18 - Cluster Analysis

Junvie Pailden

SIUE, F2017, Stat 589

November 09, 2017

Introduction

Objective: Given p-dimensional observations, group these into g significantly distinct groups, within which they are homogeneous (similar).

- Grouping or clustering is distinct from the classification methods discussed in the previous chapter.
- Classification pertains to a **known** number of groups, and the operational objective is to assign new observations to one of these groups.
- Cluster analysis make no assumptions concerning the number of groups or the group structure.
- Grouping is done on the basis of similarities or distances (dissimilarities).

Distances between pairs of items

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$. Common measurements/metrics between $\mathbf{x}$ and $\mathbf{y}$ are

1. Euclidean distance; L^2

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= ||\mathbf{x} - \mathbf{y}|| = \sqrt{(\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})} \\ &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_p - y_p)^2} \end{aligned}$$

2. Manhattan or City-block distance; L^1

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} |x_i - y_i|$$

Euclidean distance vs City-block distance

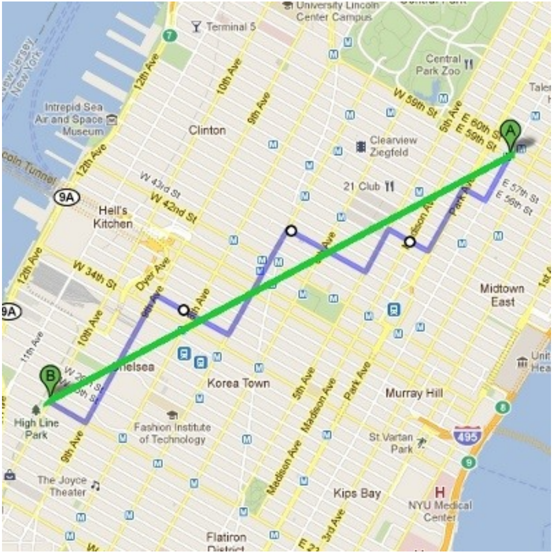


Figure 1: Source: https://goo.gl/LwB9yB

Distances between pairs of items

3. Minkowski distance; L^m

$$d(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{1/m}$$

4. Mahalanobis distance; MD

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})}$$

where $\mathbf{S}$ contains the sample variances and covariances.

Measurements (1) -(4) satisfy the definition of true distances:

1. $d(\mathbf{x},\mathbf{y}) \geq 0$, equality iff $\mathbf{x} = \mathbf{y}$
2. $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$.
3. $d(\mathbf{x},\mathbf{z}) \leq d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z})$

Which clustering algorithm to choose?

1. **Hierarchical methods**

- Clusters formed sequentially, with the number of clusters decreasing as clusters merged with other similar clusters *agglomerative hierarchical methods* or split into less homogeneous groups *divisive methods*.
- We will only cover agglomerative clustering.

2. **Nonhierarchical (partitioning) methods**

- Designed to group items into a collection of K clusters where the number of clusters may either be specified in advance or determined as part of the procedure.
- We will only cover **K-means method**.

Agglomerative clustering

1. Start with N clusters, each containing a single entity and an $N \times N$ symmetric matrix of distances $\mathbf{D} = \{d_{ik}\}$.

2. Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between "most similar" clusters U and V be d_{UV}.

3. Merge clusters U and V. Label the newly formed cluster (UV). update the entries in the distance matrix by (a) deleting the rows and columns corresponding to clusters U and V and (b) adding a row and column giving the distances between cluster (UV) and the remaining clusters.

Agglomerative clustering (cont.)

4. Repeat Steps 2 and 3 a total of $N - 1$ times. (All objects will be in a single cluster after the algorithm terminates.) Record the identity of clusters that are merged and the levels (distances or similarities) at which the mergers take place.

   Note: Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering.

Single Linkage

- Initially, we find the smallest distance in $\mathbf{D} = \{d_{ik}\}$ and merge corresponding objects to get cluster (UV).
- The general distances (in Step 3) between (UV) and any other cluster W are computed by

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\}$$

where $d_{UW}$and d_{VW} are distances between the nearest neighbors of clusters U and V and clusters V and W, respectively.

- The results of single linkage clustering can be graphically displayed in the form of a **dendogram**.
- The branches in the tree represent clusters. The branches come together (merge) at nodes.

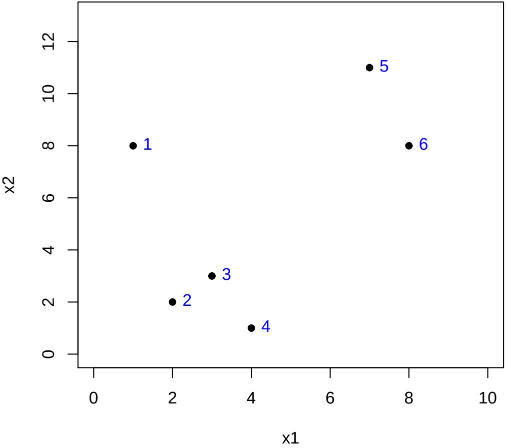
Single linkage heirarchical clustering example I

```
x1 <- c(1, 2, 3, 4, 7, 8)
x2 <- c(8, 2, 3, 1, 11, 8)
X <- cbind(x1, x2)
(X.d <- dist(X, method = "euclidean"))  # default distance
```

```
#       1     2     3     4     5
# 2  6.08
# 3  5.39  1.41
# 4  7.62  2.24  2.24
# 5  6.71 10.30  8.94 10.44
# 6  7.00  8.49  7.07  8.06  3.16
```

```
plot(X, pch =16, xlim = c(0,10), ylim = c(0,13))
text(x1, x2, 1:6, pos = 4, col = "blue")
```

Single linkage heirarchical clustering example II



Single linkage heirarchical clustering using `hclust` I

```
# use builtin function hclust() and specify
# method = "single" for single linkage
X.hc.s <- hclust(X.d, method="single")
# using the result in hclust(), cuttree()
# cuts a tree into several groups by either
# number of groups, say k=3, or the cut height h
cutree(X.hc.s, k = 3)
```

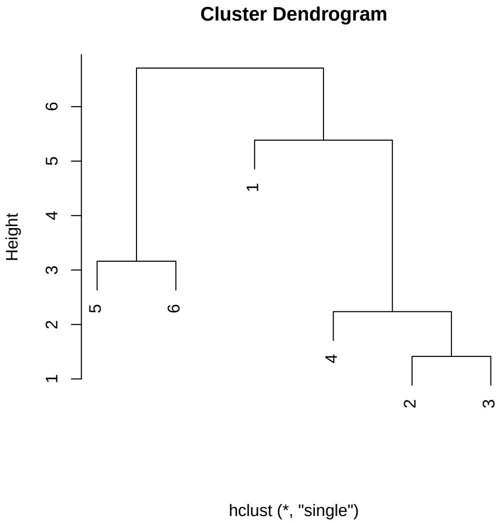
```
# [1] 1 2 2 2 3 3
```

Dendogram I

- Cluster 1 $= \{1\}$
- Cluster 2 $= \{2,3,4\}$
- Cluster 3 $= \{5,6\}$

```r
# plotting an `hclust` object creates a dendogram
plot(X.hc.s, xlab = NA)
```

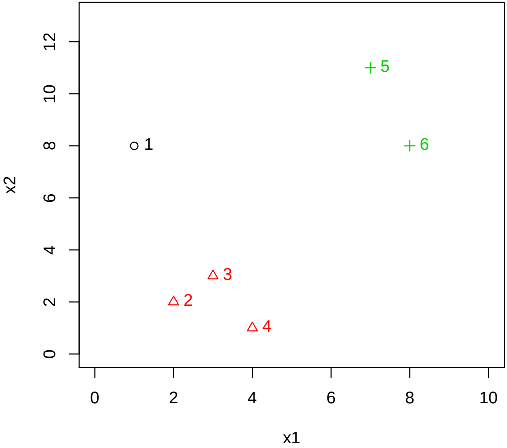
Dendogram II



Single linkage heirarchical clustering I

```
plot(X, xlim = c(0,10), ylim = c(0,13),
     pch = cutree(X.hc.s, 3),
         col = cutree(X.hc.s, 3))
text(x1, x2, 1:6, pos = 4, col=cutree(X.hc.s, 3))
```

Single linkage heirarchical clustering II



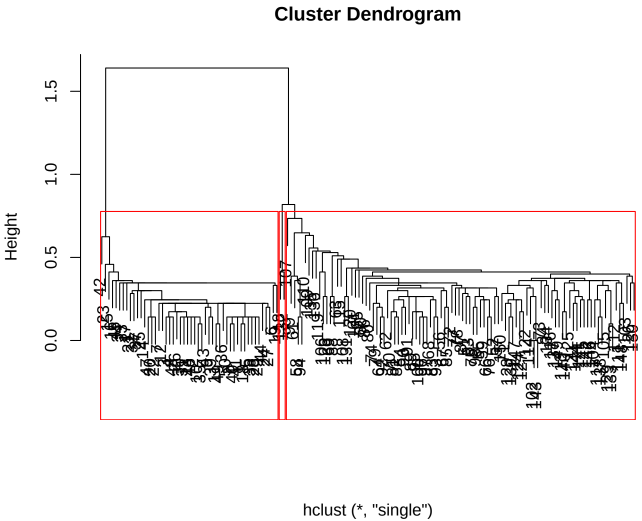
IRIS Data: Single linkage HC I

```
# remove, non-numeric column, compute euclidean distance
iris.d <- dist(iris[,-5])
iris.hc.s <- hclust(iris.d, method = "single")
# create estimated labels for each obs
table(predicted = cutree(iris.hc.s, k = 3), actual = iris[
```

```
#          actual
# predicted setosa versicolor virginica
#         1     50          0         0
#         2      0         50        48
#         3      0          0         2
```

```
plot(iris.hc.s, xlab = NA)
# creates a rectangular cluster border
rect.hclust(iris.hc.s, k = 3, border = "red")
```

IRIS Data: Single linkage HC II



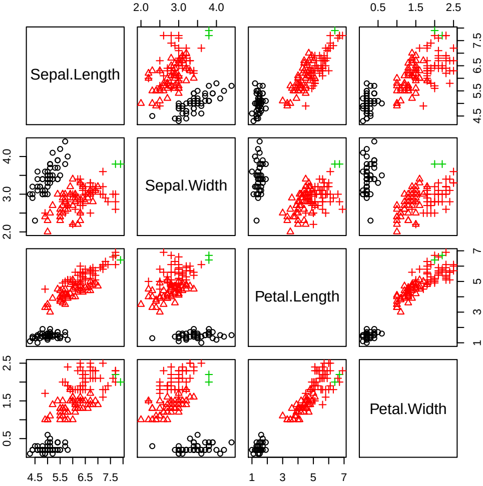
IRIS Data: Single linkage HC, analysis

- We can use the known species variable to determine how well the clustering method (which doesn't use the species variable) is able to reconstruct the species.
- If it works well, plotting character and color should match well.
- If one approach doesn't appear to work very well, we can try other linkage methods.
- For the iris data, single linkage produces only two clusters instead of the three species variable.

IRIS Data Clustered I

```r
# specify point type according to species
# specify color by clusters with k = 3
pairs(iris[,-5], pch = unclass(iris[,5]),
          col = cutree(iris.hc.s, k = 3))
```

IRIS Data Clustered II



USA Arrests Data I

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.

```
str(USArrests)
```

```
# 'data.frame': 50 obs. of  4 variables:
#  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4
#  $ Assault : int  236 263 294 190 276 204 110 238 335 211
#  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
#  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 3
```

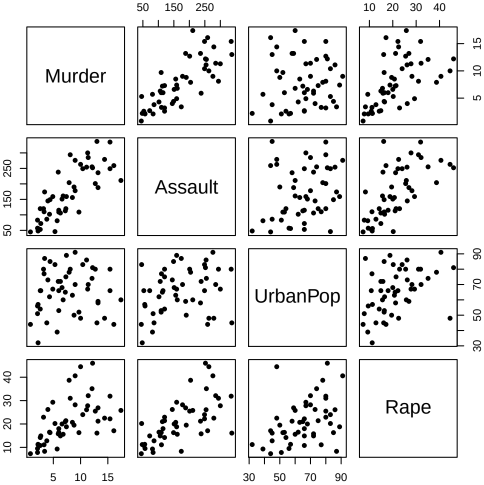
```
head(USArrests)
```

USA Arrests Data II

```
#            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
```

```r
pairs(USArrests, pch = 16)
```

USA Arrests Data III



USA Arrests: Single Linkage HC I

```
arrests.hc.s <- hclust(dist(USArrests), method = "single")
# check contents of clusters if number of cluster is speci
table(cutree(arrests.hc.s, k = 2))
```

```
#
#  1  2
# 49  1
```

```
table(cutree(arrests.hc.s, k = 3))
```

```
#
#  1  2  3
# 48  1  1
```

USA Arrests: Single Linkage HC II

```
table(cutree(arrests.hc.s, k = 4))
```

```
#
#  1  2  3  4
# 47  1  1  1
```

```
table(cutree(arrests.hc.s, k = 5))
```

```
#
#  1  2  3  4  5
# 13  1 34  1  1
```

```
table(cutree(arrests.hc.s, k = 6))
```

USA Arrests: Single Linkage HC III

```
#
#  1  2  3  4  5  6
# 13  1 14 20  1  1
```

```r
table(cutree(arrests.hc.s, k = 7))
```

```
#
#  1  2  3  4  5  6  7
#  9  1  4 14 20  1  1
```

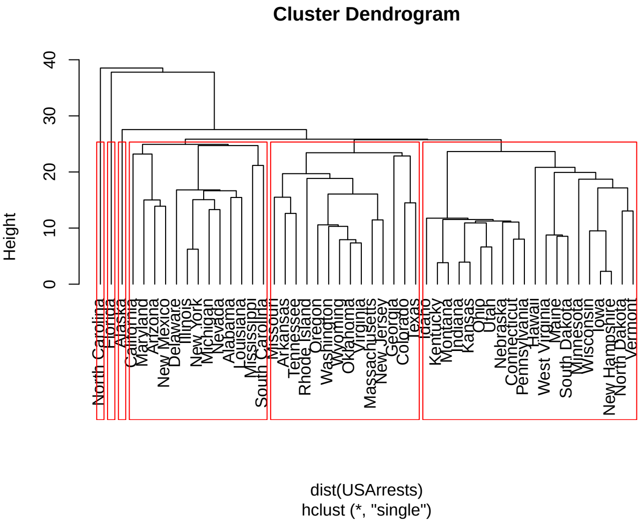
```r
table(cutree(arrests.hc.s, k = 8))
```

```
#
#  1  2  3  4  5  6  7  8
#  7  1  4 14 20  1  2  1
```

USA Arrests: Dendogram I

```r
plot(arrests.hc.s, hang = -1) # adjust labels placement
# creates a rectangular cluster border
rect.hclust(arrests.hc.s, k = 6, border = "red")
```

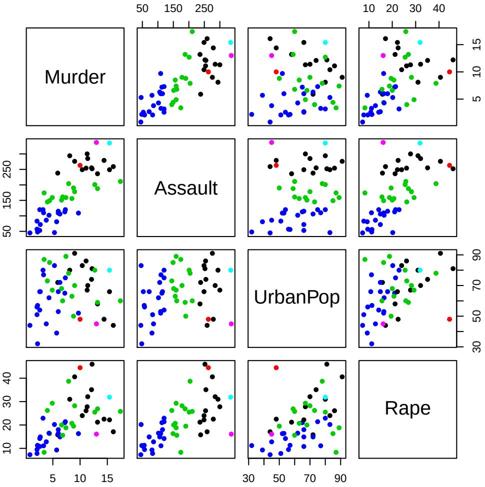
USA Arrests: Dendogram II



USA Arrests Data Clustered I

```r
# specify point type according to species
# specify color by clusters with k = 3
pairs(USArrests,
      col = cutree(arrests.hc.s, k = 6),
      pch = 16)
```

USA Arrests Data Clustered II



Complete Linkage

- At each stage, the distance between clusters is determined by the distance between the two elements from each cluster, that are **most distant**.
- Ensures that all items in a cluster are within some maximum distance of each other.
- Start by finding the minimum entry in $\mathbf{D} = \{d_{ik}\}$ ~and merging the corresponding objects to get cluster (UV).
- For Step 3, the distances between (UV) and any other cluster W are computed by

$$d^*_{(UV)W} = \max\{d_{UV}, d_{VW}\}$$

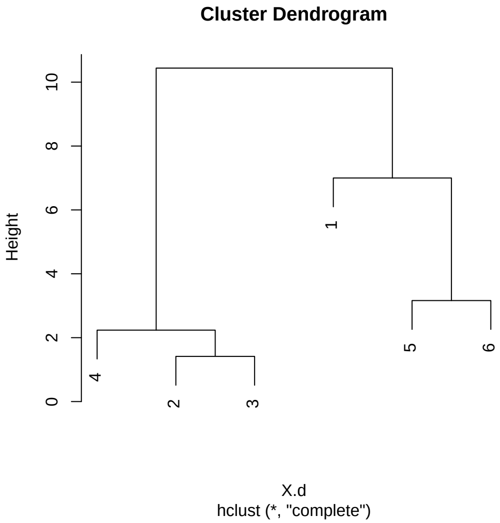
Complete linkage heirarchical clustering example I

```r
# method = "complete" for complete linkage
X.hc.c <- hclust(X.d, method = "complete")
cutree(X.hc.c, k = 3)
```

```
# [1] 1 2 2 2 3 3
```

```r
# cluster 1 = {1}, cluster 2 = {2,3,4}
# cluster 3 = {5,6}
plot(X.hc.c)
```

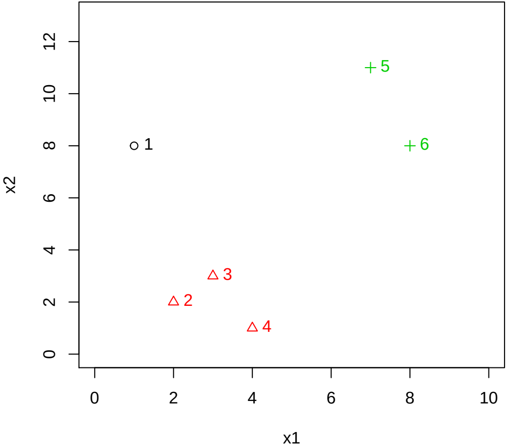
Complete linkage heirarchical clustering example II



Complete linkage heirarchical clustering example I

```r
plot(X,  xlim = c(0,10), ylim = c(0,13),
     pch = cutree(X.hc.c, 3),
         col = cutree(X.hc.c, 3))
text(x1,x2, 1:6, pos = 4, col=cutree(X.hc.c, 3))
```

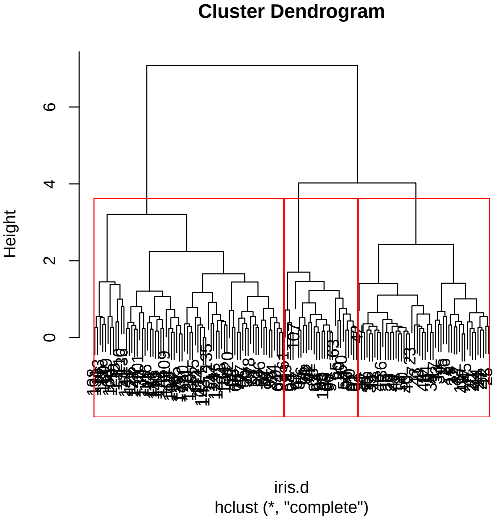
Complete linkage heirarchical clustering example II



IRIS Data: Complete linkage HC I

```r
iris.hc.c <- hclust(iris.d, method = "complete")
plot(iris.hc.c)
rect.hclust(iris.hc.c, k =3, border="red")
```

IRIS Data: Complete linkage HC II

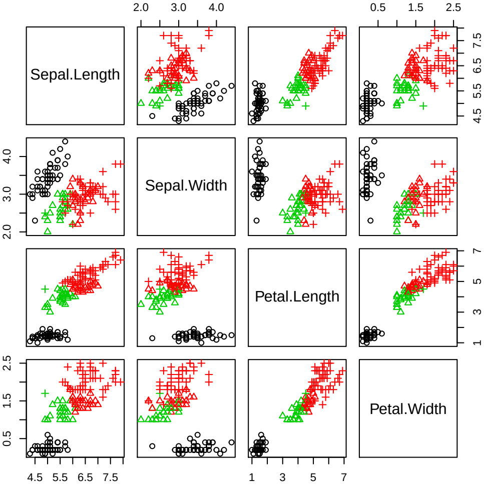


IRIS Data: Complete linkage HC I

Complete linkage clustering gives at least more balanced clusters and somewhat obeys the natural three species variable grouping.

```r
# specify point type according to species
# specify color by clusters with k = 3 using cutree()
pairs(iris[,-5], pch = unclass(iris[,5]),
         col=cutree(iris.hc.c, k = 3))
```

IRIS Data: Complete linkage HC II



USA Arrests: Complete Linkage HC I

```
arrests.hc.c <- hclust(dist(USArrests), method = "complete"
# check contents of clusters if number of cluster is speci
table(cutree(arrests.hc.c, k = 2))
```

```
#
#  1  2
# 16 34
```

```
table(cutree(arrests.hc.c, k = 3))
```

```
#
#  1  2  3
# 16 14 20
```

USA Arrests: Complete Linkage HC II

```r
table(cutree(arrests.hc.c, k = 4))
```

```
#
#  1  2  3  4
# 14 14 20  2
```

```r
table(cutree(arrests.hc.c, k = 5))
```

```
#
#  1  2  3  4  5
# 14 14 10  2 10
```

```r
table(cutree(arrests.hc.c, k = 6))
```

USA Arrests: Complete Linkage HC III

```
#
#  1  2  3  4  5  6
# 14  6 10  2 10  8
```

```r
table(cutree(arrests.hc.c, k = 7))
```

```
#
#  1  2  3  4  5  6  7
#  6  8  6 10  2 10  8
```

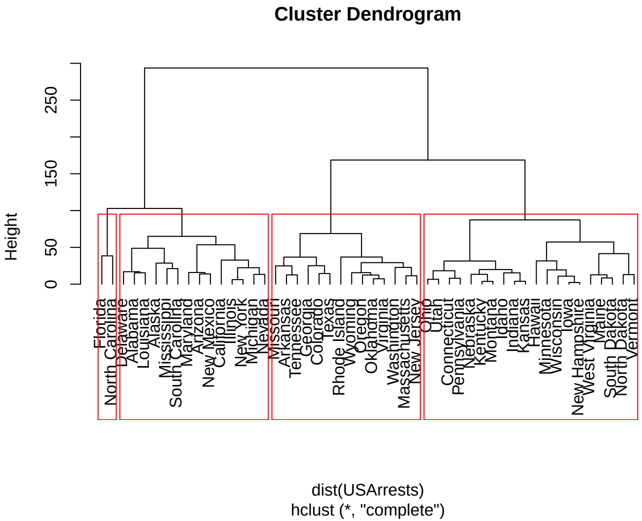
```r
table(cutree(arrests.hc.c, k = 8))
```

```
#
#  1  2  3  4  5  6  7  8
#  6  8  6 10  2  5  5  8
```

USA Arrests: Complete Linkage HC I

```r
plot(arrests.hc.c, hang = -1) # adjust labels placement
# creates a rectangular cluster border
rect.hclust(arrests.hc.c, k = 4, border = "red")
```

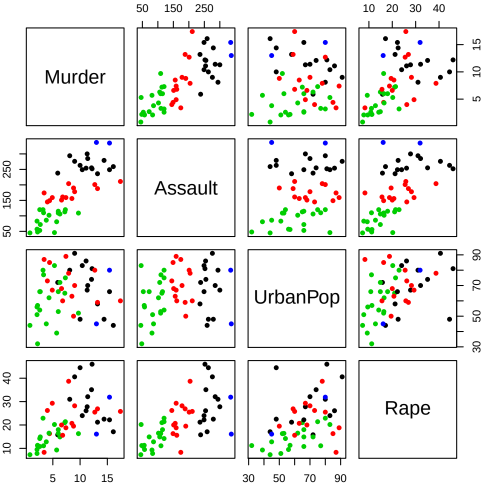
USA Arrests: Complete Linkage HC II



USA Arrests Complete Linkage HC I

```r
# specify point type according to species
# specify color by clusters with k = 3
pairs(USArrests,
      col = cutree(arrests.hc.c, k = 4),
      pch = 16)
```

USA Arrests Complete Linkage HC II



Average Linkage

- Treats the distance between two clusters as the average distance between all pairs of items where one member of a pair belongs to each cluster.
- Start by finding the minimum entry in $\mathbf{D} = \{d_{ik}\}$ ~and merging the corresponding objects to get cluster (UV).
- For Step 3, the distances between (UV) and the other cluster W are determined by

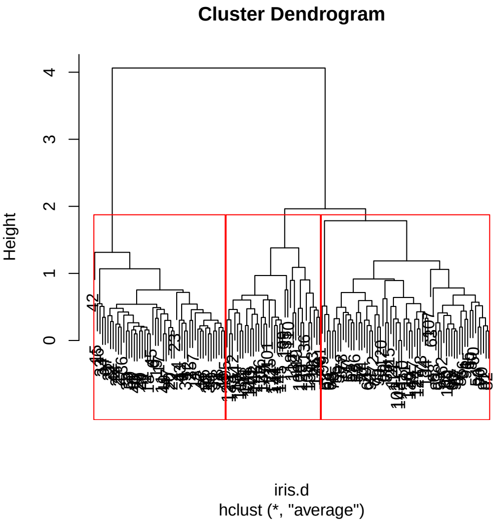
$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W}$$

where d_{ik} is the distance between object i in the cluster (UV) and object k in the cluster W, and $N_{(UV)}$ and N_W are the number of items in clusters (UV) and W.

IRIS Data: Average linkage HC I

```r
iris.hc.av <- hclust(iris.d, method = "average")
plot(iris.hc.av)
rect.hclust(iris.hc.av, k =3, border="red")
```

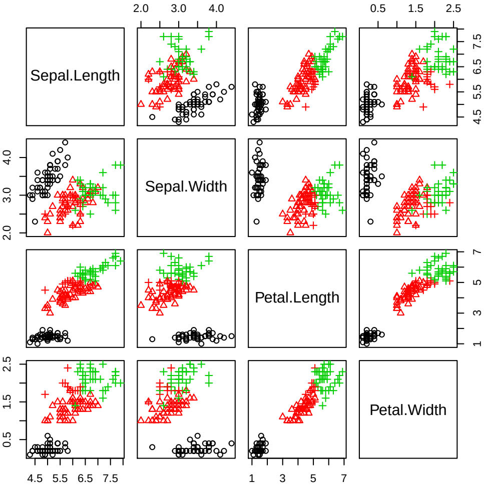
IRIS Data: Average linkage HC II



IRIS Data: Average linkage HC I

```r
pairs(iris[,-5], pch = unclass(iris[,5]),
          col = cutree(iris.hc.av, k = 3))
```

IRIS Data: Average linkage HC II



USA Arrests: Average Linkage HC I

```r
arrests.hc.av <- hclust(dist(USArrests), method = "average'
# check contents of clusters if number of cluster is speci
table(cutree(arrests.hc.av, k = 2))
```

```
#
#  1  2
# 16 34
```

```r
table(cutree(arrests.hc.av, k = 3))
```

```
#
#  1  2  3
# 16 14 20
```

USA Arrests: Average Linkage HC II

```
table(cutree(arrests.hc.av, k = 4))
```

```
#
#  1  2  3  4
# 14 14 20  2
```

```
table(cutree(arrests.hc.av, k = 5))
```

```
#
#  1  2  3  4  5
# 14 14 10  2 10
```

```
table(cutree(arrests.hc.av, k = 6))
```

USA Arrests: Average Linkage HC III

```
#
#  1  2  3  4  5  6
# 14  6 10  2 10  8
```

```
table(cutree(arrests.hc.av, k = 7))
```

```
#
#  1  2  3  4  5  6  7
# 10  4  6 10  2 10  8
```

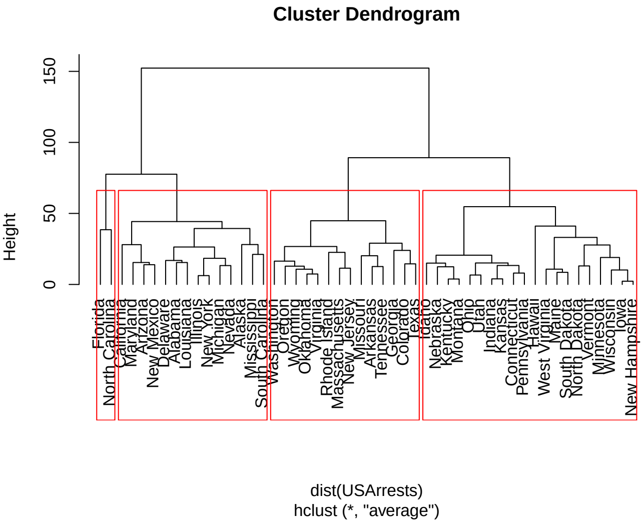
```
table(cutree(arrests.hc.av, k = 8))
```

```
#
#  1  2  3  4  5  6  7  8
# 10  4  6 10  2  1  9  8
```

USA Arrests: Average Linkage HC I

```r
plot(arrests.hc.av, hang = -1) # adjust labels placement
# creates a rectangular cluster border
rect.hclust(arrests.hc.av, k = 4, border = "red")
```

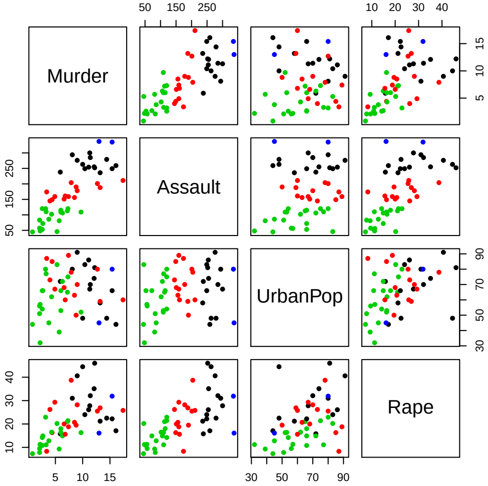
USA Arrests: Average Linkage HC II



USA Arrests Average Linkage HC I

```r
# specify point type according to species
# specify color by clusters with k = 3
pairs(USArrests,
      col = cutree(arrests.hc.av, k = 4),
      pch = 16)
```

USA Arrests Average Linkage HC II



Example 12.7 Complete vs Average linkage HC I

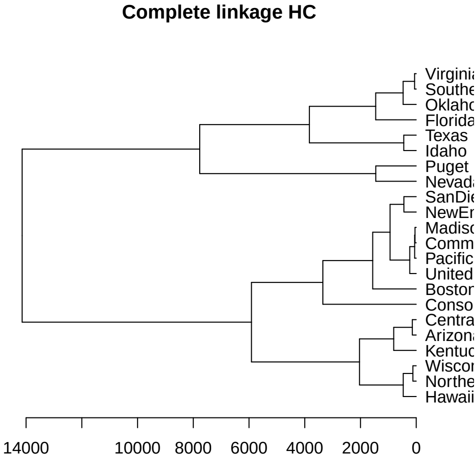
Data collected on 22 US public utility companies for the year 1975 using 8 measurements.

- X_1: Fixed-charge coverage ratio (income/debt)
- X_2: Rate of return on capital.
- X_3: Cost per KW capacity in place.
- X_4: Annual load factor.
- X_5: Peak kWh deman growth from 1974 to 1975.
- X_6: Sales (kWh use per year).
- X_7: Percent nuclear
- X_8: Total fuel costs (cents per kWh).

Example 12.7 Complete vs Average linkage HC II

```r
X <- read.table("T12-5.DAT")
utility <- X[,-9]
rownames(utility) <- X[,9]
utility.d <- dist(utility)
# complete linkage
utility.hc.c = hclust(utility.d, method="complete")
# average linkage
utility.hc.av = hclust(utility.d, method="average")
plot(as.dendrogram(utility.hc.c), horiz = T,
    main = "Complete linkage HC")
```

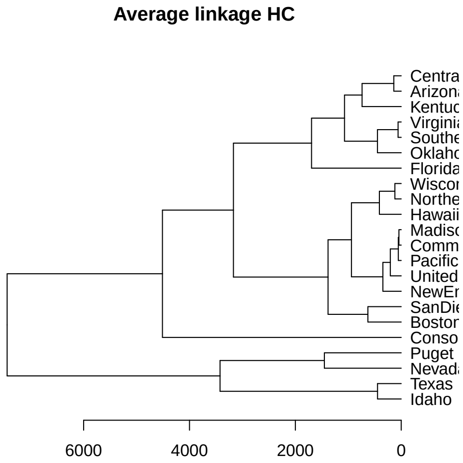
Example 12.7 Complete vs Average linkage HC III



Example 12.7 Complete vs Average linkage HC IV

```
plot(as.dendrogram(utility.hc.av), horiz = T,
    main = "Average linkage HC")
```

Example 12.7 Complete vs Average linkage HC V



Comments on Hierarchical Procedures

- All agglomerative procedures follow the basic algorithm discussed in this section.
- Sources of error and variation are not formally considered in hierarchical procedures - meaning that HC procedures are sensitive to outliers.
- It a good practice to try several clustering procedures methods and withing a given method, a couple of different ways of assigning distances.
- If the outcomes from several methods are (roughly) consistent with one another, perhaps a case for "natural" groupings can be advanced.