Clustering

Clustering:

Clustering is the process of grouping a set of data objects into multiple groups (clusters) so that objects in a cluster are similar to one another, but are dissimilar to objects in other clusters.

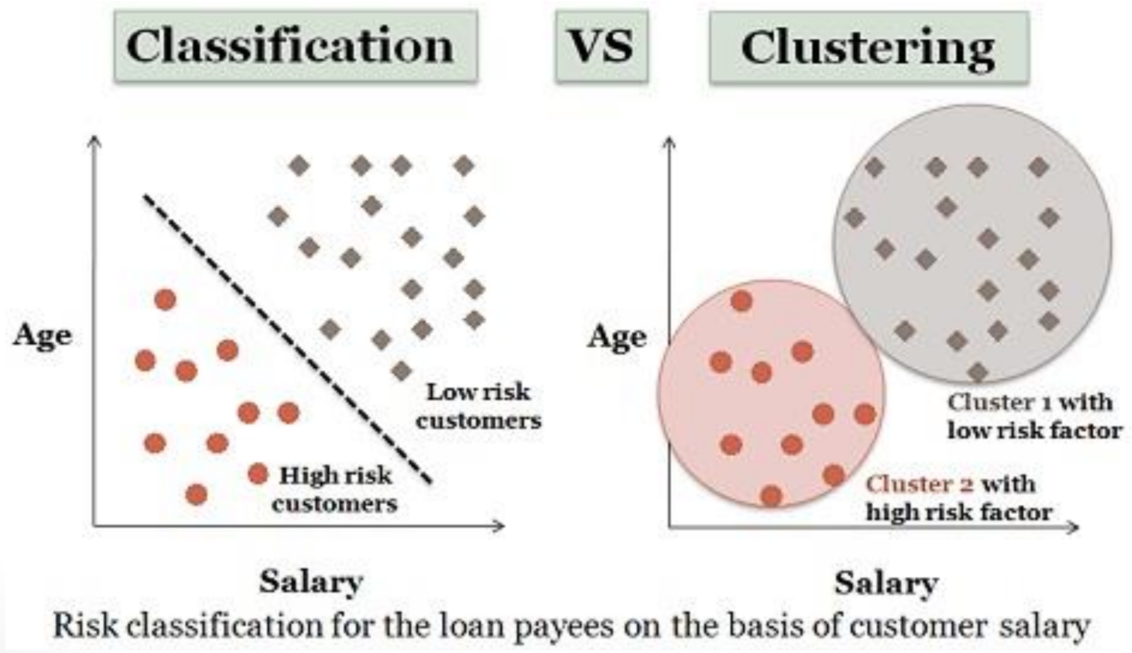
- Clustering as a data mining tool has its roots in many application areas such as biology, security, business intelligence, and Web search.
- Different clustering methods may generate different clusterings on the same data set

Note: Clustering is also called data segmentation in some applications.

Classification vs. Clustering:

Both are used for the categorization of objects into one or more classes based on the attributes.
Classification, we have a set of predefined classes and want to know which class a new object belongs to.

Clustering tries to group a set of objects and find whether there is some similarities between the objects.



Risk classification for the loan payees on the basis of customer salary

Why Clustering?

1. Gain some insight into the structure of the data.
2. Reduce the size and complexity of the dataset.
3. Prepare for other data mining (and AI) techniques.
4. Ability to deal with the noise data.
5. Detecting outliers.

Requirements for Cluster Analysis:

1. Scalability.

2. Ability to deal with different types of attributes.

3. Discovery of clusters with arbitrary shape.

Note: Standardizing (scaling) is not necessary when the input variables are measured on the same scale

Clustering Methods:

The clustering methods can be classified as follows:

1. Hierarchical methods.
2. Density-based methods.
3. Partitioning methods.
4. Grid-based methods.
5. Fuzzy methods.

1. K-Means Method:

K-means clustering is a simple approach for partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

- K-Means is one of the most popular "clustering" algorithms.

Steps of the K-Means Clustering:

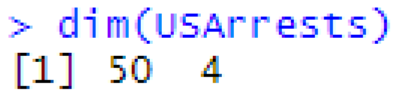
We need to pre-specifiy number of clusters,

1. Choose the number of clusters k.
2. Selects K centroids (K rows chosen at random).
3. Assigns all data points to its closest centroid.
4. Recalculates the centroids as the average of all data points in a cluster.
5. Assigns data points to their closest centroids.
6. Continue steps 3 and 4 until the observations are not reassigned.

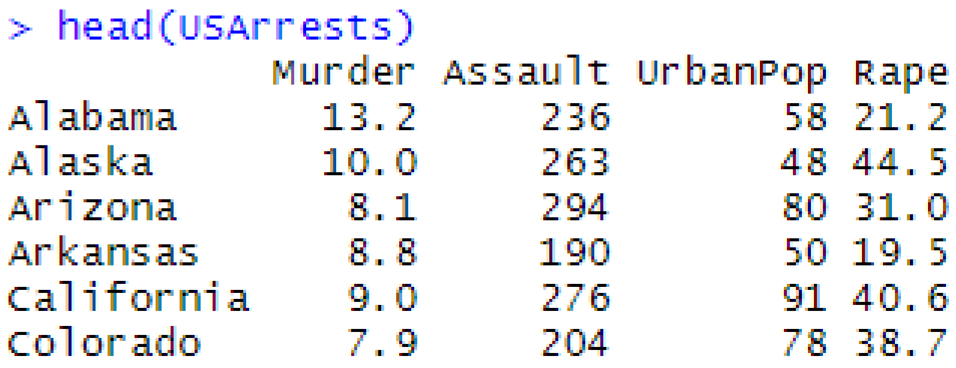
Example 1: (USArrests)

USArrests (available in r). This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

1. Show the dataset dimension and head.



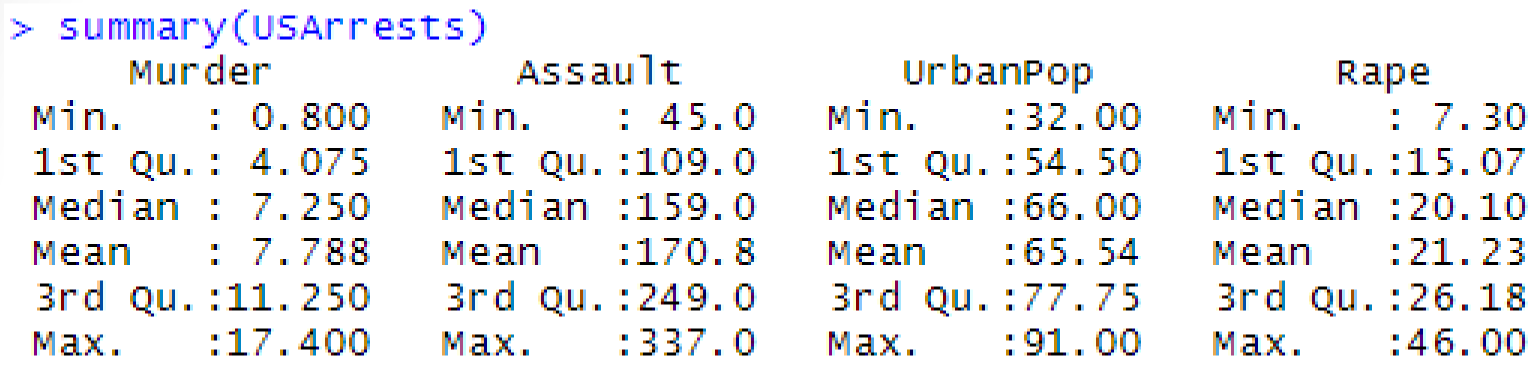
```
> dim(USArrests)
[1] 50  4
```



```
> head(USArrests)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

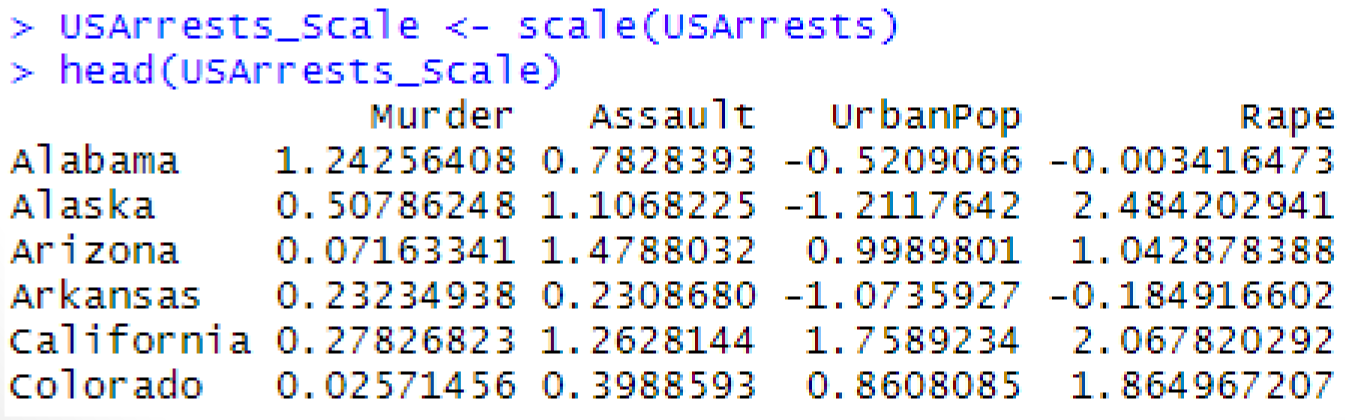
Example 1: (USArrests)

2. Check for missing data.



```
> summary(USArrests)
     Murder          Assault         UrbanPop          Rape      
 Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30  
 1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07  
 Median : 7.250   Median :159.0   Median :66.00   Median :20.10  
 Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23  
 3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18  
 Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00  
```

3. One of the clustering requirements is scaling. Scale the data and show the dataset head.



```
> USArrests_Scale <- scale(USArrests)
> head(USArrests_Scale)
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

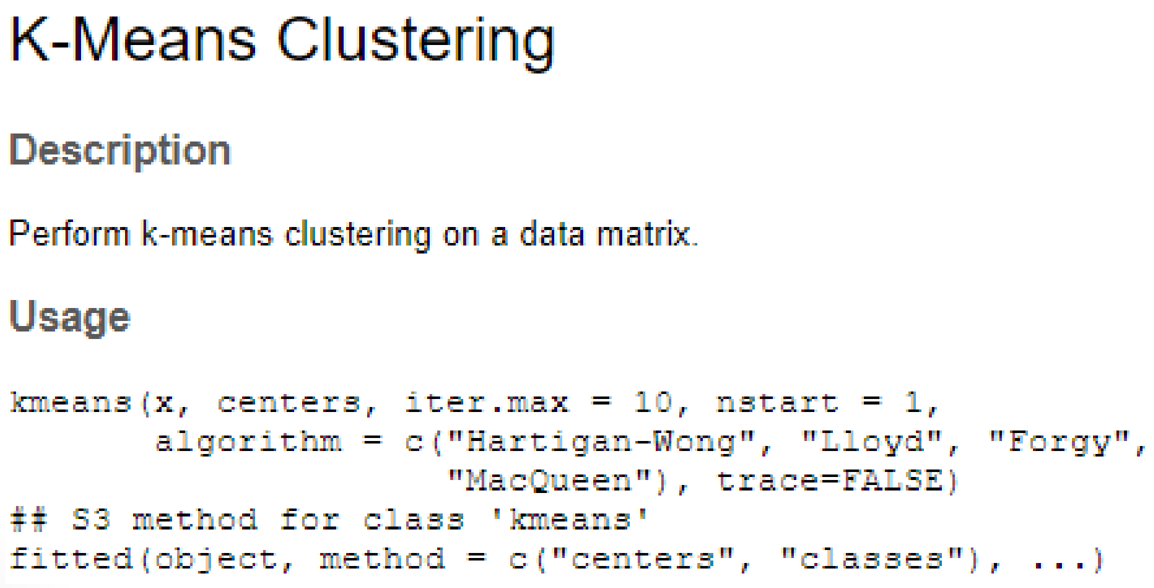
Example 1: (USArrests)

4. Install the factoextra package.

5. Check the description of kmeans() function.

```
install.packages("factoextra")
library("factoextra")
```

```
> ?kmeans
```



K-Means Clustering

Description

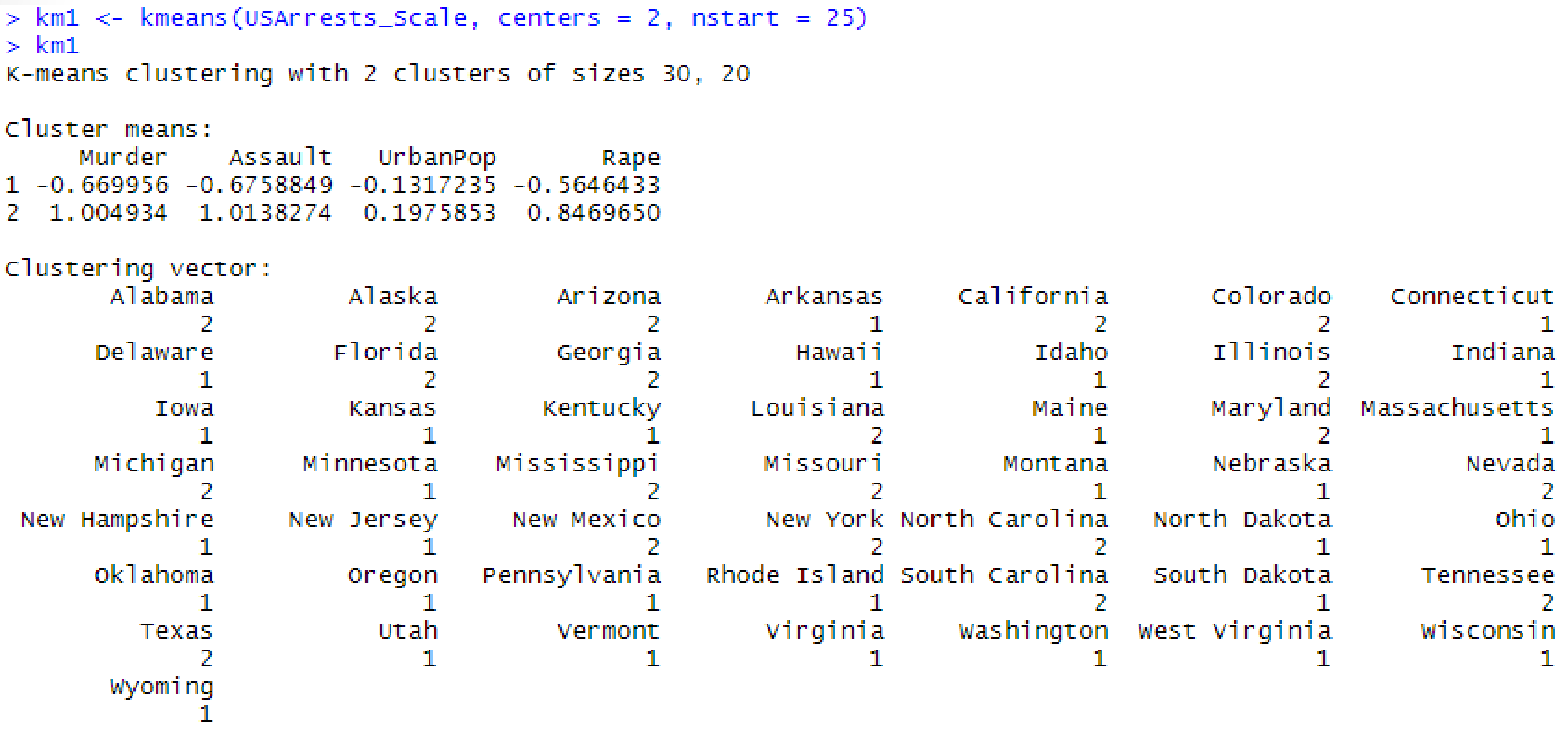
Perform k-means clustering on a data matrix.

Usage

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
                     "MacQueen"), trace=FALSE)
## S3 method for class 'kmeans'
fitted(object, method = c("centers", "classes"), ...)
```

Example 1: (USArrests)

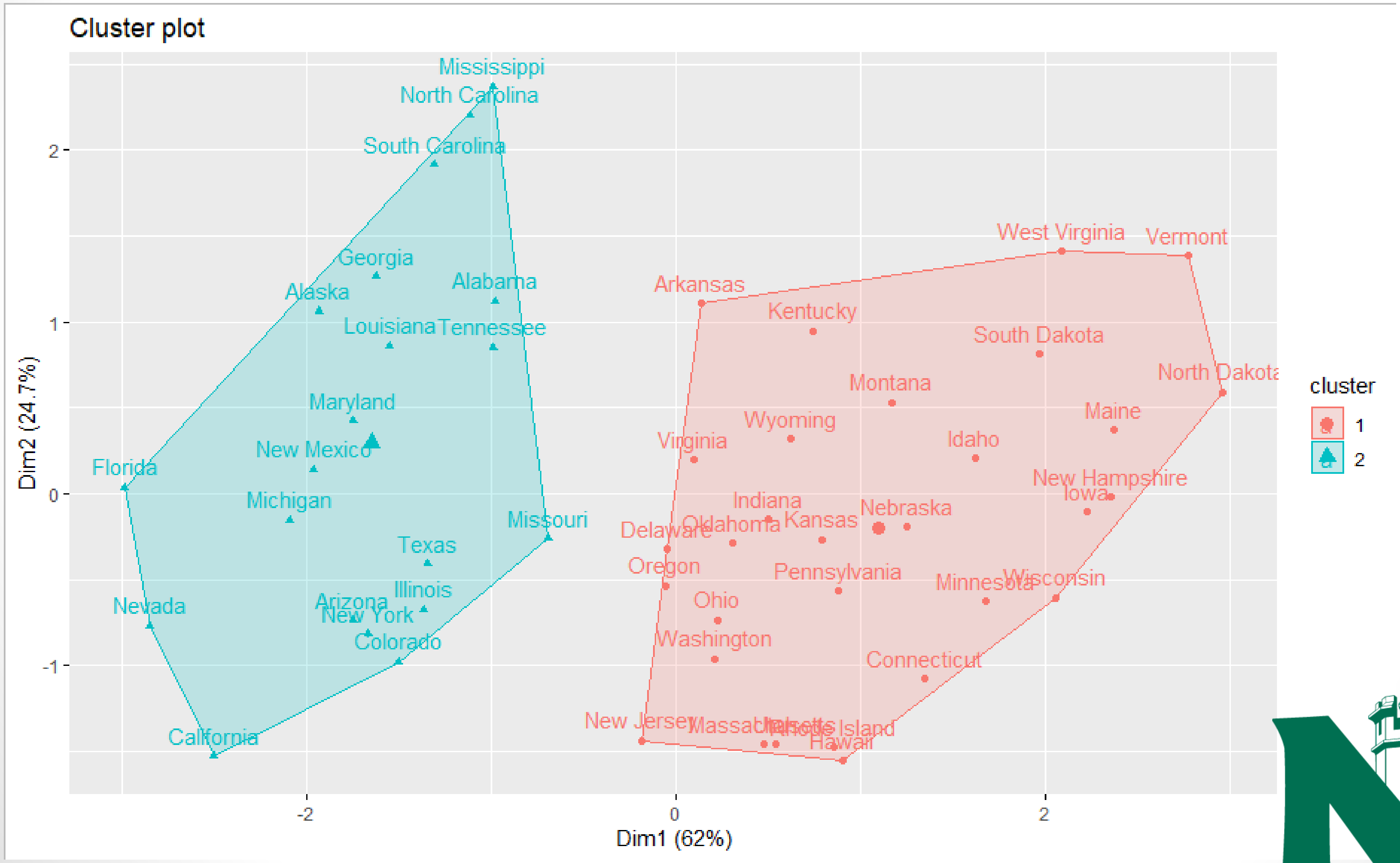
6. Compute k-means with the kmeans function. Group the data into two clusters (centers = 2). Use option nstart that attempts multiple initial configurations and reports on the best one.



```
> km1 <- kmeans(USArrests_Scale, centers = 2, nstart = 25)
> km1
K-means clustering with 2 clusters of sizes 30, 20

Cluster means:
     Murder    Assault   UrbanPop       Rape
1 -0.669956 -0.6758849 -0.1317235 -0.5646433
2  1.004934  1.0138274  0.1975853  0.8469650

Clustering vector:
       Alabama         Alaska        Arizona       Arkansas     California       Colorado    Connecticut 
             2              2              2              1              2              2              1 
      Delaware        Florida        Georgia         Hawaii          Idaho       Illinois        Indiana 
             1              2              2              1              1              2              1 
          Iowa         Kansas       Kentucky      Louisiana          Maine       Maryland  Massachusetts 
             1              1              1              2              1              2              1 
      Michigan      Minnesota    Mississippi       Missouri        Montana       Nebraska         Nevada 
             2              1              2              2              1              1              2 
 New Hampshire     New Jersey     New Mexico       New York North Carolina   North Dakota           Ohio 
             1              1              2              2              2              1              1 
      Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee 
             1              1              1              1              2              1              2 
         Texas           Utah        Vermont       Virginia     Washington  West Virginia      Wisconsin 
             2              1              1              1              1              1              1 
       Wyoming 
             1 
```

Example 1: (USArrests)

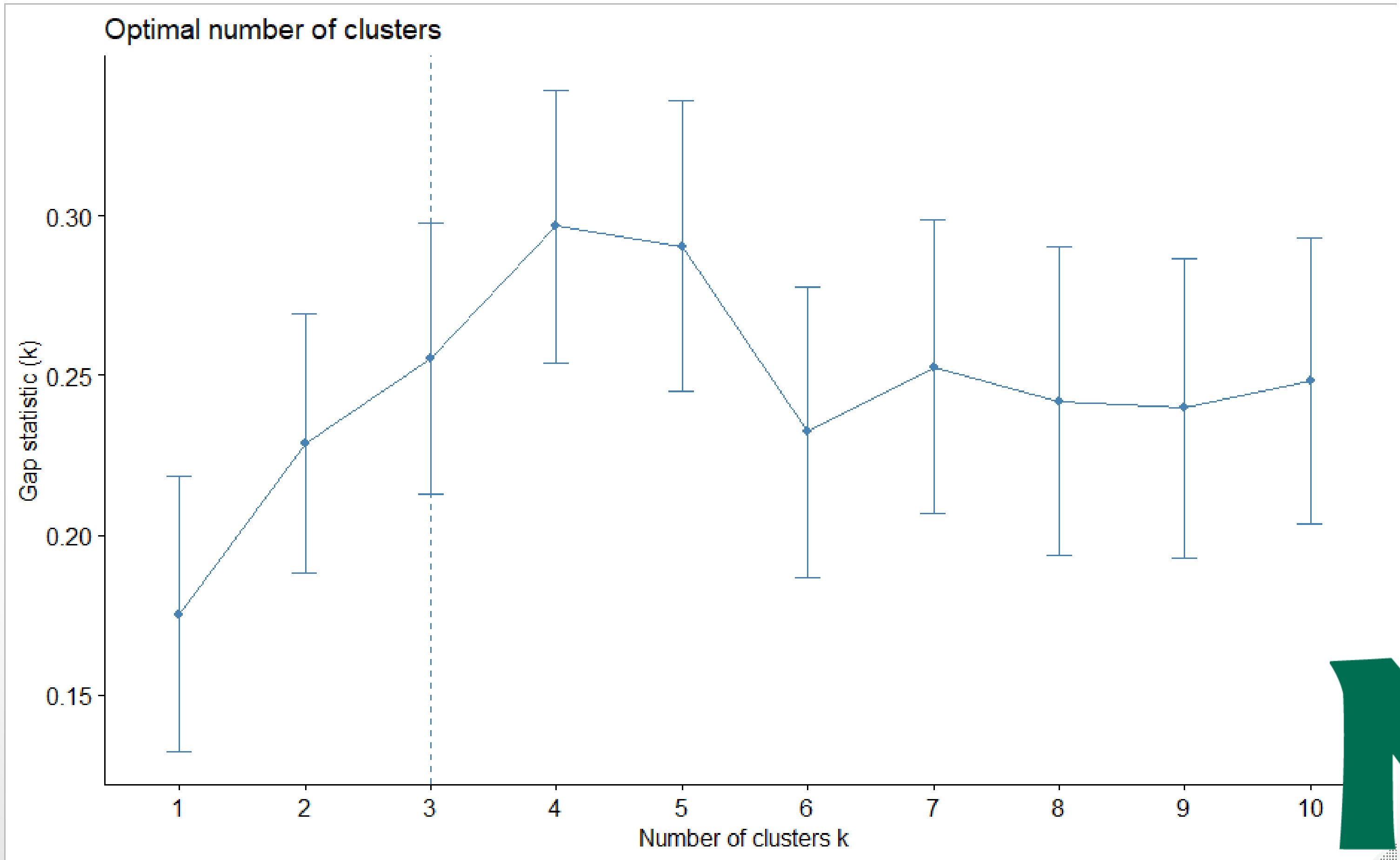
7. Use fviz_cluster() function to provides a nice illustration of the clusters.



Example 1: (USArrests)

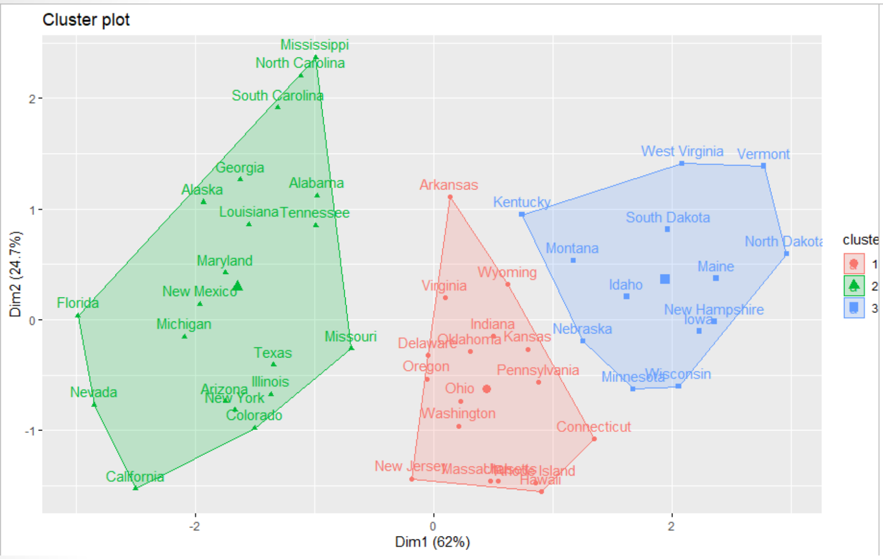
8. Find the Optimal Clusters.

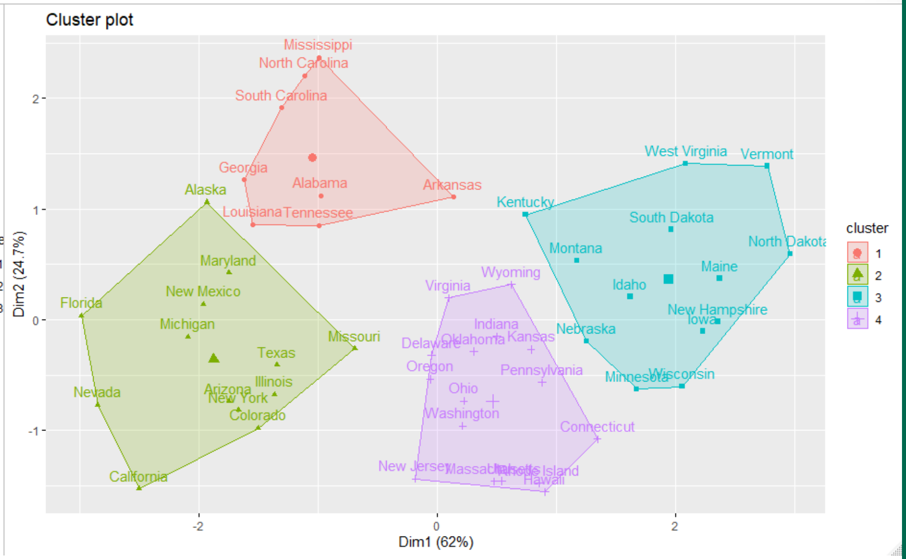
```
> fviz_nbclust(USArrests_Scale, kmeans, method = "gap_stat")
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 100)  [one "." per sample]:
.................................................. 50
.................................................. 100
```

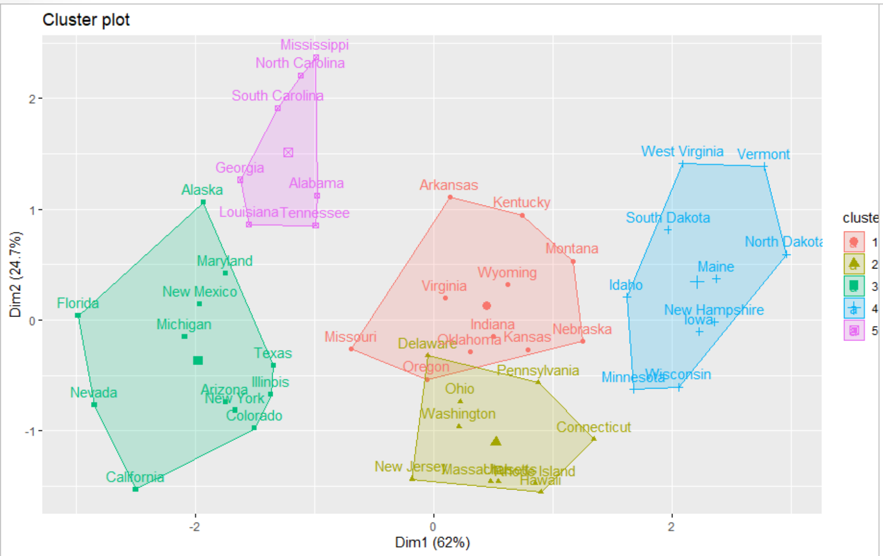


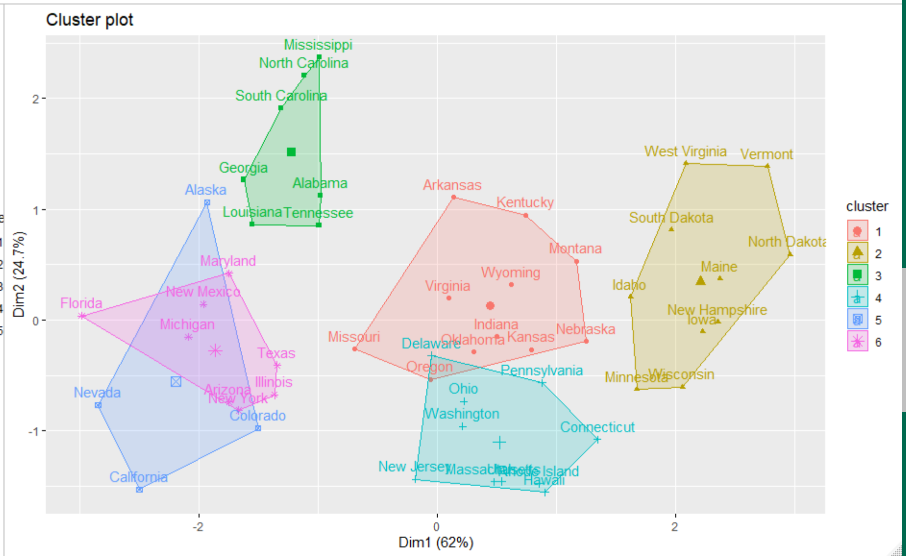
Example 1: (USArrests)

9. Compare between different number of clusters.



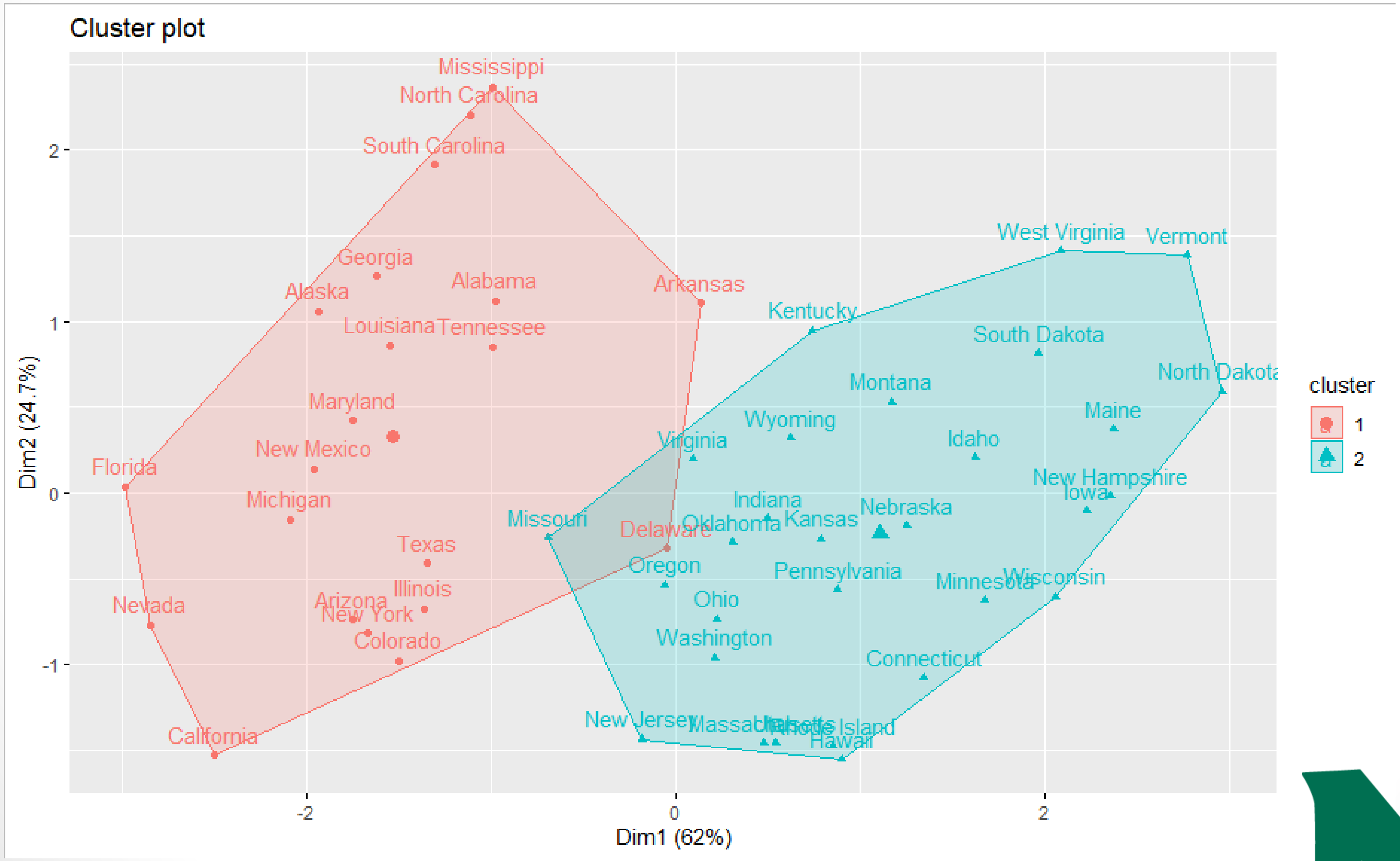






Example 1: (USArrests)

10. Compute k-means for the original dataset (without scaling).



Note: we can observe overlapping in the results.

Advantages of the K-Means :

1. Easy to understand and implement.
2. Works well with small or large data.
3. Do not have to calculate the distance measures between all pairs of subjects.

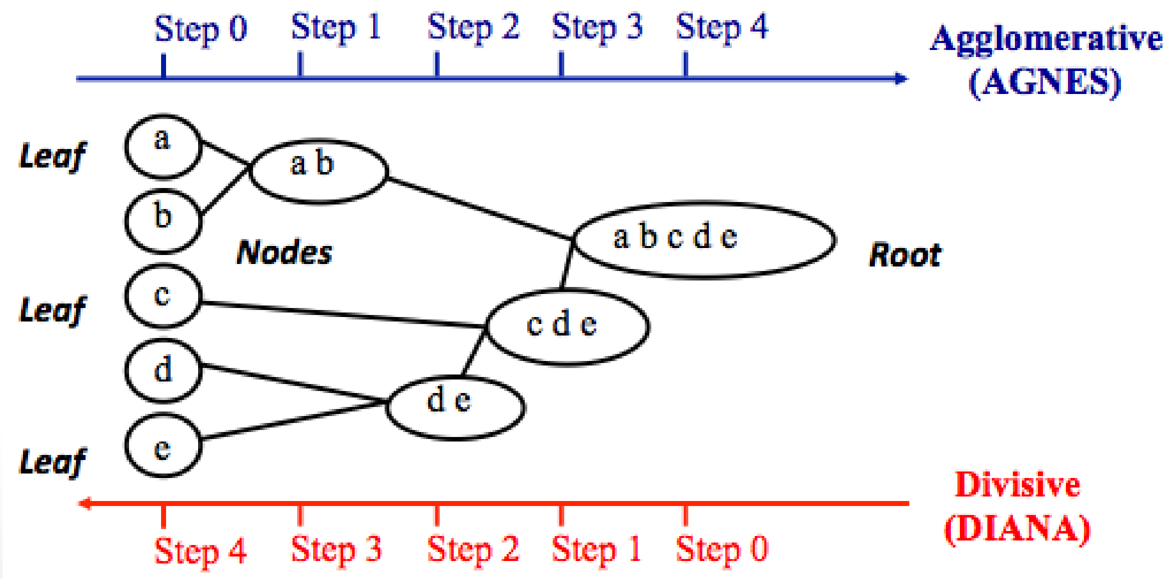
Disadvantages of the K-Means :

1. Needs to chose number of clusters.
2. It can be very sensitive to the choice of initial cluster centers.

2. Hierarchical Methods:

1. **Agglomerative method (bottom-up):** starts by defining each data point as a cluster. Then, the two closest clusters are combined into a new cluster.
   - This is the most common type of hierarchical clustering

2. **Divisive method:** starts by putting all data points into a single cluster. Then we divide this cluster into two clusters.



Combining Clusters in the Agglomerative Approach:

There are five different methods for this approach:

1. Single Linkage.

2. Complete Linkage.

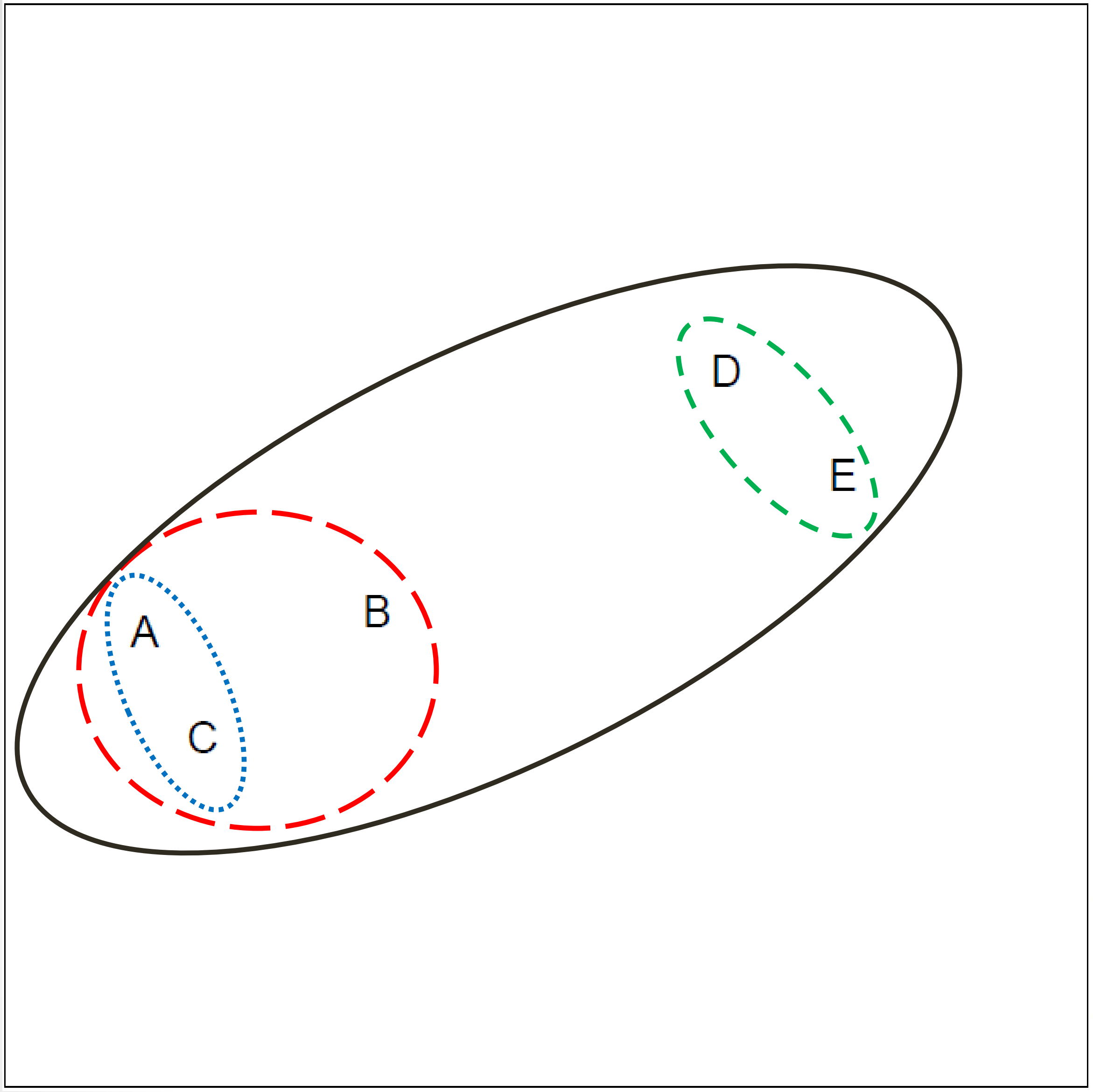
3. Average Linkage.

4. Centroid Method.

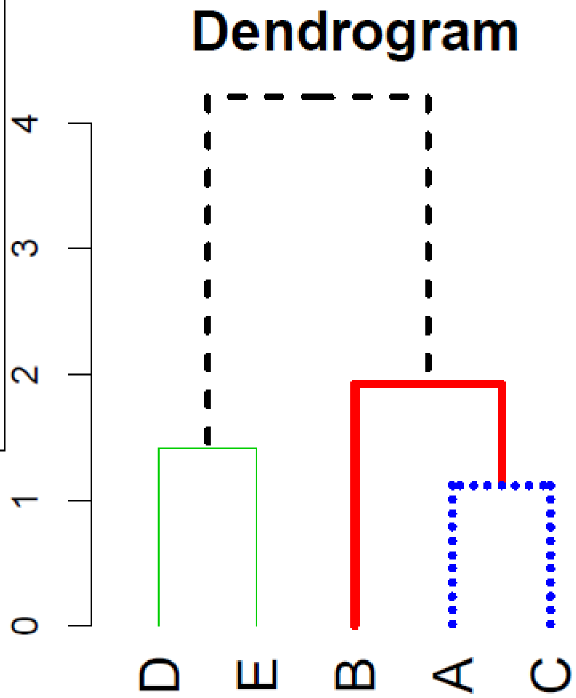
5. Ward’s Method.

Agglomerative Approaches:

Single Linkage	$d_{12} = \min_{i,j} d(\mathbf{X}_i, \mathbf{Y}_j)$
Complete Linkage	$d_{12} = \max_{i,j} d(\mathbf{X}_i, \mathbf{Y}_j)$
Average Linkage	$d_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(\mathbf{X}_i, \mathbf{Y}_j)$
Centroid Method	$d_{12} = d(\bar{\mathbf{x}}, \bar{\mathbf{y}})$



D
E
B
A
C

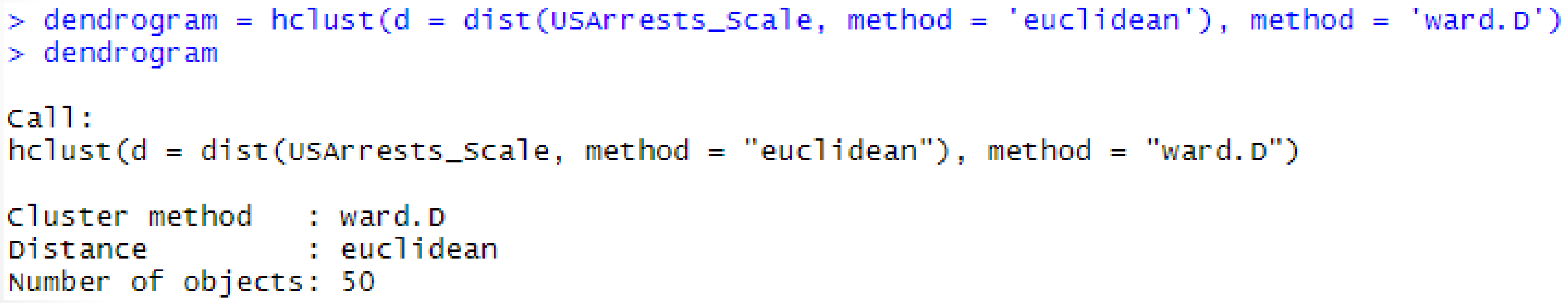


Dendrogram
0
1
2
3
4
D
E
B
A
C

Example 2: (USArrests)

For the same dataset in example 1 (*USArrests*).

1. Perform Agglomerative Hierarchical clustering using Ward's method.

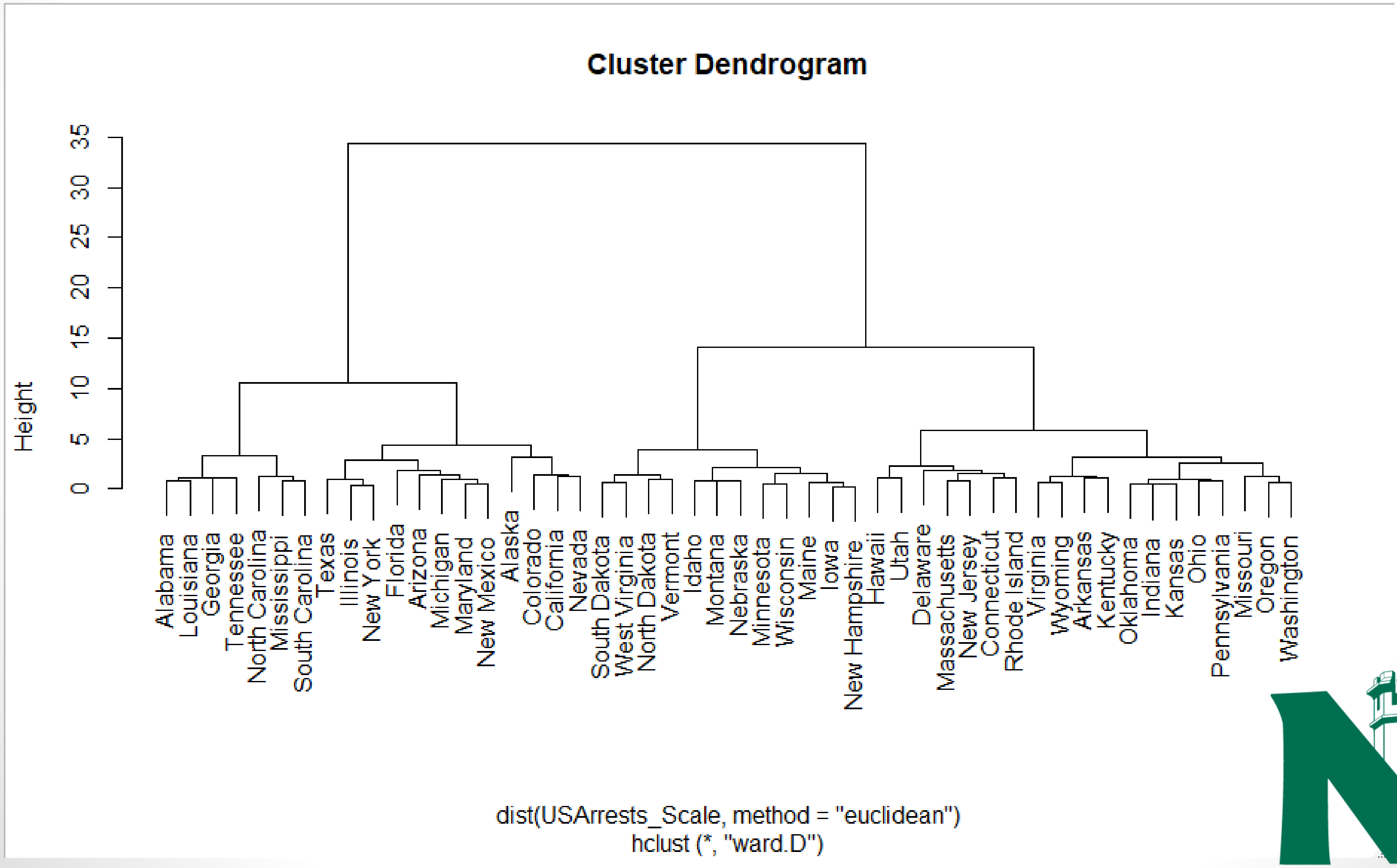


```
> dendrogram = hclust(d = dist(USArrests_Scale, method = 'euclidean'), method = 'ward.D')
> dendrogram

Call:
hclust(d = dist(USArrests_Scale, method = "euclidean"), method = "ward.D")

Cluster method   : ward.D
Distance         : euclidean
Number of objects: 50
```

Example 2: (USArrests)

2. Plot the dendrogram.

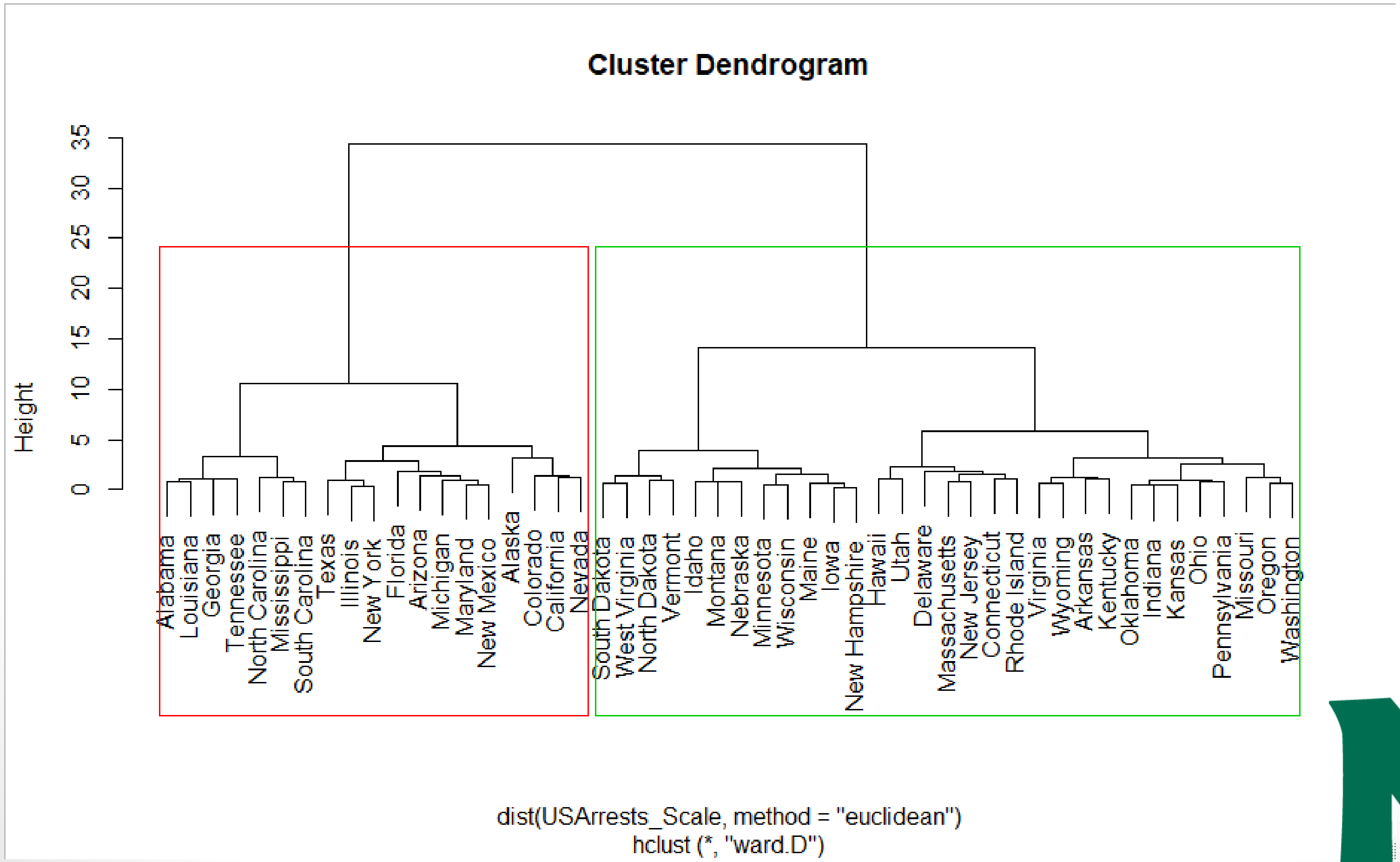
```
> plot(dendrogram)
```



Example 2: (USArrests)

3. Draw the dendrogram with a border around the 2 clusters.

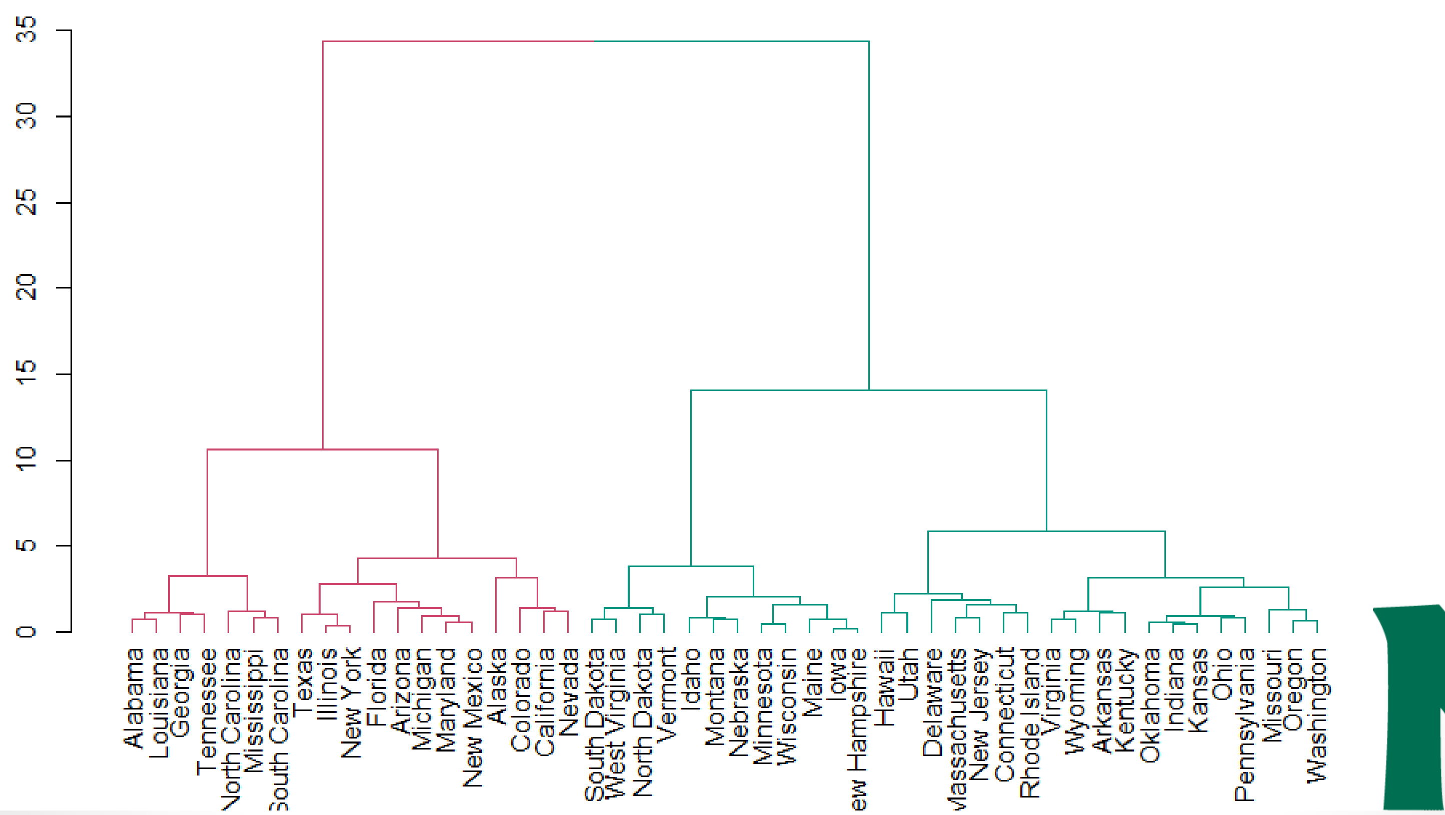
```
> plot(dendrogram, main="centroid")
> rect.hclust(dendrogram, k = 2, border = 2:5)
```



Example 2: (USArrests)

4. Use the color_branches() function from the dendextend library to visualize your tree with different colored branches.

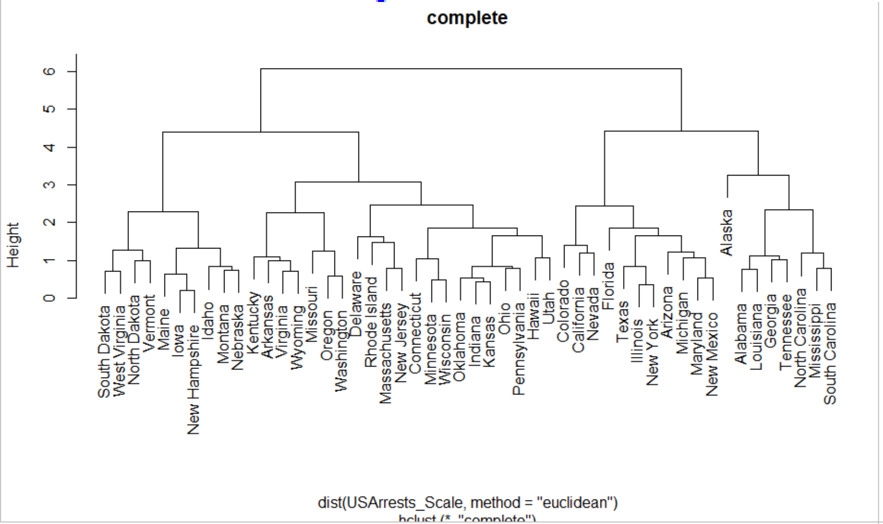
```
> dend1 <- as.dendrogram(dendrogram)
> dend1_G <- color_branches(dend1, h = 15)
> plot(dend1_G)
```

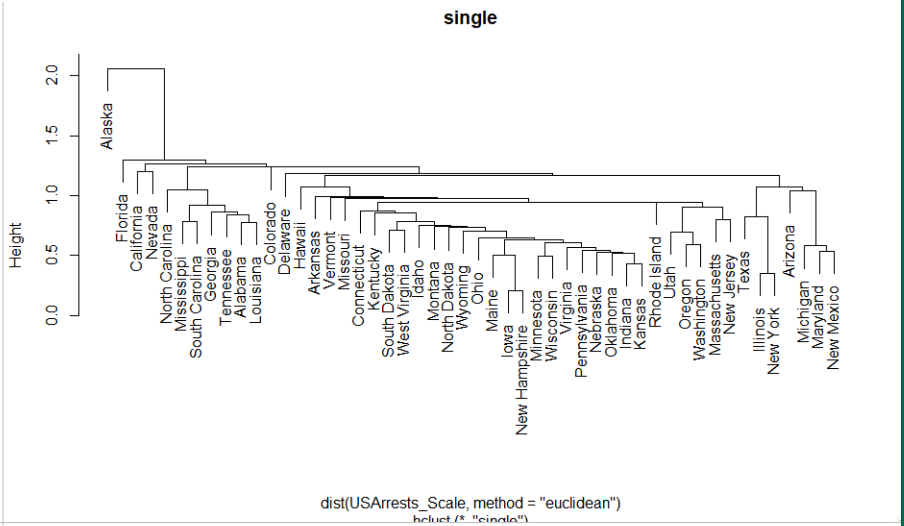


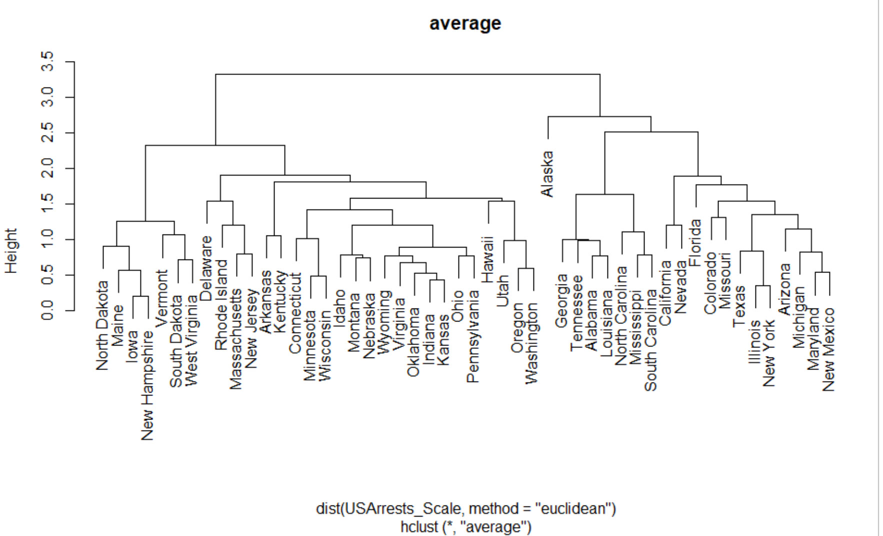
Example 2: (USArrests)

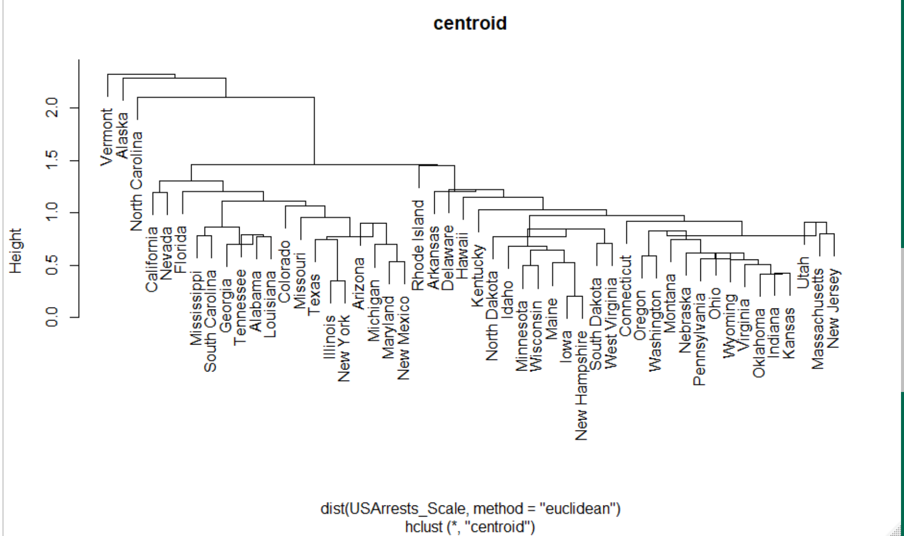
5. Compare the four different linkage methods: single, complete, average and centroid.

```
> dendrogram1 = hclust(dist(USArrests_Scale, method = 'euclidean'), method = 'single')
> plot(dendrogram1, main="single")
```









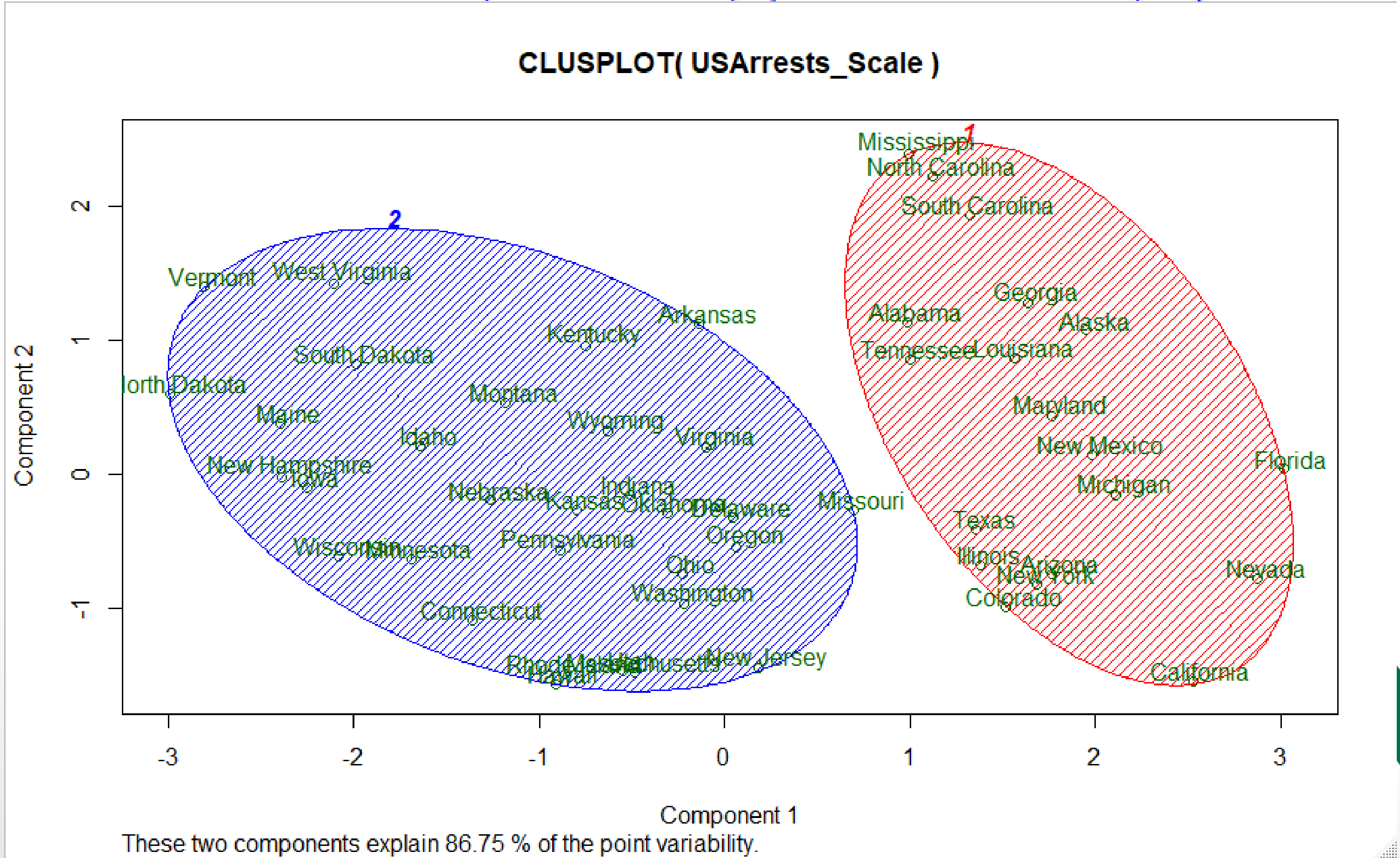
Example 3: (USArrests)

6. Present the clusters visually.

We need to install cluster package.

```
install.packages("cluster")
library("cluster")
```

```
> clusplot(USArrests_Scale, y_hc, lines = 0, shade = TRUE,
+          color = TRUE, labels= 2, plotchar = FALSE, span = TRUE)
```

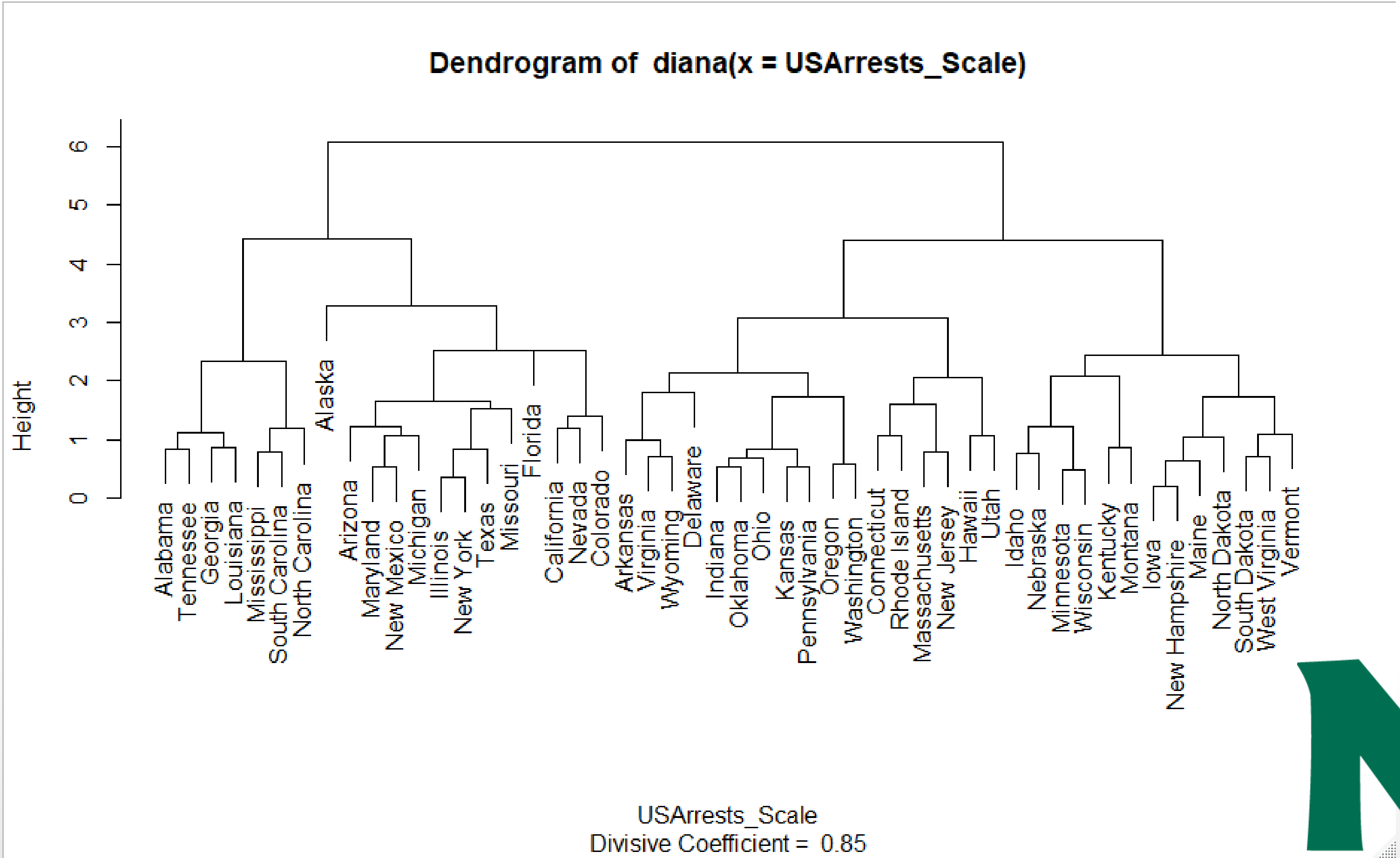


Example 3: (USArrests)

7. Perform Divisive method.

We need to install cluster package.

```
> hc1 = diana(USArrests_Scale)
> plot(hc1)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

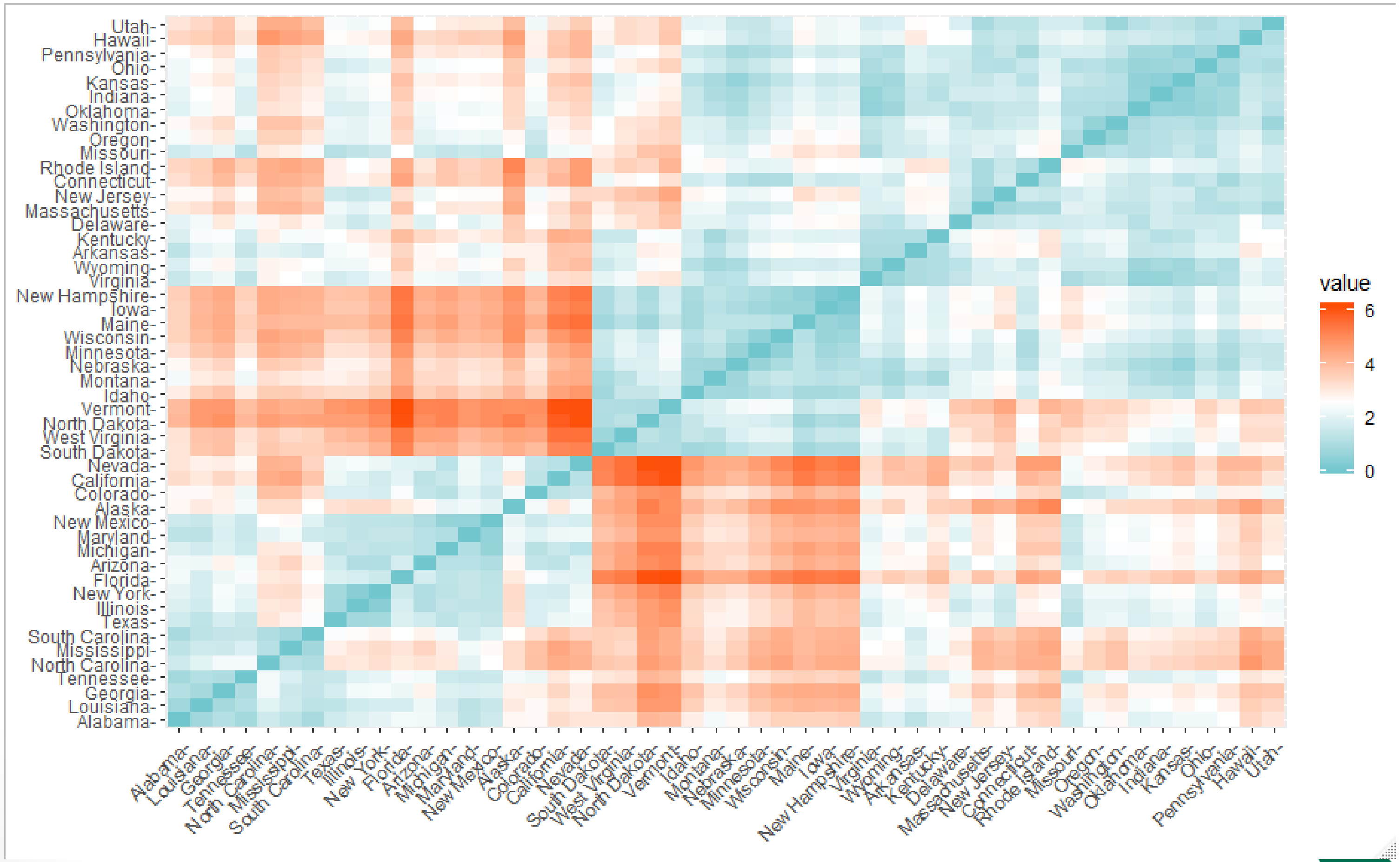


```
install.packages("factoextra")
library("factoextra")
```

Example 4: (Iris)

8. Compute and visualize the distance matrix.

```
> distance = get_dist(USArrests_Scale)
> fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



➢ States have large dissimilarities (red) versus those that appear to be fairly similar (teal)