1. Agglomerative clustering – Cars Dataset

# Load the data

data("cars")

# Standardize the data

df<- scale(cars)

# Show the first 6 rows

head(df, nrow = 6)

# Compute the dissimilarity matrix

# df = the standardized data

res.dist<- dist(df, method = "euclidean")

#To see easily the distance information between objects, we reformat the

#results of the function dist() into a matrix using the as.matrix() function.

#In this matrix, value in the cell formed by the row i, the column j, represents

#the distance between object i and object j in the original data set. For instance,

#element 1,1 represents the distance between object 1 and itself (which is zero).

#Element 1,2 represents the distance between object 1 and object 2, and so on.

#Displays the first 6 rows and columns of the distance matrix:

as.matrix(res.dist)[1:6, 1:6]

#creates hierarchial tree

res.hc<- hclust(d = res.dist, method = "ward.D2")

#for creating dendrogram

# cex: label size

library("factoextra")

fviz\_dend(res.hc, cex = 0.5)

# Please refer notes section

# Compute cophentic distance

#Thecophenetic distance between two objects is the height of the

#dendrogram where the two branches that include the two objects merge

#into a single branch.

res.coph<- cophenetic(res.hc)

# Correlation between cophenetic distance and

# the original distance

cor(res.dist, res.coph)

# Cut tree into 4 groups

grp<- cutree(res.hc, k = 4)

head(grp, n = 4)

# Number of members in each cluster

table(grp)

# Get the names for the members of cluster 1

rownames(df)[grp == 1]

fviz\_dend(grp, n=4)

#-------------------------

library("cluster")

# Agglomerative Nesting (Hierarchical Clustering)

res.agnes<- agnes(x = USArrests, # data matrix

stand = TRUE, # Standardize the data

metric = "euclidean", # metric for distance matrix

method = "ward" # Linkage method

)

# DIvisiveANAlysis Clustering

res.diana<- diana(x = USArrests, # data matrix

stand = TRUE, # standardize the data

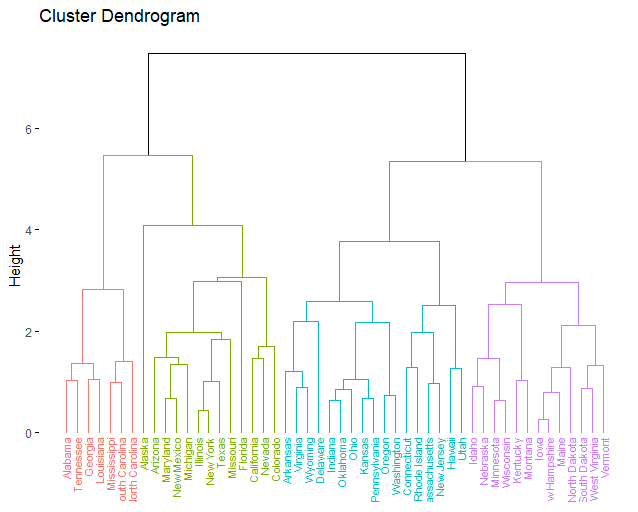
metric = "euclidean" # metric for distance matrix

)

fviz\_dend(res.agnes, cex = 0.6, k = 4)

fviz\_dend(res.diana, cex = 0.6, k = 4)

**OUTPUT:**



|  |
| --- |
| # Load the data  > data("cars")  >  > # Standardize the data  > df<- scale(cars)  >  >  > # Show the first 6 rows  > head(df, nrow = 6)  speed dist  [1,] -2.155969 -1.5902596  [2,] -2.155969 -1.2798136  [3,] -1.588609 -1.5126481  [4,] -1.588609 -0.8141446  [5,] -1.399489 -1.0469791  [6,] -1.210369 -1.2798136  >  > # Compute the dissimilarity matrix  > # df = the standardized data  > res.dist<- dist(df, method = "euclidean")  >  >  > #To see easily the distance information between objects, we reformat the  > #results of the function dist() into a matrix using the as.matrix() function.  > #In this matrix, value in the cell formed by the row i, the column j, represents  > #the distance between object i and object j in the original data set. For instance,  > #element 1,1 represents the distance between object 1 and itself (which is zero).  > #Element 1,2 represents the distance between object 1 and object 2, and so on.  >  > #Displays the first 6 rows and columns of the distance matrix:  > as.matrix(res.dist)[1:6, 1:6]  1 2 3 4 5 6  1 0.0000000 0.3104460 0.5726442 0.9613804 0.9313520 0.9952574  2 0.3104460 0.0000000 0.6132778 0.7339928 0.7915015 0.9456006  3 0.5726442 0.6132778 0.0000000 0.6985035 0.5026072 0.4441594  4 0.9613804 0.7339928 0.6985035 0.0000000 0.2999639 0.5999277  5 0.9313520 0.7915015 0.5026072 0.2999639 0.0000000 0.2999639  6 0.9952574 0.9456006 0.4441594 0.5999277 0.2999639 0.0000000  >  > #creates hierarchial tree  > res.hc<- hclust(d = res.dist, method = "ward.D2")  >  > #for creating dendrogram  > # cex: label size  > library("factoextra")  > fviz\_dend(res.hc, cex = 0.5)  >  > # Please refer notes section  > # Compute cophentic distance  > #Thecophenetic distance between two objects is the height of the  > #dendrogram where the two branches that include the two objects merge  > #into a single branch.  > res.coph<- cophenetic(res.hc)  >  > # Correlation between cophenetic distance and  > # the original distance  > cor(res.dist, res.coph)  [1] 0.6158925  >  > # Cut tree into 4 groups  > grp<- cutree(res.hc, k = 4)  > head(grp, n = 4)  [1] 1 1 1 1  >  > # Number of members in each cluster  > table(grp)  grp  1 2 3 4  6 17 23 4  >  > # Get the names for the members of cluster 1  > rownames(df)[grp == 1]  NULL  >  > fviz\_dend(grp, n=4)  Error in fviz\_dend(grp, n = 4) : Can't handle an object of class integer  > #-------------------------  >  >  > library("cluster")  > # Agglomerative Nesting (Hierarchical Clustering)  > res.agnes<- agnes(x = USArrests, # data matrix  + stand = TRUE, # Standardize the data  + metric = "euclidean", # metric for distance matrix  + method = "ward" # Linkage method  + )  >  >  > # DIvisiveANAlysis Clustering  > res.diana<- diana(x = USArrests, # data matrix  + stand = TRUE, # standardize the data  + metric = "euclidean" # metric for distance matrix  + )  >  > fviz\_dend(res.agnes, cex = 0.6, k = 4)  > fviz\_dend(res.diana, cex = 0.6, k = 4) |
|  |
| |  | | --- | |  | |

1. Kmeans clustering – IRIS Dataset

library(dplyr)

PATH <-"https://raw.githubusercontent.com/guru99-edu/R-Programming/master/computers.csv"

df <- read.csv(PATH) %>%

select(-c(X, cd, multi, premium))

glimpse(df)

summary(df)

rescale\_df <- df % > %

mutate(price\_scal = scale(price),

hd\_scal = scale(hd),

ram\_scal = scale(ram),

screen\_scal = scale(screen),

ads\_scal = scale(ads),

trend\_scal = scale(trend)) % > %

select(-c(price, speed, hd, ram, screen, ads, trend))

#R base has a function to run the k mean algorithm. The basic function of k mean is:

kmeans(df, k)

#Train the model

install.packages("animation")

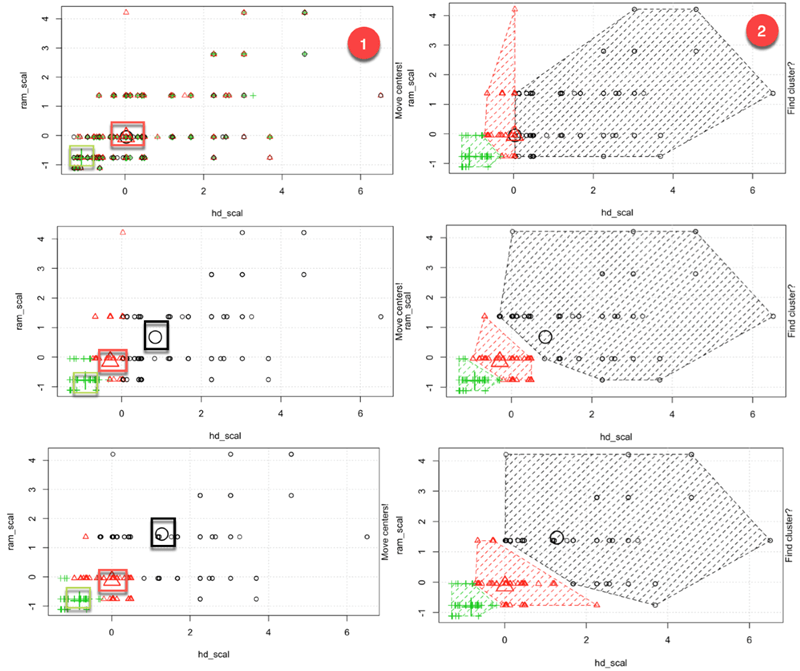
set.seed(2345)

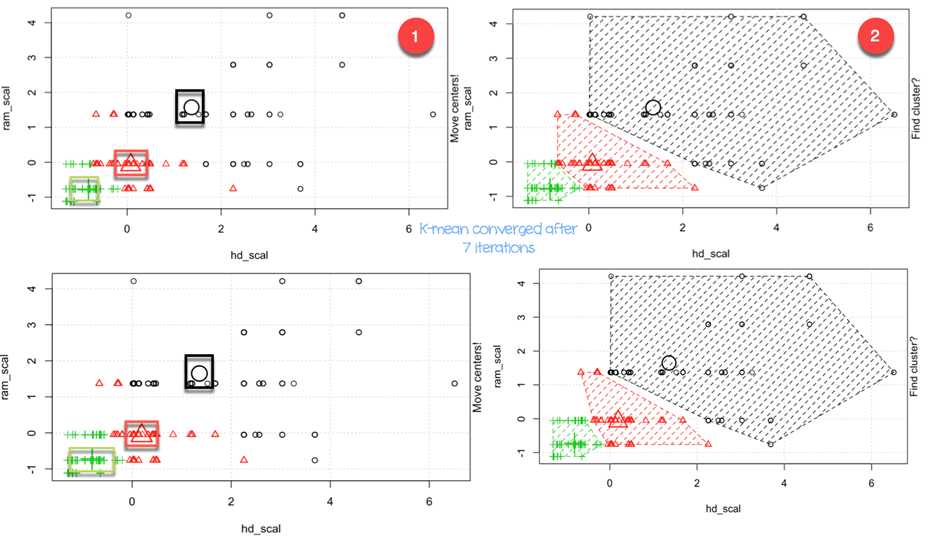
library(animation)

kmeans.ani(rescale\_df[2:3], 3)

pc\_cluster <-kmeans(rescale\_df, 5)

#OUTPUT:





|  |
| --- |
| library(dplyr)  > PATH <-"https://raw.githubusercontent.com/guru99-edu/R-Programming/master/computers.csv"  > df <- read.csv(PATH) %>%  + select(-c(X, cd, multi, premium))  > glimpse(df)  Observations: 6,259  Variables: 7  $ price *<int>* 1499, 1795, 1595, 1849, 3295, 3695, 1720, 1995, 2225, 2575, 2195, 2605, 2045, 2295, 2...  $ speed *<int>* 25, 33, 25, 25, 33, 66, 25, 50, 50, 50, 33, 66, 50, 25, 50, 50, 33, 33, 33, 66, 33, 6...  $ hd *<int>* 80, 85, 170, 170, 340, 340, 170, 85, 210, 210, 170, 210, 130, 245, 212, 130, 85, 210,...  $ ram *<int>* 4, 2, 4, 8, 16, 16, 4, 2, 8, 4, 8, 8, 4, 8, 8, 4, 2, 4, 4, 8, 4, 4, 16, 4, 8, 2, 4, 8...  $ screen *<int>* 14, 14, 15, 14, 14, 14, 14, 14, 14, 15, 15, 14, 14, 14, 14, 14, 14, 15, 15, 14, 14, 1...  $ ads *<int>* 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 94, 9...  $ trend *<int>* 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...  >  > summary(df)  price speed hd ram screen ads  Min. : 949 Min. : 25.00 Min. : 80.0 Min. : 2.000 Min. :14.00 Min. : 39.0  1st Qu.:1794 1st Qu.: 33.00 1st Qu.: 214.0 1st Qu.: 4.000 1st Qu.:14.00 1st Qu.:162.5  Median :2144 Median : 50.00 Median : 340.0 Median : 8.000 Median :14.00 Median :246.0  Mean :2220 Mean : 52.01 Mean : 416.6 Mean : 8.287 Mean :14.61 Mean :221.3  3rd Qu.:2595 3rd Qu.: 66.00 3rd Qu.: 528.0 3rd Qu.: 8.000 3rd Qu.:15.00 3rd Qu.:275.0  Max. :5399 Max. :100.00 Max. :2100.0 Max. :32.000 Max. :17.00 Max. :339.0  trend  Min. : 1.00  1st Qu.:10.00  Median :16.00  Mean :15.93  3rd Qu.:21.50  Max. :35.00 |
|  |
| |  | | --- | | > | |

1. DBSCAN – Mall\_Customer Dataset

#DBSCAN Code in R

library(fpc)

library(dbscan)

library(factoextra)

#install.packages("fpc")

#install.packages("dbscan")

#install.packages("factoextra")

# Load the data

data("Mall\_Customers", package = "factoextra")

df <- multishapes[, 1:2]

#Apply k-means clustering on multishapes dataset and see the graph

set.seed(123)

km.res <- kmeans(df, 5, nstart = 25)

fviz\_cluster(km.res, df, geom = "point",

ellipse= FALSE, show.clust.cent = FALSE,

palette = "jco", ggtheme = theme\_classic())

#there are 5 five clusters in the data, but it can be seen that

#k-means method inaccurately identify the 5 clusters.

#---------------------------------------------

#Now apply dbscan on same dataset and see the clusters

## Compute DBSCAN using fpc package

library("fpc")

set.seed(123)

db <- fpc::dbscan(df, eps = 0.15, MinPts = 5)

# Plot DBSCAN results

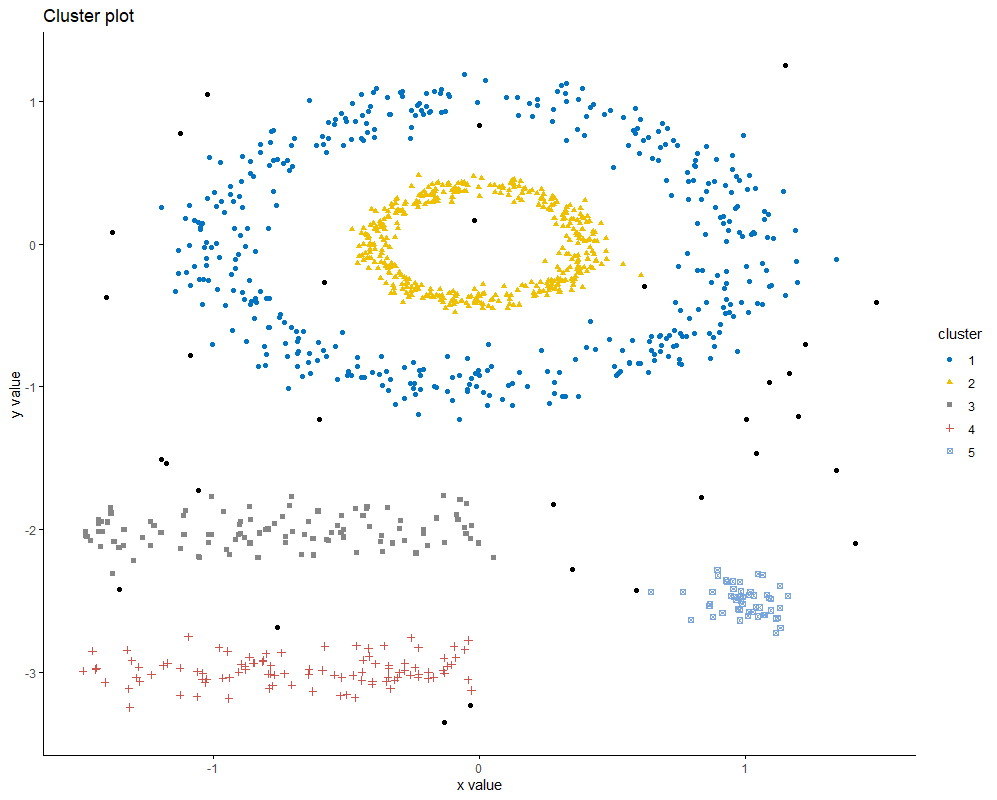
library("factoextra")

fviz\_cluster(db, data = df, stand = FALSE,

ellipse = FALSE, show.clust.cent = FALSE,

geom = "point",palette = "jco", ggtheme = theme\_classic())

#OUTPUT:



|  |
| --- |
| #DBSCAN Code in R  > library(fpc)  > library(dbscan)  > library(factoextra)  > #install.packages("fpc")  > #install.packages("dbscan")  > #install.packages("factoextra")  > # Load the data  > data("Mall\_Customers", package = "factoextra")  Warning message:  In data("Mall\_Customers", package = "factoextra") :  data set ‘Mall\_Customers’ not found  > df <- multishapes[, 1:2]  >  > #Apply k-means clustering on multishapes dataset and see the graph  > set.seed(123)  > km.res <- kmeans(df, 5, nstart = 25)  > fviz\_cluster(km.res, df, geom = "point",  + ellipse= FALSE, show.clust.cent = FALSE,  + palette = "jco", ggtheme = theme\_classic())  > #there are 5 five clusters in the data, but it can be seen that  > #k-means method inaccurately identify the 5 clusters.  >  > #---------------------------------------------  > #Now apply dbscan on same dataset and see the clusters  >  > ## Compute DBSCAN using fpc package  > library("fpc")  > set.seed(123)  > db <- fpc::dbscan(df, eps = 0.15, MinPts = 5)  >  > # Plot DBSCAN results  > library("factoextra")  > fviz\_cluster(db, data = df, stand = FALSE,  + ellipse = FALSE, show.clust.cent = FALSE,  + geom = "point",palette = "jco", ggtheme = theme\_classic()) |
|  |
| |  | | --- | | > | |