Assignment No	12
Title	Apriori and Clustering
Objective	12.1 Aprior Algorithm 12.2 K means Algorithm 12.3 Agglomerative Hierarchical
Roll No	MCA2565

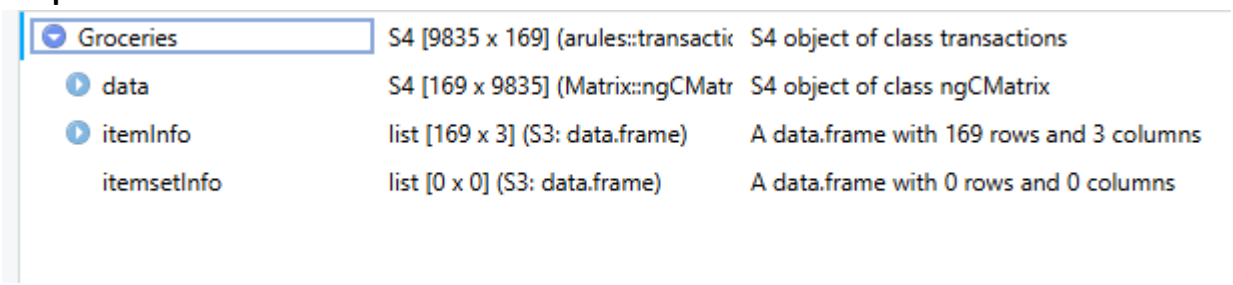
12.1) Apriori Algorithm

Source code :

1) apriori sample dataset.R

```
install.packages("arules")
install.packages("arulesViz")

library(arules)
library(arulesViz)
data("Groceries")
inspect(head(Groceries,3))
View(Groceries)
rules<-apriori(Groceries,parameter=
            list(support=0.01,confidence=0.09,maxlen=3,minlen=2))
rules

inspect(rules)
```

Output:



```
> data("Groceries")
> inspect(head(Groceries,3))
    items
[1] {citrus fruit,
     semi-finished bread,
     margarine,
     ready soups}
[2] {tropical fruit,
     yogurt,
     coffee}
[3] {whole milk}
> View(Groceries)
> rules<-apriori(Groceries,parameter=
+                  list(support=0.01,confidence=0.09,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen ma
       0.09    0.1    1 none FALSE            TRUE       5    0.01      2
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 98

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ...[88 item(s)] done [0.00s].
creating S4 object  ... done [0.00s].

Warning message:
In apriori(Groceries, parameter = list(support = 0.01, confidence = 0.09,  :
  Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!

> rules
set of 445 rules
> inspect(rules)
     lhs                     rhs                  support    confidence
[1]  {hard cheese}        => {whole milk}         0.01006609 0.41078838
[2]  {butter milk}        => {other vegetables}   0.01037112 0.37090909
[3]  {butter milk}        => {whole milk}         0.01159126 0.41454545
[4]  {ham}                => {whole milk}         0.01148958 0.44140625
[5]  {sliced cheese}      => {whole milk}         0.01077783 0.43983402
[6]  {oil}                => {whole milk}         0.01128622 0.40217391
[7]  {onions}             => {other vegetables}   0.01423488 0.45901639
[8]  {onions}             => {whole milk}         0.01209964 0.39016393
[124] 0.13950178 1.7423115 140
[125] 0.05856634 1.3686151 145
[ reached 'max' / getOption("max.print") -- omitted 320 rows ]
```

2) apriori algorithm

apriorialogorithm.R

```
mba_data <- read.csv("data_apriori.csv", stringsAsFactors = FALSE)
getwd()
setwd("C:/Users/mcamock/Desktop/Advanced DBMS")

#Check the structure
str(mba_data)
View(mba_data)

#Ensure correct column names (case-sensitive)
colnames(mba_data) <- c("Customer_id", "Products")

# Split 'Products' into individual items
trans_list <- strsplit(mba_data$Products,",") #Split by comma
trans_list <- lapply(trans_list, trimws)      #remove leading/trained

# Assign customer IDs as transaction names
names(trans_list) <- mba_data$Customer_id

# Convert list to 'transactions' object
trans <- as(trans_list, "transactions")

#inspect the transactions
inspect(head(trans, 5))
itemLabels(trans)
summary(trans)
#plot item frequencies
itemFrequencyPlot(trans,topN=10,type="absolute",main="Top 10 most frequent items")
#generate Apriori rules
rules <- apriori(
  trans,
  parameter = list(support = 0.05, confidence = 0.03, minlen = 2, maxlen=3)
)
#check rule summary
summary(rules)
#Inspect top rules by lift
inspect(head(sort(rules,by="lift"),10))
#visualize rules (interactive graph)
plot(rules,method="graph",engine="htmlwidget")

#optional: Export rules to CSV
rules_df <- as(rules,"data.frame")
write.csv(rules_df,"aprior_rules_output.csv", row.names = FALSE)

#print final countdown
```

cat("Apriori analysis complete! Rules saved to 'apriori_rules_output.csv'\n")

Output :-

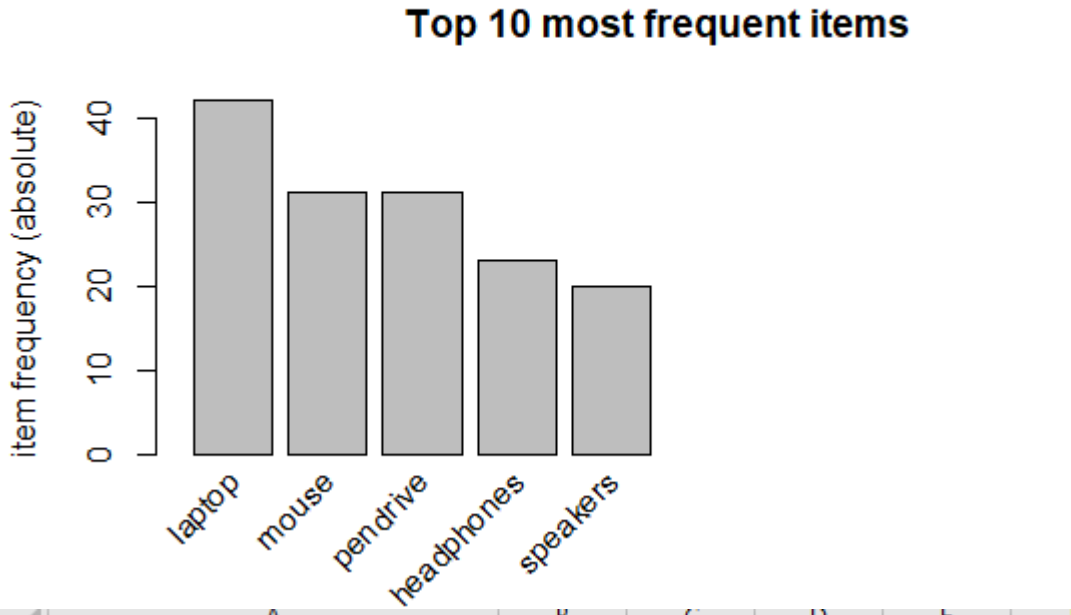
```
> mba_data <- read.csv("data_apriori.csv", stringsAsFactors = FALSE)
> getwd()
[1] "C:/Users/mcamock/Desktop/Advanced DBMS"
> setwd("C:/Users/mcamock/Desktop/Advanced DBMS")
> #Check the structure
> str(mba_data)
'data.frame':   50 obs. of  2 variables:
 $ Customer_Id: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Products   : chr  "laptop, mouse , headphones, pendrive, speakers" "laptop, headpho
nes" "laptop, mouse, pendrive" "mouse, speakers" ...
> View(mba_data)
> View(mba_data)
> #Ensure correct column names (case-sensitive)
> colnames(mba_data) <- c("Customer_id", "Products")
> # Split 'Products' into individual items
> trans_list <- strsplit(mba_data$Products,",") #Split by comma
> trans_list <- lapply(trans_list, trimws)      #remove leading/trained
> # Assign customer IDs as transaction names
> names(trans_list) <- mba_data$Customer_id
> # Convert list to 'transactions' object
> trans <- as(trans_list, "transactions")
> #inspect the transactions
> inspect(head(trans, 5))
    items                                          transactionID
[1] {headphones, laptop, mouse, pendrive, speakers} 1
[2] {headphones, laptop}                            2
[3] {laptop, mouse, pendrive}                       3
[4] {mouse, speakers}                               4
[5] {laptop, pendrive}                              5
> itemLabels(trans)
[1] "headphones" "laptop"     "mouse"      "pendrive"   "speakers"
> summary(trans)
transactions as itemMatrix in sparse format with
 50 rows (elements/itemsets/transactions) and
 5 columns (items) and a density of 0.588

most frequent items:
    laptop      mouse   pendrive headphones   speakers    (Other)
        42         31         31         23         20          0

element (itemset/transaction) length distribution:
sizes
 2  3  5
27 11 12

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    2.00    2.00    2.94    3.00    5.00

includes extended item information - examples:
      labels
1 headphones
2     laptop
3      mouse

includes extended transaction information - examples:
  transactionID
1             1
2             2
3             3
```

```
> #plot item frequencies
> itemFrequencyPlot(trans,topN=10,type="absolute",main="Top 10 most frequent items")
> #generate Apriori rules
> rules <- apriori(
+   trans,
+   parameter = list(support = 0.05, confidence = 0.03, minlen = 2, maxlen=3)
+ )
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
       0.03    0.1    1 none FALSE            TRUE       5    0.05      2      3
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 2
> #check rule summary
> summary(rules)
set of 50 rules

rule length distribution (lhs + rhs):sizes
 2  3
20 30

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    2.0     2.0     3.0     2.6     3.0     3.0

summary of quality measures:
    support         confidence        coverage           lift
 Min.   :0.2400   Min.   :0.2857   Min.   :0.2400   Min.   :0.7143
 1st Qu.:0.2400   1st Qu.:0.5217   1st Qu.:0.2400   1st Qu.:0.9677
 Median :0.2400   Median :0.6916   Median :0.4600   Median :1.1905
 Mean   :0.3012   Mean   :0.7297   Mean   :0.4512   Mean   :1.2709
 3rd Qu.:0.3600   3rd Qu.:1.0000   3rd Qu.:0.6200   3rd Qu.:1.6129
 Max.   :0.6200   Max.   :1.0000   Max.   :0.8400   Max.   :2.5000
     count
 Min.   :12.00
 1st Qu.:12.00
 Median :12.00
 Mean   :15.06
 3rd Qu.:18.00
 Max.   :31.00

> #Inspect top rules by lift
> inspect(head(sort(rules,by="lift"),10))
     lhs                      rhs          support confidence coverage lift
[1]  {headphones, mouse}    => {speakers}   0.24    1.0000000  0.24     2.500000
[2]  {headphones, pendrive} => {speakers}   0.24    1.0000000  0.24     2.500000
[3]  {pendrive, speakers}   => {headphones} 0.24    1.0000000  0.24     2.173913
[4]  {laptop, speakers}     => {headphones} 0.24    1.0000000  0.24     2.173913
[5]  {speakers}             => {mouse}      0.40    1.0000000  0.40     1.612903
[6]  {mouse}                => {speakers}   0.40    0.6451613  0.62     1.612903
[7]  {headphones, speakers} => {mouse}      0.24    1.0000000  0.24     1.612903
[8]  {headphones, speakers} => {pendrive}   0.24    1.0000000  0.24     1.612903
[9]  {pendrive, speakers}   => {mouse}      0.24    1.0000000  0.24     1.612903
[10] {laptop, speakers}     => {mouse}      0.24    1.0000000  0.24     1.612903
     count
[1]  12
[2]  12
[3]  12
[4]  12
[5]  20
[6]  20
[7]  12
[8]  12
[9]  12
[10] 12
> #visualize rules (interactive graph)
> plot(rules,method="graph",engine="htmlwidget")
> #optional: Export rules to CSV
> rules_df <- as(rules,"data.frame")
> write.csv(rules_df,"aprior_rules_output.csv", row.names = FALSE)
> #print final countdown
> cat("Apriori analysis complete! Rules saved to 'apriori_rules_output.csv'\n")
Apriori analysis complete! Rules saved to 'apriori_rules_output.csv'
>
```

	Customer_id	Products
1	1	laptop, mouse , headphones, pendrive, speakers
2	2	laptop, headphones
3	3	laptop, mouse, pendrive
4	4	mouse, speakers
5	5	laptop, pendrive
6	6	laptop, mouse , headphones, pendrive, speakers
7	7	laptop, headphones
8	8	laptop, mouse, pendrive
9	9	mouse, speakers

Files Plots Packages Help Viewer Presentation

Zoom Export



	A	B	C	D	E	F
1	rules	support	confidenc	coverage	lift	count
2	{speakers} => {headphones}	0.24	0.6	0.4	1.304348	12
3	{headphones} => {speakers}	0.24	0.521739	0.46	1.304348	12
4	{speakers} => {mouse}	0.4	1	0.4	1.612903	20
5	{mouse} => {speakers}	0.4	0.645161	0.62	1.612903	20
6	{speakers} => {pendrive}	0.24	0.6	0.4	0.967742	12
7	{pendrive} => {speakers}	0.24	0.387097	0.62	0.967742	12
8	{speakers} => {laptop}	0.24	0.6	0.4	0.714286	12

12.2) K means Algorithm

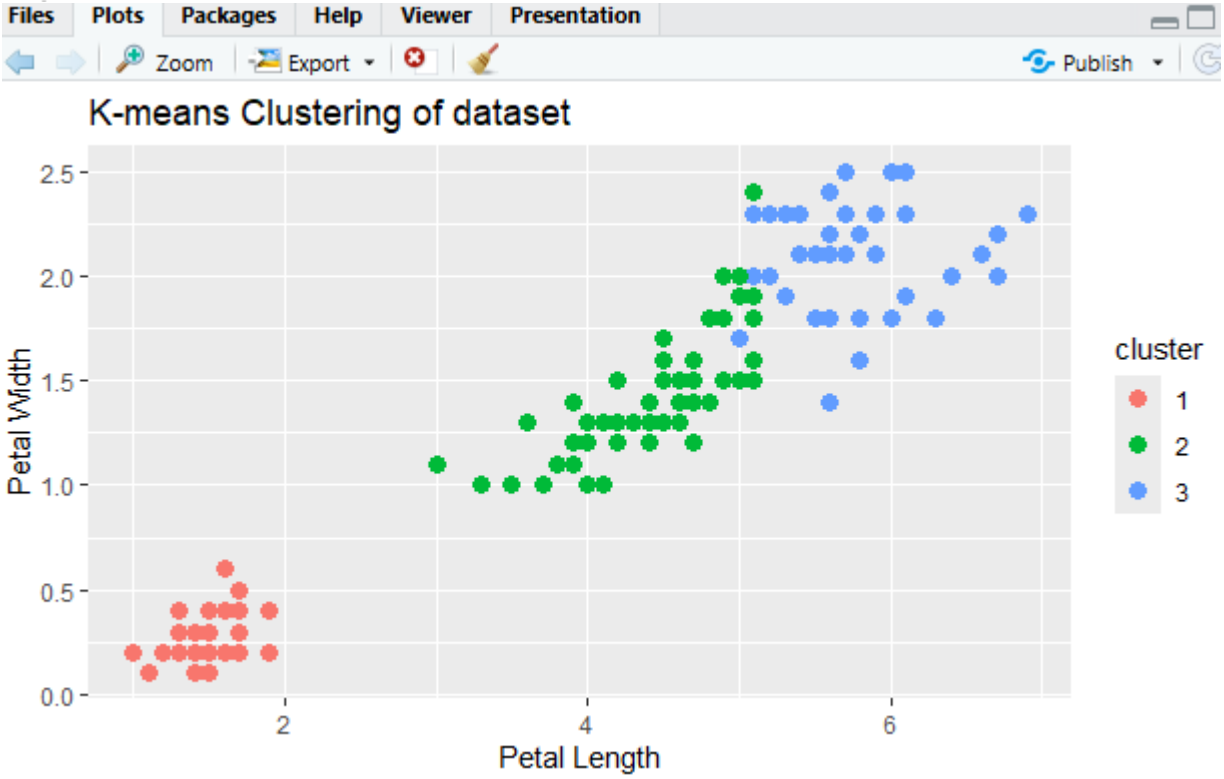
Source Code:-
K means.R

```
install.packages("ggplot2")
install.packages("dplyr")

#Load necessary libraries
library(ggplot2)
library(dplyr)
#Select only numeric columns (1 to 4) from the iris dataset
mydata <- select(iris, c(1,2,3,4))
#Apply k-means clustering with 3 clusters
model <- kmeans(mydata, 3)
model
#Display number of items in each cluster
model$size
#Compare actual species with cluster assignments
table(model$cluster, iris$Species)
#Convert cluster number to factor for plotting
iris$cluster <- as.factor(model$cluster)
#Visualize clusters by petal dimensions
ggplot(iris, aes(Petal.Length, Petal.Width, color = cluster)) +
  geom_point(size = 3) +
  labs(title = "K-means Clustering of dataset",
       x = "Petal Length",
       y = "Petal Width")
```

Output :-

```
> #Select only numeric columns (1 to 4) from the iris dataset
> mydata <- select(iris, c(1,2,3,4))
> #Apply k-means clustering with 3 clusters
> model <- kmeans(mydata, 3)
> model
K-means clustering with 3 clusters of sizes 50, 62, 38

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

Clustering vector:
  [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
 [40] 1 1 1 1 1 1 1 1 1 1 1 2 2 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 3
 [79] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3
[118] 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

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

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
> #Displat number of items in each cluster
> model$size
[1] 50 62 38
```

```
> #Compare actual species with cluster assignments
> table(model$cluster, iris$Species)

    setosa versicolor virginica
  1     50          0         0
  2      0         48        14
  3      0          2        36
> #Convert cluster number to factor for plotting
> iris$cluster <- as.factor(model$cluster)
> #Visualize clusters by petal dimensions
> ggplot(iris, aes(Petal.Length, Petal.Width, color = cluster)) +
+   geom_point(size = 3) +
+   labs(title = "K-means Clustering of dataset",
+        x = "Petal Length",
+        y = "Petal Width")
```



12.3)Agglomerative

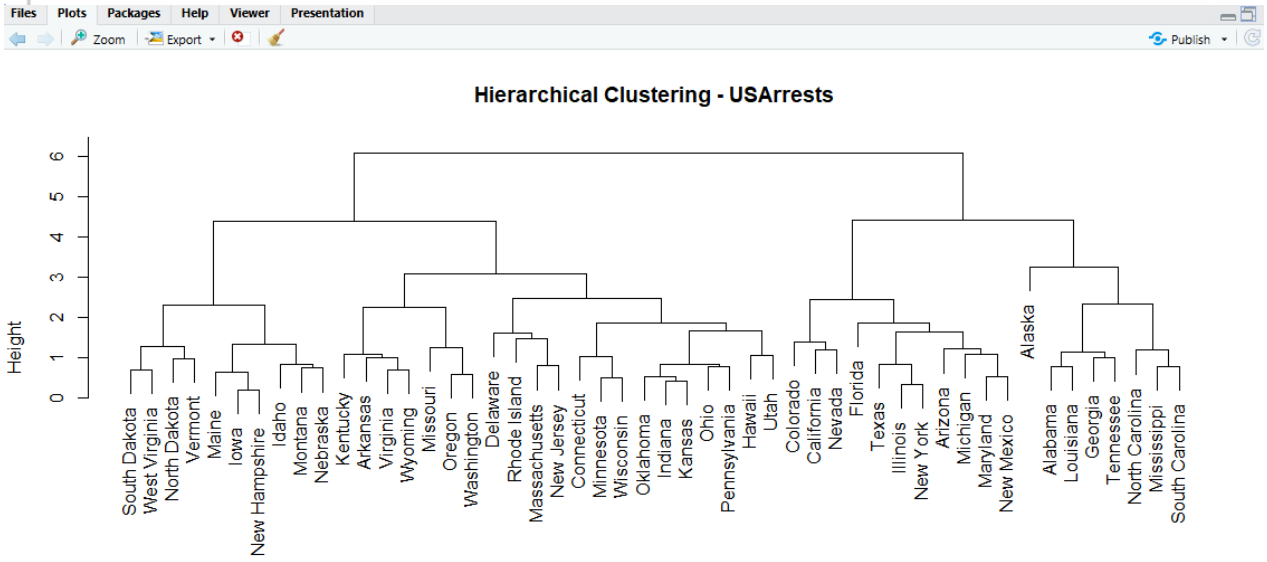
Source Code :-

Agglomerative-hierarchical.R

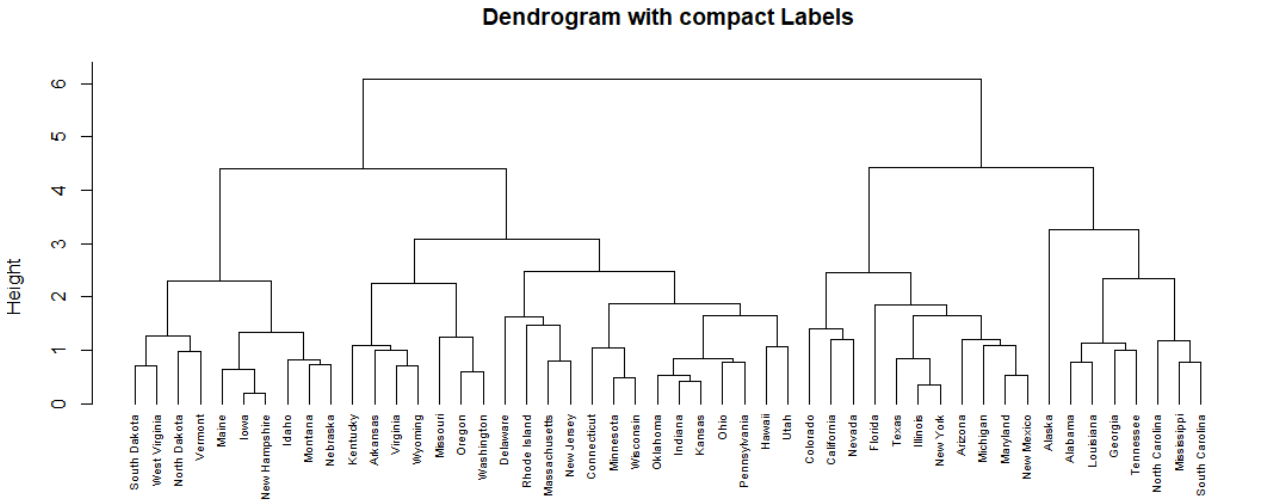
```
df <- USArrests
View(df)
df <- na.omit(df)
df
d <- scale(df)
head(d)
dist_matrix <- dist(d, method = "euclidean")
hc <- hclust(dist_matrix, method="complete")
plot(hc,
    main = "Hierarchical Clustering - USArrests",
    xlab = "",
    sub = "")
plot(hc,
    cex = 0.6,    #reduce label size
    hang =-1,     #Align leaves at the same height
    main = "Dendrogram with compact Labels")

hcd <- as.dendrogram(hc)
plot(hcd,
    type = "triangle",
    main = "Triangular Dendogram - USArrests")
groups <- cutree(hc,k = 4)
df$cluster <- groups
head(df)
plot(hc, cex = 0.6, hang = -1)
rect.hclust(hc, k = 4, border = "red")
cut_tree <- cut(hcd, h = 75)
str(cut_tree)
plot(cut_tree$upper, main = "Upper Dendogram (Top cluster structure)")
plot(cut_tree$lower[[2]], main = "Detailed view of cluster 2")
```

Output :

```
>
> df <- USArrests
> View(df)
> df <- na.omit(df)
> df
               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
Connecticut       3.3     110       77 11.1
Delaware          5.9     238       72 15.8
Florida          15.4     335       80 31.9
Georgia          17.4     211       60 25.8
Hawaii            5.3      46       83 20.2
> d <- scale(df)
> head(d)
              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
> dist_matrix <- dist(d, method = "euclidean")
> hc <- hclust(dist_matrix, method="complete")
> plot(hc,
+      main = "Hierarchical Clustering - USArrests",
+      xlab = "",
+      sub = "")
```



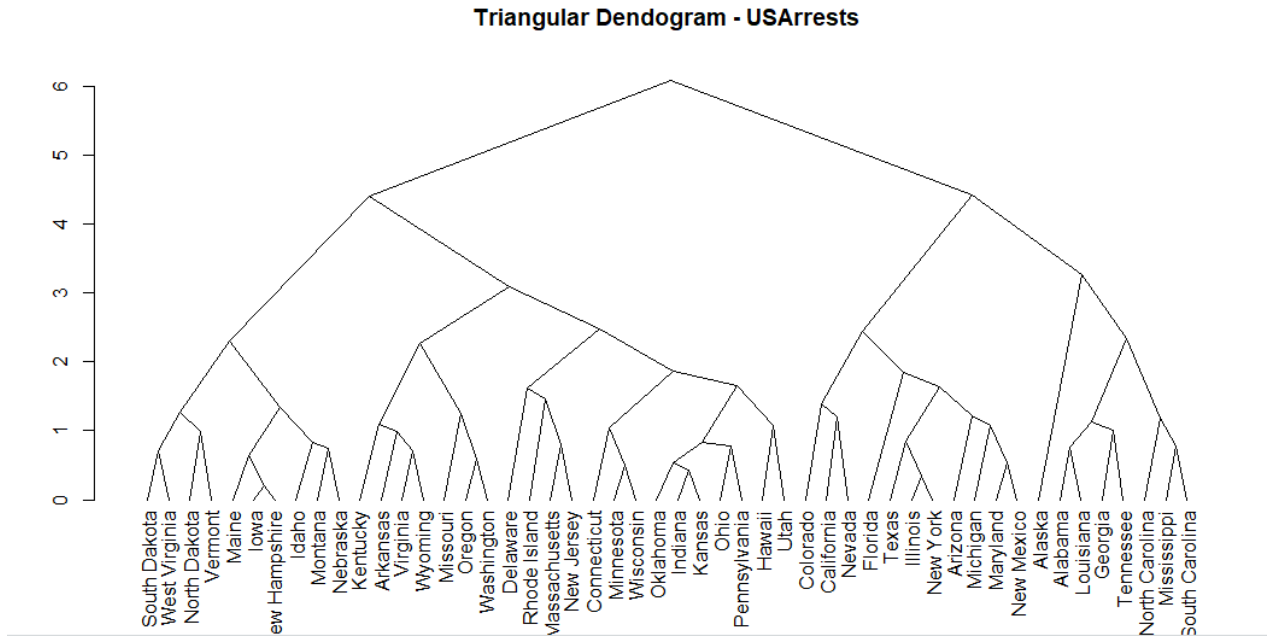
```
> plot(hc,
+      cex = 0.6,     #reduce label size
+      hang =-1,      #Align leaves at the same height
+      main = "Dendrogram with compact Labels")
>
```



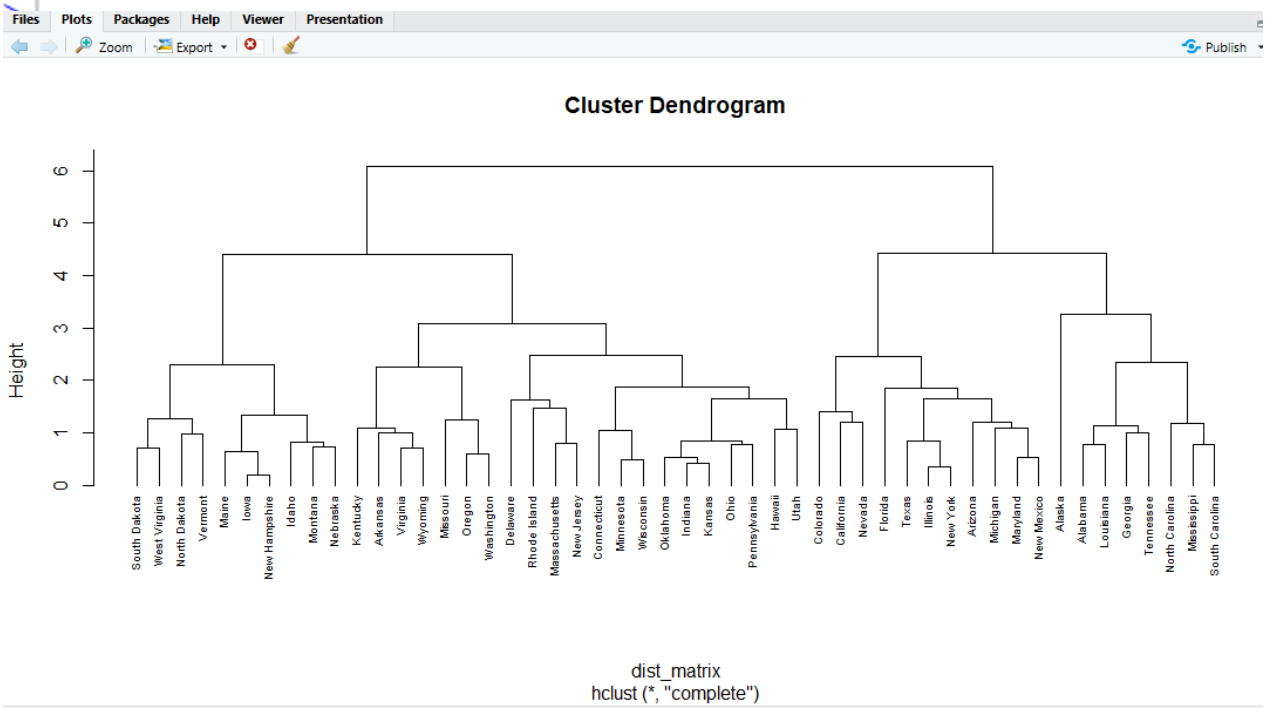

```
> hcd <- as.dendrogram(hc)
> plot(hcd,
+      type = "triangle",
+      main = "Triangular Dendogram - USArrests")
>
```

Files | Plots | Packages | Help | Viewer | Presentation

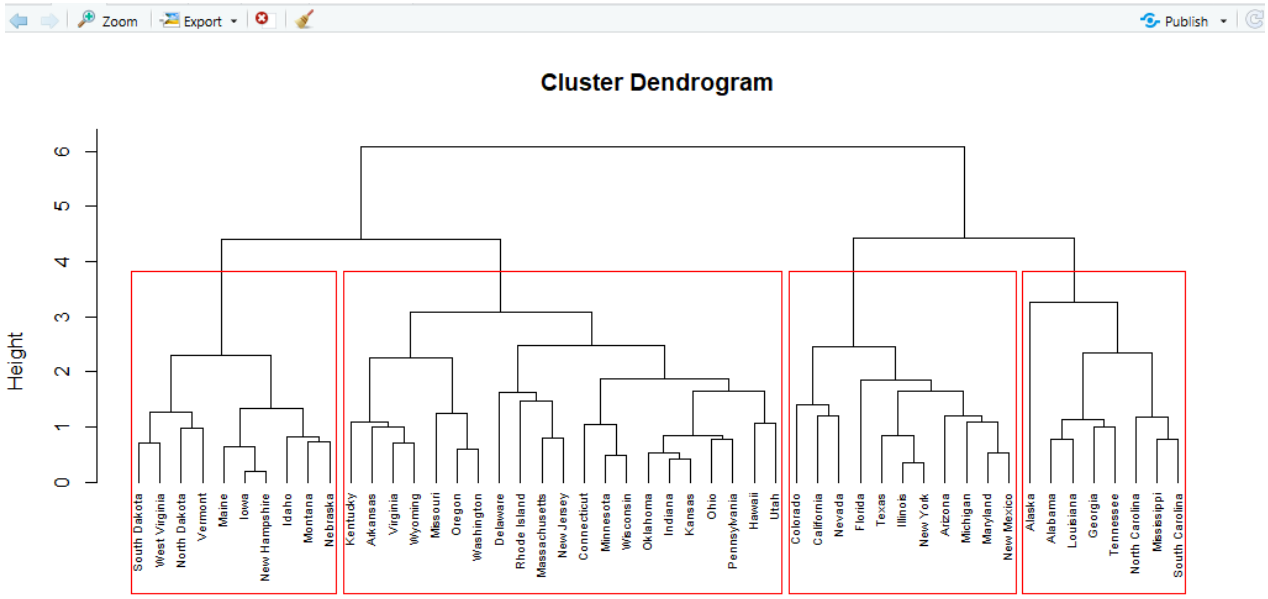
Zoom | Export | Publish



```
> groups <- cutree(hc,k = 4)
> df$cluster <- groups
> head(df)
           Murder Assault UrbanPop Rape cluster
Alabama      13.2     236       58 21.2       1
Alaska       10.0     263       48 44.5       1
Arizona       8.1     294       80 31.0       2
Arkansas      8.8     190       50 19.5       3
California    9.0     276       91 40.6       2
Colorado      7.9     204       78 38.7       2
> plot(hc, cex = 0.6, hang = -1)
```

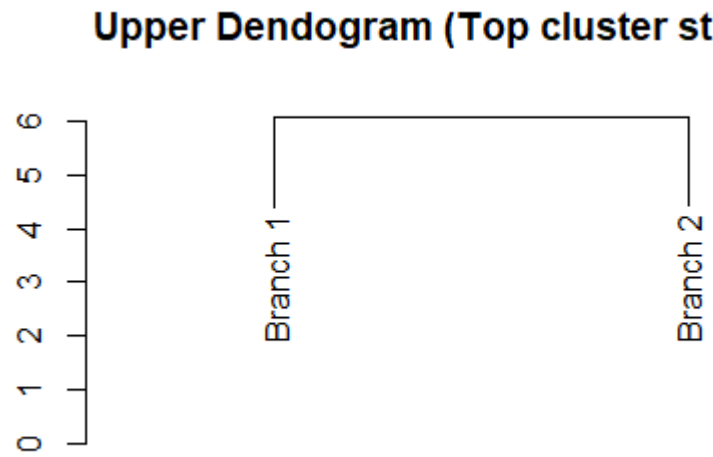


```
> rect.hclust(hc, k = 4, border = "red")
```




```
> cut_tree <- cut(hcd, h = 75)
> str(cut_tree)
List of 2
 $ upper:  ..--[dendrogram w/ 2 branches and 2 members at h = 6.08, midpoint = 24.7, x.member = 50]
  ..   |--leaf "Branch 1" (h= 4.4 midpoint = 10.5, x.member = 31 )
  ..   `--leaf "Branch 2" (h= 4.42 midpoint = 7.88, x.member = 19 )
 $ lower:List of 2
  ..$ :  .. ..--[dendrogram w/ 2 branches and 31 members at h = 4.4, midpoint = 10.5]
  .. ..  |--[dendrogram w/ 2 branches and 10 members at h = 2.3, midpoint = 3.88]
  .. ..  |  |--[dendrogram w/ 2 branches and 4 members at h = 1.27, midpoint = 1.5]
  .. ..  |  |  |--[dendrogram w/ 2 branches and 2 members at h = 0.711, midpoint = 0.5]
  .. ..  |  |  |  |--leaf "South Dakota"
  .. ..  |  |  |  `--leaf "West Virginia"
  .. ..  |  |  `--[dendrogram w/ 2 branches and 2 members at h = 0.982, midpoint = 0.5]
  .. ..  |  |     |--leaf "North Dakota"
  .. ..  |  |     `--leaf "Vermont"
  .. ..  |  `--[dendrogram w/ 2 branches and 6 members at h = 1.33, midpoint = 2.25]
  .. ..  |     |--[dendrogram w/ 2 branches and 3 members at h = 0.646, midpoint = 0.75]
  .. ..  |     |  |--leaf "Maine"
  .. ..  |     |  `--[dendrogram w/ 2 branches and 2 members at h = 0.206, midpoint = 0.5]
  .. ..  |     |     |--leaf "Iowa"
  .. ..  |     |     `--leaf "New Hampshire"
  .. ..  |     `--[dendrogram w/ 2 branches and 3 members at h = 0.829, midpoint = 0.75]
  .. ..  |        |--leaf "Idaho"
  .. ..  |        `--[dendrogram w/ 2 branches and 2 members at h = 0.739, midpoint = 0.5]
  .. ..  |           |--leaf "Montana"
  .. ..  |           `--leaf "Nebraska"
  .. ..  `--[dendrogram w/ 2 branches and 21 members at h = 3.09, midpoint = 7.07]
  .. ..     |--[dendrogram w/ 2 branches and 7 members at h = 2.26, midpoint = 2.81]
  .. ..     |  |--[dendrogram w/ 2 branches and 4 members at h = 1.09, midpoint = 0.875]
```

```
plot(cut_tree$upper, main = "Upper Dendogram (Top cluster structure)")
```




```
> plot(cut_tree$lower[[2]], main = "Detailed view of cluster 2")
```