Cluster Analysis using R

Cluster analysis or **clustering** is the task of assigning a set of objects into groups (called **clusters**) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. [1]

K-means clustering

This is a prototype-based, partitional clustering technique that attempts to find a user-specified number of clusters (k), which are presented by their centroids. [2]

K-means clustering is based on partitional clustering approach. A partitional clustering is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset. In K-means clustering, each cluster is associated with a centroid (center point) and each point is assigned to the cluster with the closest centroid. In this type of clustering, number of clusters, denoted by K, must be specified.

K-means clustering using R

In R, the function kmeans() performs k-means clustering on a data matrix.[3]

Usage

kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))

Arguments

x	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
centers	either the number of clusters, say *k*, or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in x is chosen as the initial centres.
iter.max	the maximum number of iterations allowed.
nstart	if centers is a number, how many random sets should be chosen?
algorithm	character: may be abbreviated.

Value

An object of class "kmeans" which has a print method and is a list with components: [3]

cluster	A vector of integers (from 1:k) indicating the cluster to which each point is allocated.
centers	A matrix of cluster centres.
withinss	The within-cluster sum of squares for each cluster.
totss	The total within-cluster sum of squares.
tot.withinss	Total within-cluster sum of squares, i.e., sum (withinss).
betweenss	The between-cluster sum of squares.
size	The number of points in each cluster.

<u>Example</u>

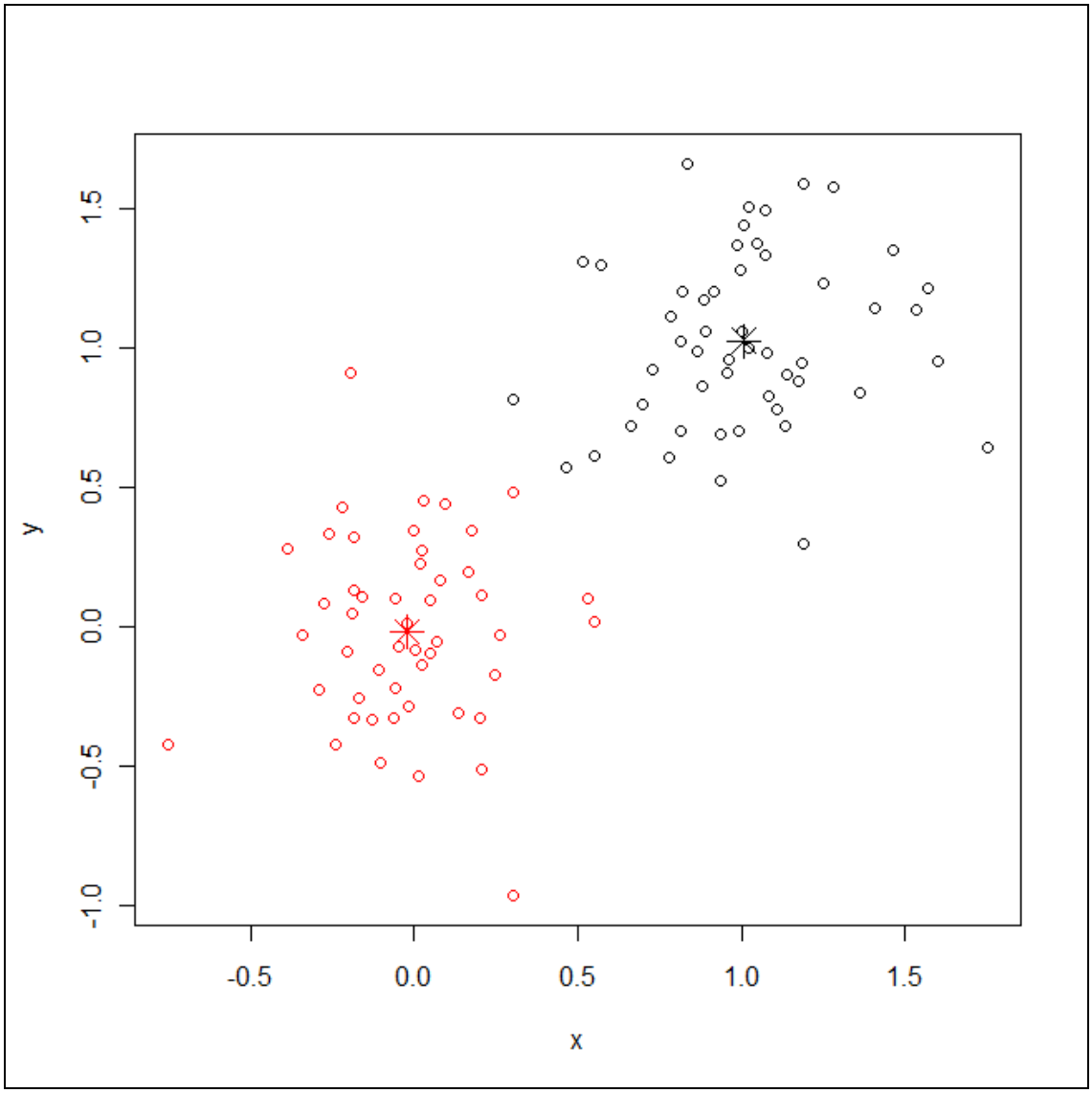
```
> # a 2-dimensional example
> x = rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
+           matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
> colnames(x) = c("x", "y")
> (cl = kmeans(x, 2))
```

```
K-means clustering with 2 clusters of sizes 51, 49
Cluster means:
          x          y
1  1.00553507  1.0245900
2 -0.02176654 -0.0187013
Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [75] 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
```

```
Within cluster sum of squares by cluster:
[1] 9.203628 7.976549
 (between_SS / total_SS =  75.7 %)
Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

> plot(x, col = cl$cluster)

> points(cl$centers, col = 1:2, pch = 8, cex=2)



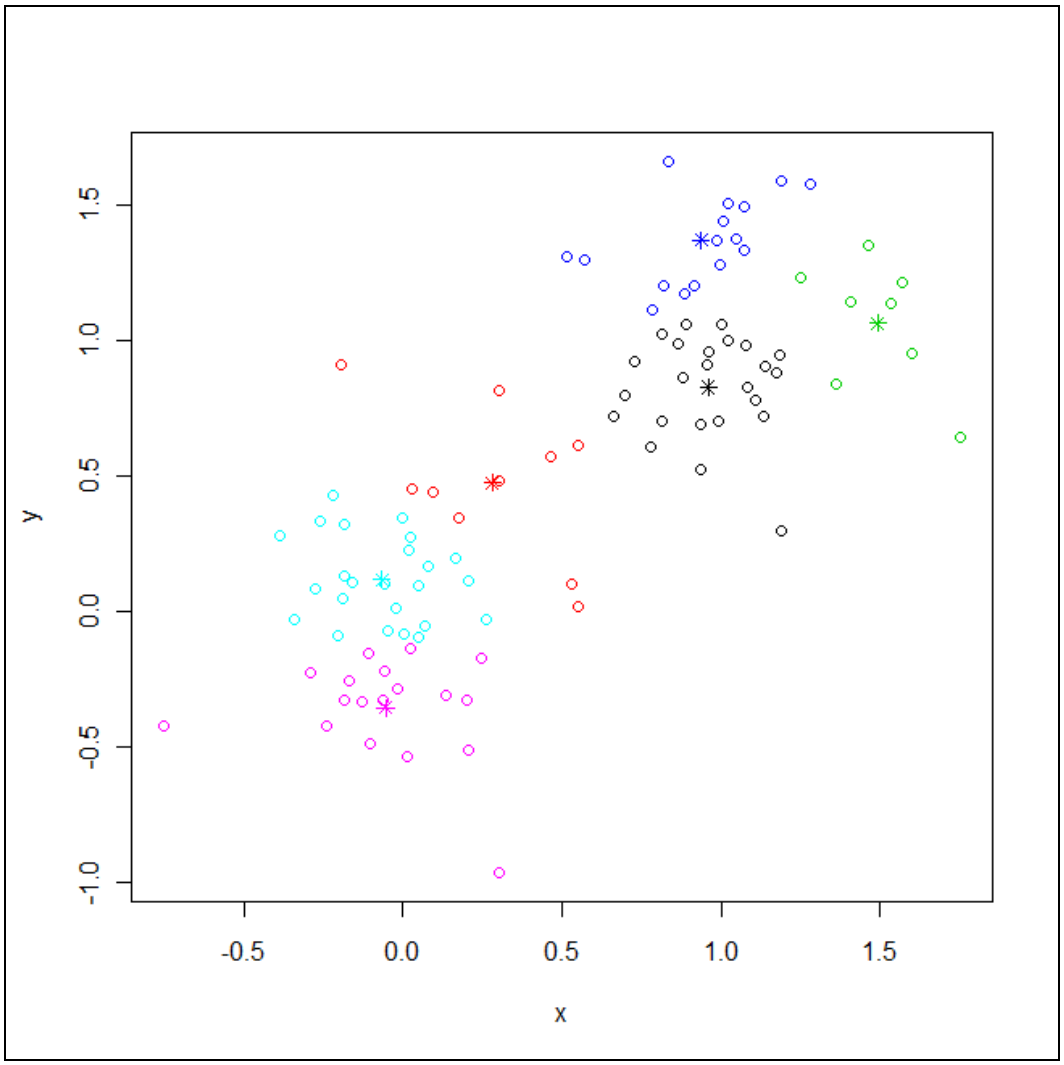
random starts with too many clusters

> (cl = kmeans(x, 6, nstart = 20))

```
K-means clustering with 6 clusters of sizes 24, 10, 8, 16, 24, 18

Cluster means:

          x         y

1  0.95990438  0.8266205

2  0.28148695  0.4738840

3  1.49251708  1.0630865

4  0.93662759  1.3693039

5 -0.06602533  0.1157600

6 -0.05435648 -0.3573222

Clustering vector:

 [1] 5 5 5 5 5 6 5 6 5 6 6 5 6 6 6 5 5 6 5 6 6 5 6 2 6 2 2 6 5 2 5 5 6 5 2 6 2

[38] 6 2 5 5 2 5 5 5 5 6 5 5 6 4 1 3 4 4 4 2 1 2 1 4 3 1 1 1 4 3 1 4 1 4 4 3 1

[75] 4 4 1 1 1 1 1 4 3 1 1 4 1 1 1 1 1 3 3 4 1 4 3 4 1 1

Within cluster sum of squares by cluster:

[1] 1.3609441 1.2832563 0.5552351 0.9997794 1.2523257 1.6469588

 (between_SS / total_SS =  90.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"

[6] "betweenss"    "size"
```

> plot(x, col = cl$cluster)

> points(cl$centers, col = 1:6, pch = 8)



Hierarchical Clustering

This type of clustering produces a set of nested clusters organized as a hierarchical tree. It can be visualized as a dendogram – a tree like diagram that records the sequences of merge or splits. [2]

There are two main types of hierarchical clustering:

Agglomerative – This type of clustering starts with the points as individual clusters. Then at each step, the closest pair of clusters are merged until only one cluster (or k clusters) left.

Divisive – This type of clustering starts with one, all-inclusive cluster. Then at each step, cluster is split until each cluster contains a point (or there are k clusters).

Hierarchical Clustering using R

In R, the function hclust () performs hierarchical clustering. [4]

Usage

```
hclust(d, method = "complete", members=NULL)

plot(x, labels = NULL, hang = 0.1,
    axes = TRUE, frame.plot = FALSE, ann = TRUE,
    main = "Cluster Dendrogram",
    sub = NULL, xlab = NULL, ylab = "Height", ...)

plclust(tree, hang = 0.1, unit = FALSE, level = FALSE, hmin = 0,
       square = TRUE, labels = NULL, plot. = TRUE,
       axes = TRUE, frame.plot = FALSE, ann = TRUE,
       main = "", sub = NULL, xlab = NULL, ylab = "Height")
```

Arguments

d	a dissimilarity structure as produced by dist.
method	the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid".
members	NULL or a vector with length size of d. See the ‘Details’ section.
x,tree	an object of the type produced by hclust.
hang	The fraction of the plot height by which labels should hang below the rest of the plot. A negative value will cause the labels to hang down from 0.
labels	A character vector of labels for the leaves of the tree. By default the row names or row numbers of the original data are used. If labels=FALSE no labels at all are plotted.
axes, frame.plot, ann	logical flags as in plot.default.
main, sub, xlab, ylab	character strings for title. sub and xlab have a non-NULL default when there's a tree$call.
...	Further graphical arguments.
unit	logical. If true, the splits are plotted at equally-spaced heights rather than at the height in the object.
hmin	numeric. All heights less than hmin are regarded as being hmin: this can be used to suppress detail at the bottom of the tree.
level, square, plot.	as yet unimplemented arguments of plclust for S-PLUS compatibility.

Value

An object of class **hclust** which describes the tree produced by the clustering process. The object is a list with components: [4]

merge	an *n-1* by 2 matrix. Row *i* of merge describes the merging of clusters at step *i* of the clustering. If an element *j* in the row is negative, then observation *-j* was merged at this stage. If *j* is positive then the merge was with the cluster formed at the (earlier) stage *j* of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons.
height	a set of *n-1* non-decreasing real values. The clustering *height*: that is, the value of the criterion associated with the clustering method for the particular agglomeration.
order	a vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches.
labels	labels for each of the objects being clustered.
call	the call which produced the result.
method	the cluster method that has been used.
dist.method	the distance that has been used to create d (only returned if the distance object has a "method" attribute).

Example

In the data set mtcars, at first distance matrix can be computed: [5]

```
> d = dist(as.matrix(mtcars))   # find distance matrix
```

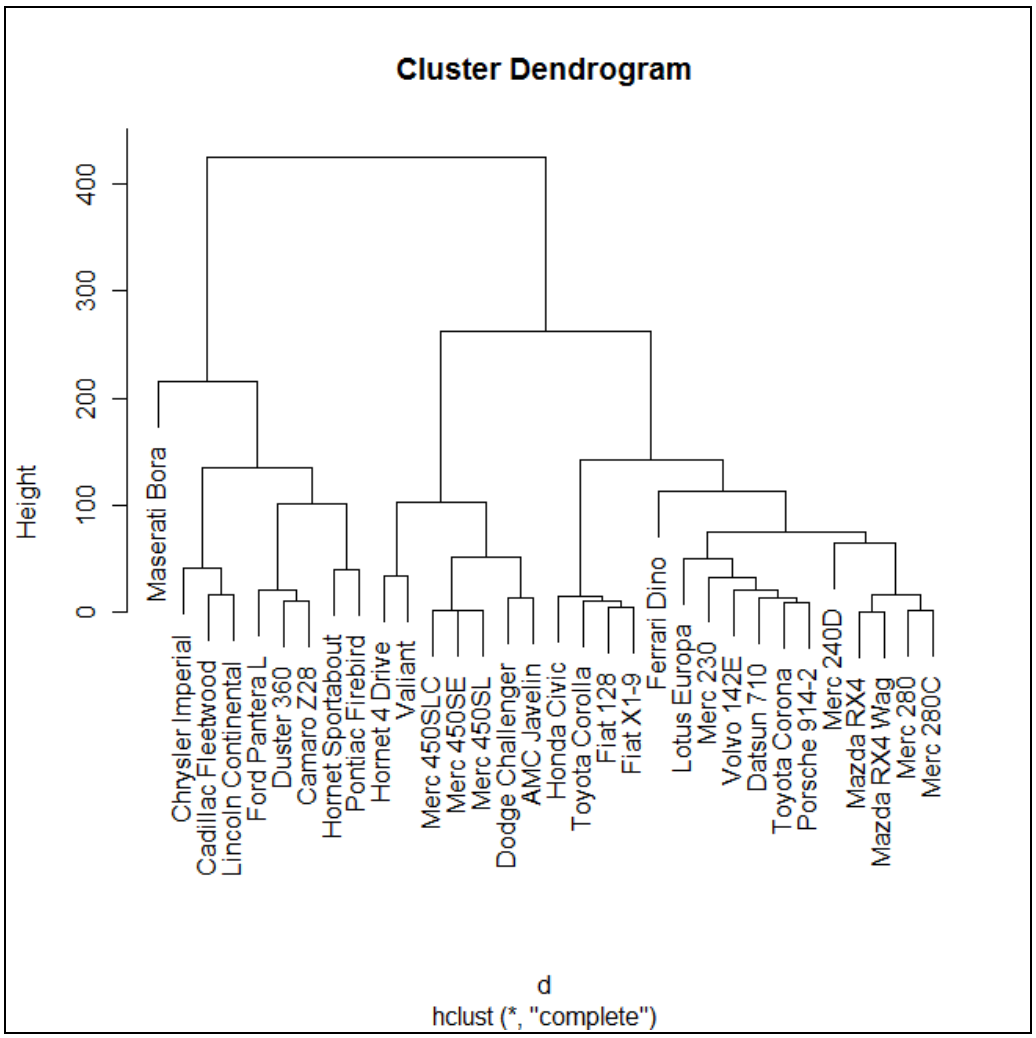
Then, distance matrix is run with hclust:

```
> hc = hclust(d)                # apply hirarchical clustering
```

```
Call:

hc = hclust(d)
```

Plot a dendogram that displays a hierarchical relationship among the vehicles:

```
> plot(hc)
```



rect.hclust()

Description

It draws rectangles around the branches of a dendrogram highlighting the corresponding clusters. First the dendrogram is cut at a certain level, then a rectangle is drawn around selected branches. [6]

Usage

```
rect.hclust(tree, k = NULL, which = NULL, x = NULL, h = NULL,
        border = 2, cluster = NULL)
```

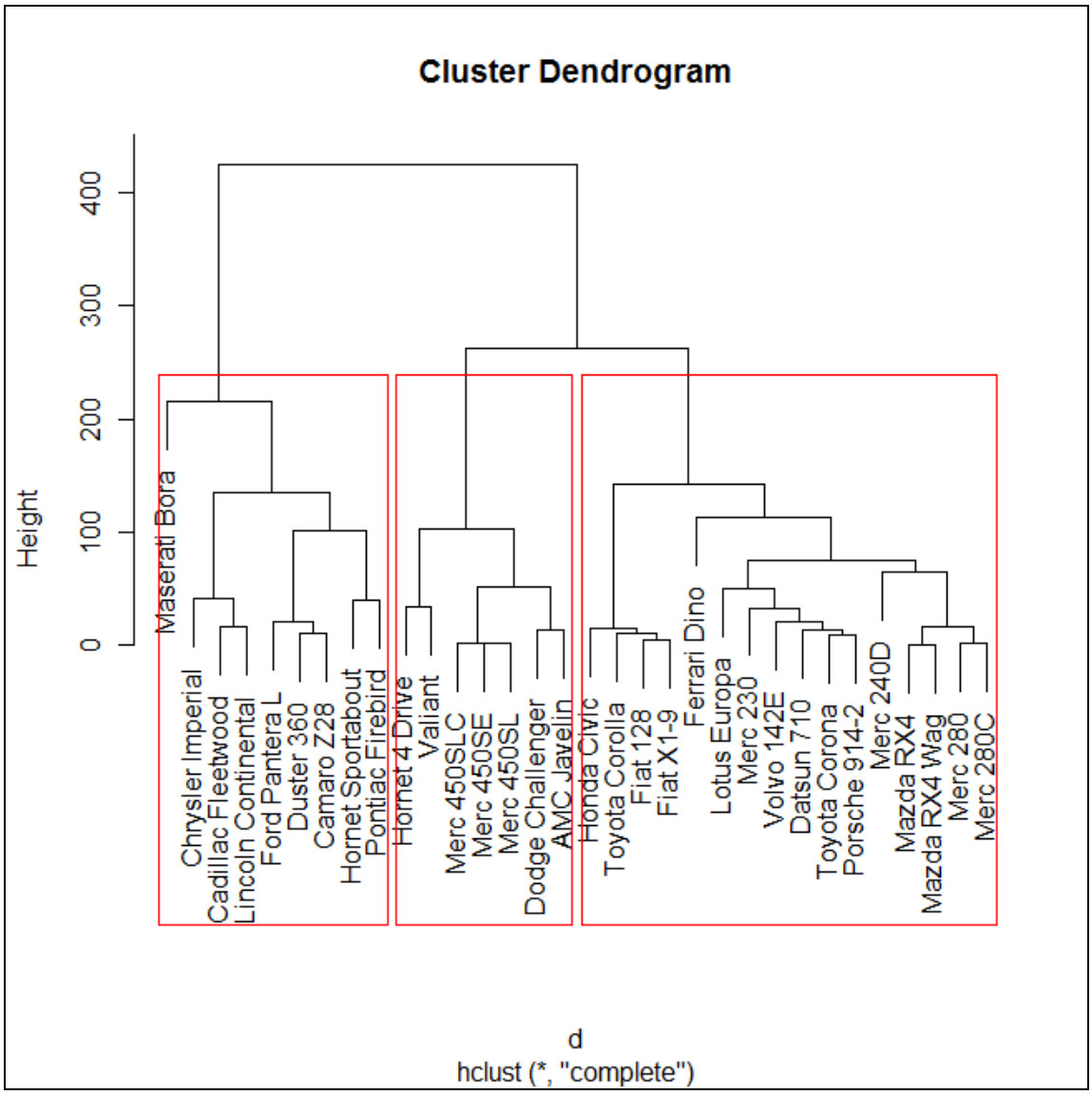
Arguments

tree	an object of the type produced by hclust.
k, h	Scalar. Cut the dendrogram such that either exactly k clusters are produced or by cutting at height h.
which, x	A vector selecting the clusters around which a rectangle should be drawn. which selects clusters by number (from left to right in the tree), x selects clusters containing the respective horizontal coordinates. Default is which = 1:k.
border	Vector with border colors for the rectangles.
cluster	Optional vector with cluster memberships as returned by cutree(hclust.obj, k = k), can be specified for efficiency if already computed.

Example

> plot(hc)

> rect.hclust(hc, k=3, border="red")



References

[1] http://www.wikipedia.org

[2] http://www-users.cs.umn.edu/~kumar/dmbook

[3] http://127.0.0.1:30448/library/stats/html/kmeans.html

[4] http://127.0.0.1:22477/library/stats/html/hclust.html

[5] http://www.r-tutor.com

[6] http://127.0.0.1:22477/library/stats/html/rect.hclust.html