linkage**



Agglomerative Coefficient = 0.95

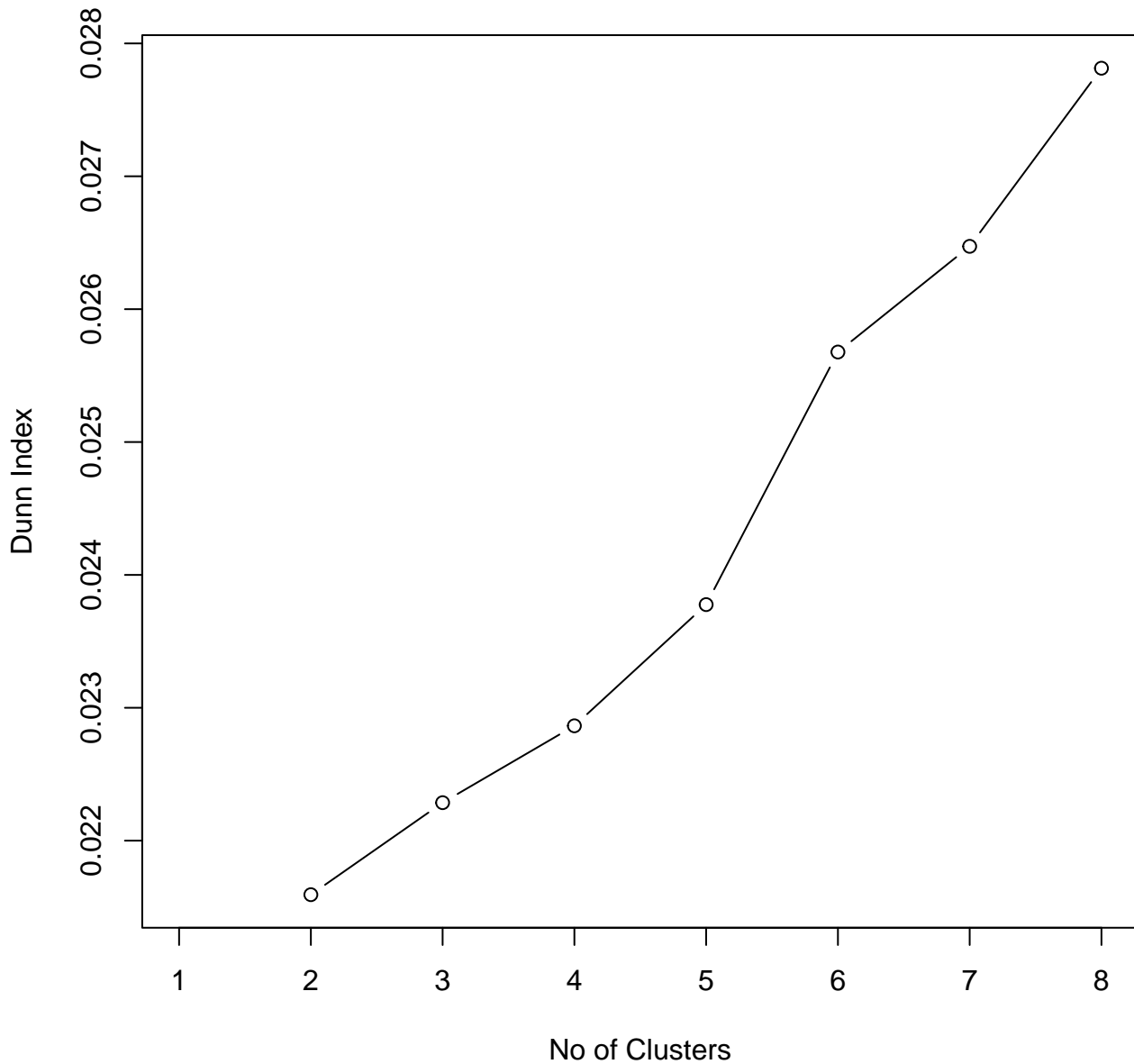
**Dendrogram of complete linkage**



data  
Agglomerative Coefficient = 0.95

```
dun_index=c()  
for(i in 2:8){  
  clust=cutree(obj4,i)  
  dun_index[i]=dunn(dist(data),clust) }  
which.max(dun_index)  
  
[1] 8
```

```
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```

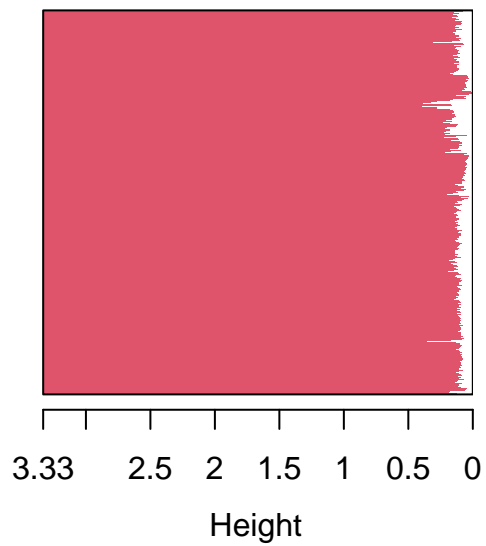


- From plot of Dunn Index the suitable no. of Cluster is 8 but it is keep on increasing, we cannot conclude it because due to high computational time I cannot manage to push the for loop for more number of clustering .

v. divisive clustering algorithm

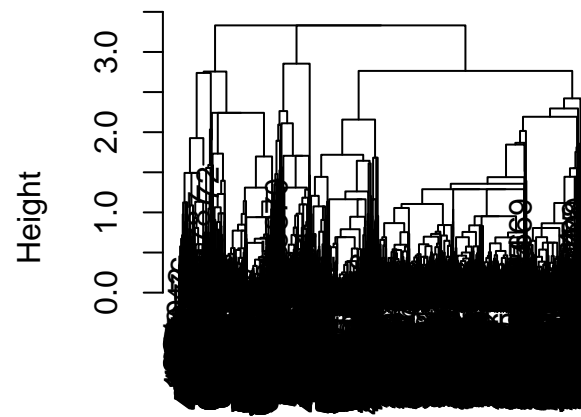
```
par(mfrow=c(1,2))
obj5=diana(data)
plot(obj5,which=1,main="Banner plot")
plot(obj5,which=2,main="Dendogram")
```

**Banner plot**



Divisive Coefficient = 0.94

**Dendrogram**

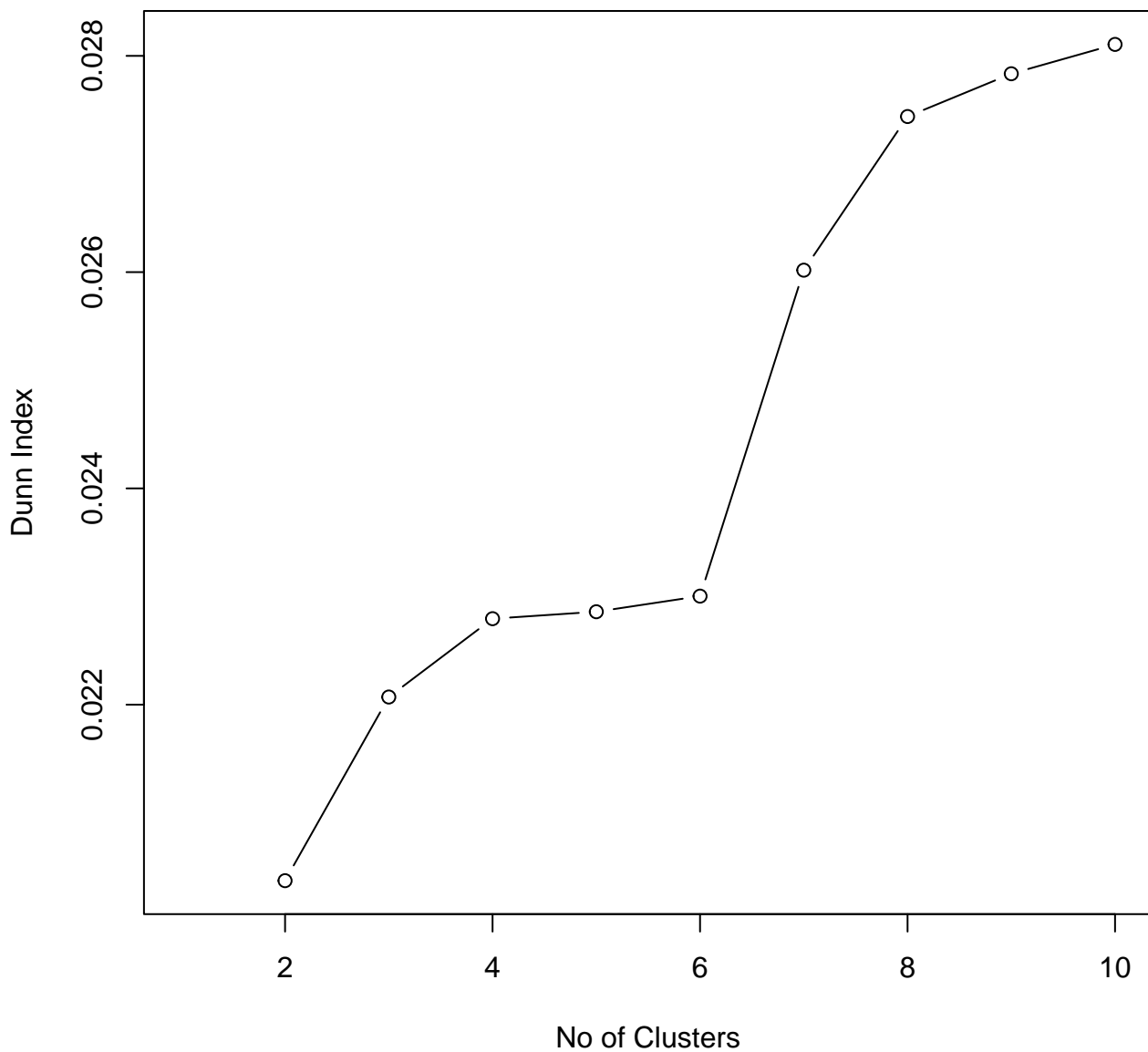


data  
Divisive Coefficient = 0.94

```
dun_index=c()  
for(i in 2:10){  
  clust=cutree(obj5,i)  
  dun_index[i]=dunn(dist(data),clust) }  
which.max(dun_index)  
  
[1] 10
```

```
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```





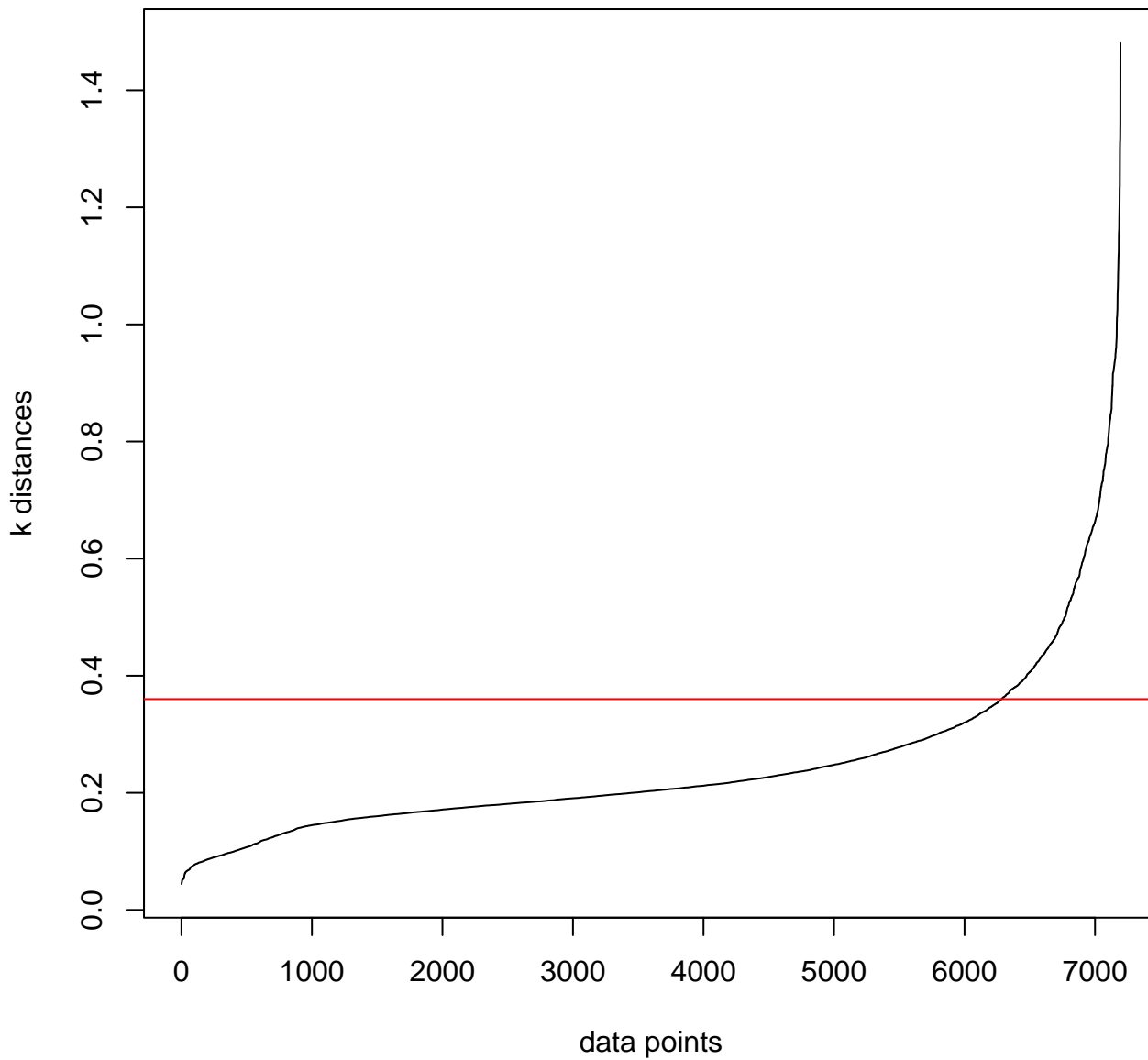
- From plot of Dunn Index the suitable no. of Cluster is 10 but it is keep on increasing, we cannot conclude it because due to high computational time I cannot manage to push the for loop for more number of clustering .

vi. DBSCAN algorithm (with appropriate values of the parameters)

- eps and minPts are selected using the rule mention in webpage ([click here](#))

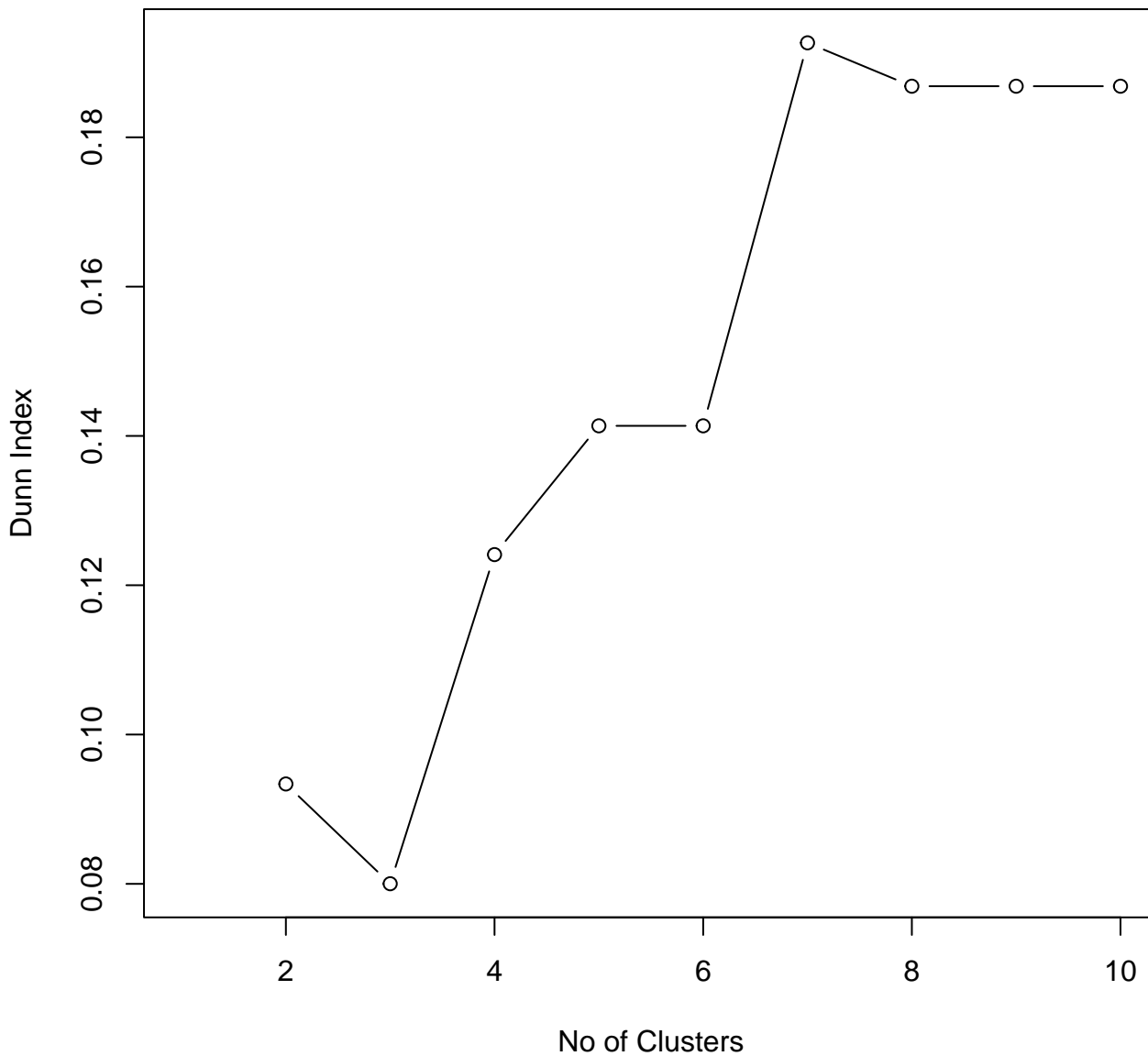
```
--taking minpts as 23 as It is greater than the dimension of the data
--choosing eps from k-distance plot
mat=kNN(data, k = 23)
k_dist=sort(mat$dist[,23])
plot(k_dist,typ="l",xlab="data points",ylab="k distances",main="k distance plot")
abline(h=0.36,col="red")
```

## k distance plot



- We selected a range of values around the value where elbow is created in the plot and calculated the dunn index to select optimal eps value.
- Creating the clustering with best eps and minPts values

```
db_seq=seq(0.3,0.4,0.01)
dun_index=c()
for(i in 2:10){
  obj6=dbscan(data,eps=db_seq[i],minPts=23)
  dun_index[i]=dunn(dist(data),obj6$cluster) }
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```



```
which.max(dun_index)

[1] 7

dbscan(data,eps=db_seq[7],minPts=23)

DBSCAN clustering for 7195 objects.
Parameters: eps = 0.36, minPts = 23
Using euclidean distances and borderpoints = TRUE
The clustering contains 2 cluster(s) and 516 noise points.

  0    1    2
516 6621  58

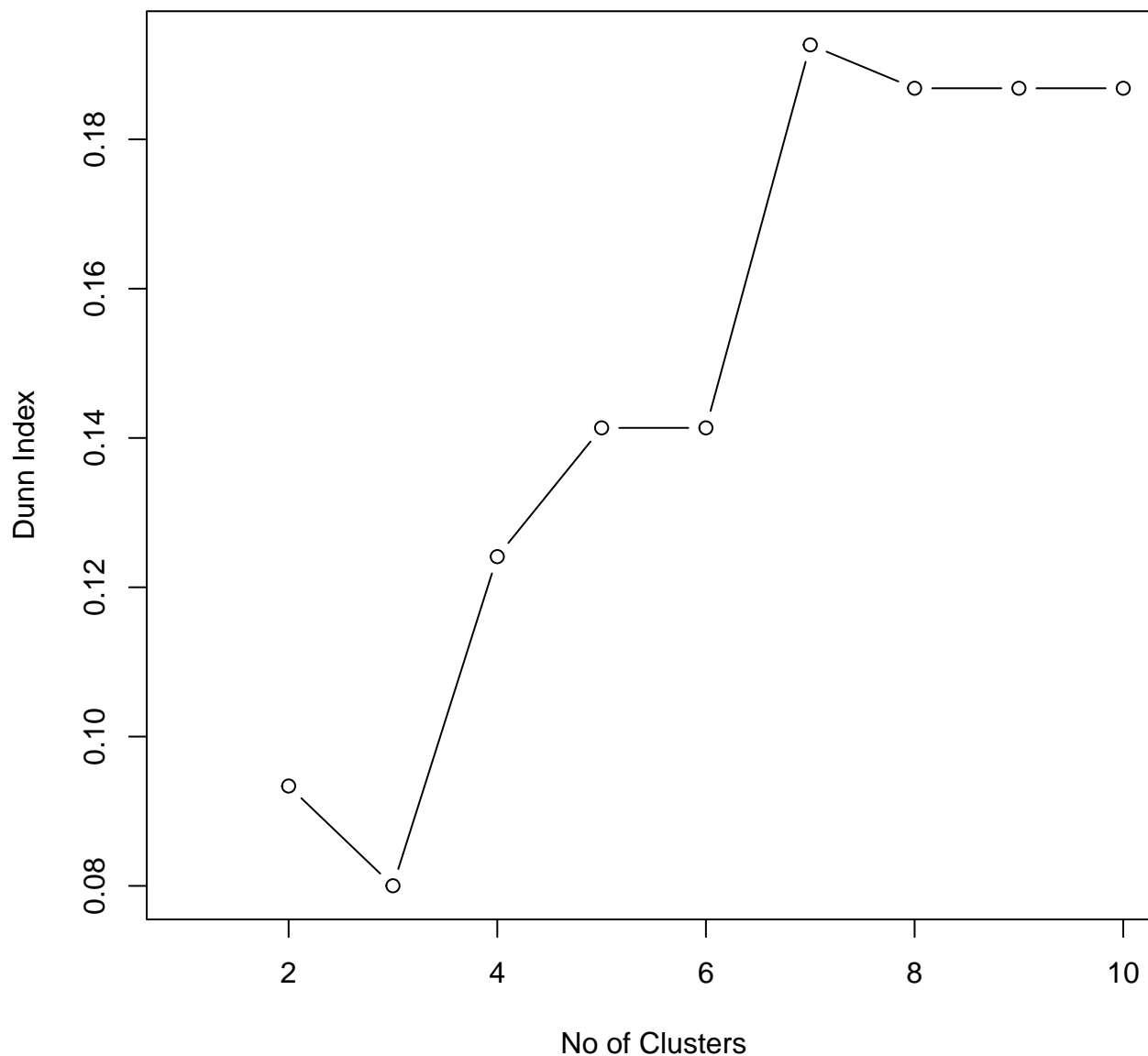
Available fields: cluster, eps, minPts, metric, borderPoints
```

- We have been able to classify the data into 2 clusters using DBSCAN with eps 0.36 and minPts 23 .

vii. OPTICS algorithm (with appropriate values of the parameters)

- We will use the same parameters and create the best clusters.

```
dun_index=c()
for(i in 2:10){
  obj6=optics(data,eps=db_seq[i],minPts=23)
  dun_index[i]=dunn(dist(data),extractDBSCAN(obj6,db_seq[i])$cluster) }
plot(dun_index,typ="b",xlab="No of Clusters",ylab="Dunn Index")
```



```
which.max(dun_index)

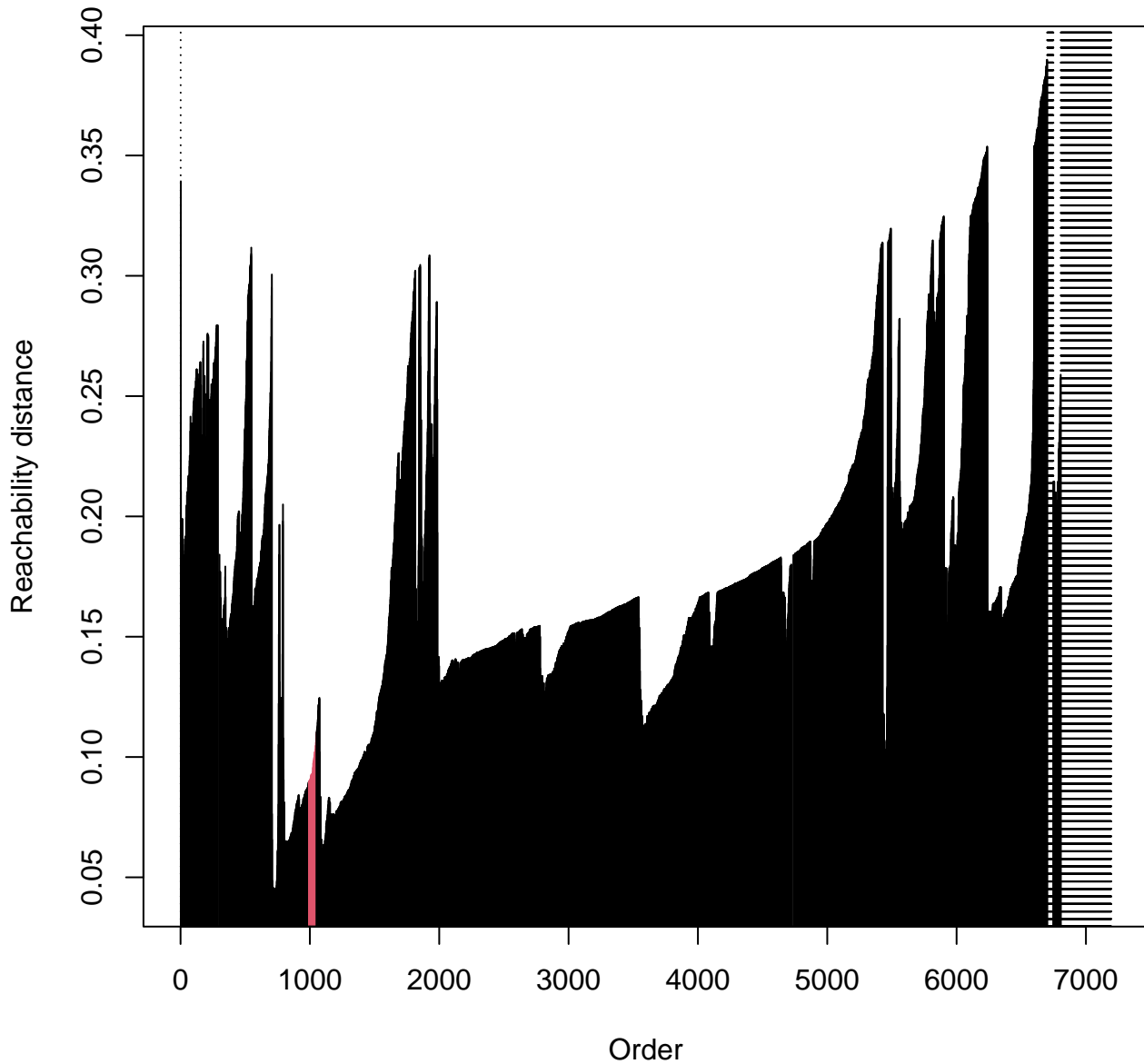
[1] 7

table(extractDBSCAN(optics(data,eps=db_seq[7],minPts=23),db_seq[7])$cluster)

 0    1    2
516 6621  58

plot(obj6,ylab="Reachability distance",col=extractDBSCAN(obj6,db_seq[7])$cluster)
```

## Reachability Plot



- From the Reachability Plot we can clearly observe that 2 clusters in the data and they have different density.
- We have been able to classify the data in 2 clusters using OPTICS with eps 0.36 and minPts 23 .

### viii. Deconvolution of Gaussian Mixture Models through MLE

- Since the data is very large fitting for different no of mixture models very time consuming and the function already gives the model with highest BIC so we are not checking for different numbers of models.

```
gmm_fit <- Mclust(data)
summary(gmm_fit)
```

```
-----
Gaussian finite mixture model fitted by EM algorithm
-----
```

```
Mclust VEV (ellipsoidal, equal shape) model with 7 components:
```

log-likelihood	n	df	BIC	ICL
----------------	---	----	-----	-----

258263.3 7195 1805 500496.2 500318.4

Clustering table:

1	2	3	4	5	6	7
769	2420	1593	609	1136	356	312

- It could classify the data into 7 clusters and the optimal model is selected using BIC.

**Conclusion :** The methods we applied 4 of them (K-means, Single linkage , DBSCAN , OPTICS ) giving the optimal no of clusters as 2 and K-medoids and Gaussian Mixture Models giving optimal cluster no. 4 and 7 respectively .[Divisive and complete linkage methods are inconclusive due to machine inability to process the large data]. So from here we can conclude by majority voting that optimal no of clusters should be 2 but in the data we have information on 4 family of frogs so clearly our methods are inconsistent to split the data in 4 well separated clusters.