



Chapter 9: Cluster Analysis

Jingyuan Liu
Department of Statistics, School of Economics
Wang Yanan Institute for Studies in Economics
Xiamen University

Outline

1 Introduction to Cluster Analysis

- Intuition and Applications
- Measures of Similarity

2 Hierarchical Clustering

- Introduction to Hierarchical Clustering
- Types of Hierarchical Methods
- Choosing Number of Clusters
- Application Example with R Implementation

3 Nonhierarchical Methods: Partitioning

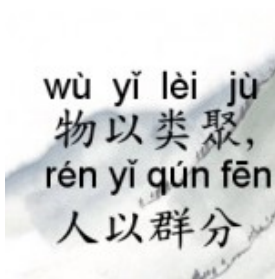
- k -Means Method
- Application Example with R Implementation

Outline

1 Introduction to Cluster Analysis

- Intuition and Applications
- Measures of Similarity

Intuition of Clustering



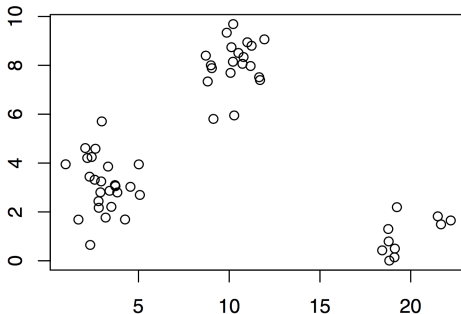
“An intelligent being cannot treat every object it sees as a unique entity unlike anything else in the universe. It has to put objects in categories so that it may apply its hard-won knowledge about similar objects in the past to the object at hand.” – Pinker 1997.

Introduction of Cluster Analysis

- One of the most basic abilities of living creatures involves the grouping of similar objects.
- In cluster analysis, we search for “natural” patterns in a data set by grouping the (multivariate) observations into different clusters.
- The goal is to find an optimal grouping for which the observations or objects within each cluster are similar, but the clusters are dissimilar to each other.

Cluster Analysis: Toy Example

Consider the following scatterplot:



The conclusion that there are 3 natural groups or clusters of dots is clear. Clusters are identified by the assessment of the relative distances between points, and in this example the homogeneity of each cluster and the separation makes the task very simple.

Clustering vs. Classification

Clustering differs fundamentally from classification:

- In classification analysis, we allocate the observations to a known number of predefined groups or populations.
- In cluster analysis, neither the number of groups nor the groups themselves are known in advance. It is a more primitive technique in that no assumptions are made concerning the number of groups or the group structure.
- In terminology of machine learning, cluster analysis is a **unsupervised learning method**, and classification is a **supervised learning method**.

Applications of Cluster Analysis

- In marketing, cluster analysis is used to segment customers based on demographic and transaction history information, and marketing strategies are tailored for different segments; or to identify groups of similar products according to competitive measures of similarity (market structure analysis).
- In finance, cluster analysis is used to create balanced portfolios: Given data on a variety of investment opportunities (e.g., stocks), one may find clusters based on financial performance variables such as return, volatility, and other characteristics, such as market capitalization. Selecting securities from different clusters can help create a balanced portfolio.
- Cluster analysis has also been applied in areas including astronomy, archaeology, medicine, chemistry, education, psychology, linguistics, and sociology.

Outline

1 Introduction to Cluster Analysis

- Intuition and Applications
- Measures of Similarity

Measures of Similarity: Motivation

- As discussed, cluster analysis attempts to form groups or clusters of similar items based on several measurements made on these items. Grouping is done on the basis of measures of similarities or distances/dissimilarities.
- We must first develop a quantitative scale on which to measure the association (similarity) between objects.

Measures of Similarity

Recall that the distance between two p -variate observations (items) $\mathbf{x} = (x_1, \dots, x_p)'$, and $\mathbf{y} = (y_1, \dots, y_p)'$ can be measured by

- **Euclidean distance:** $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})} = \sqrt{\sum_j (x_j - y_j)^2}$,
- **Mahalanobis distance:** $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}$.

Additional popular measures of “distance” are

- **Minkowski metric:** $d(\mathbf{x}, \mathbf{y}) = \left(\sum_j |x_j - y_j|^m \right)^{1/m}$;
- **Canberra metric:** $d(\mathbf{x}, \mathbf{y}) = \sum_j |x_j - y_j| / (x_j + y_j)$;
- **Czekanowski coefficient:**
 $d(\mathbf{x}, \mathbf{y}) = 1 - 2 \sum_j \min(x_j, y_j) / \sum_j (x_j + y_j)$.

Measurements of Similarity: Remarks

- Canberra metric and Czekanowski coefficient are defined for nonnegative variables only.
- For the n observations $\mathbf{y}_1, \dots, \mathbf{y}_n$, we can compute an $n \times n$ matrix $\mathbf{D} = (d_{ij})$ of distances/dissimilarities, where $d_{ij} = d(\mathbf{y}_i, \mathbf{y}_j)$. So \mathbf{D} will be symmetric with diagonals 0.
- Mahalanobis distance uses a pooled covariance matrix \mathbf{S} , leading to distortion of the variance and covariances if there really are natural clusters. Thus, it is preferred to use the Euclidean distance.

Outline

2 Hierarchical Clustering

- Introduction to Hierarchical Clustering
- Types of Hierarchical Methods
- Choosing Number of Clusters
- Application Example with R Implementation

Introduction to Hierarchical Clustering

- Considering all possible clusters are not computationally feasible, especially when there are many objects. Thus a hierarchical technique is preferred.
- There are two directions of hierarchical clustering:
 - **Agglomerative hierarchical methods** start with the individual objects. The most “similar” objects are merged every step, until all subgroups are fused into a single cluster.
 - **Divisive hierarchical methods** work in the opposite direction. An initial single group of objects is divided into two subgroups such that the objects in one subgroup are “far from” the objects in the other. The process continues until each object forms a group.
- We focus on the agglomerative method here.

Types of Hierarchical Clusters

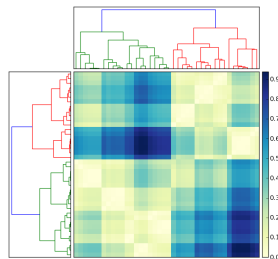
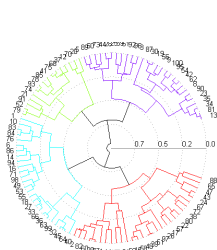
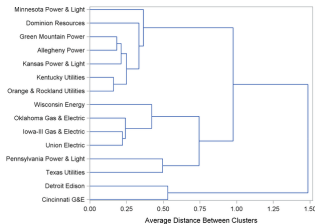
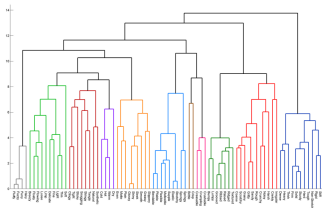
In either direction of the hierarchical methods, we need to measure the “similarity” between two groups/clusters. We introduce two techniques:

- Linkage methods: based on an aforementioned distance
 - Single linkage
 - Complete linkage
 - Average linkage
 - Centroid method
- Ward's method: based on the within- and between-cluster variability

Dendrogram

The results of a hierarchical clustering procedure can be displayed graphically using a **dendrogram**, which shows all the steps in the hierarchical procedure, including the distances or changes in variabilities at which clusters are merged.

Dendrogram: Examples



Outline

2 Hierarchical Clustering

- Introduction to Hierarchical Clustering
- Types of Hierarchical Methods
- Choosing Number of Clusters
- Application Example with R Implementation

Single Linkage (Nearest Neighbor)

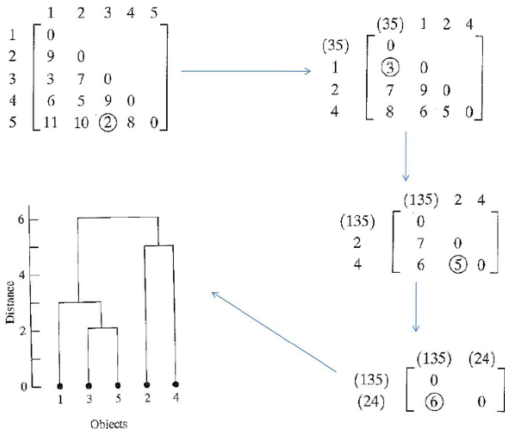
Single linkage method/Nearest neighbor method uses the minimum distance between a point in cluster A and a point in cluster B to define the distance between A and B :

$$D(A, B) = \min\{d(\mathbf{y}_i, \mathbf{y}_j), \text{ for } \mathbf{y}_i \text{ in } A \text{ and } \mathbf{y}_j \text{ in } B\},$$

where $d(\mathbf{y}_i, \mathbf{y}_j)$ is some measure of distance, such as the Euclidian distance.

Agglomerative Single Linkage Method

We illustrate the method with the following distance matrix:



Single Linkage: Example

Example: The crime rates per 100,000 population for various US cities were compared. The data are in the table below.

City	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto Theft
Atlanta	16.5	24.8	106	147	1112	905	494
Boston	4.2	13.3	122	90	982	669	954
Chicago	11.6	24.7	340	242	808	609	645
Dallas	18.1	34.2	184	293	1668	901	602
Denver	6.9	41.5	173	191	1534	1368	780
Detroit	13.0	35.7	477	220	1566	1183	788
Hartford	2.5	8.8	68	103	1017	724	468
Honolulu	3.6	12.7	42	28	1457	1102	637
Houston	16.8	26.6	289	186	1509	787	697
Kansas City	10.8	43.2	255	226	1494	955	765
Los Angeles	9.7	51.8	286	355	1902	1386	862
New Orleans	10.3	39.7	266	283	1056	1036	776
New York	9.4	19.4	522	267	1674	1392	848
Portland	5.0	23.0	157	144	1530	1281	488
Tucson	5.1	22.9	85	148	1206	756	483
Washington	12.5	27.6	524	217	1496	1003	793

For illustration purpose, we focus on the first 6 cities. The distance matrix **D** is given by

City	Distance between Cities					
Atlanta	0	536.6	516.4	590.2	693.6	716.2
Boston	536.6	0	447.4	833.1	915.0	881.1
Chicago	516.4	447.4	0	924.0	1073.4	971.5
Dallas	590.2	833.1	924.0	0	527.7	464.5
Denver	693.6	915.0	1073.4	527.7	0	358.7
Detroit	716.2	881.1	971.5	464.5	358.7	0

The smallest distance is 358.7 between Denver and Detroit, and therefore the two cities are joined at the first step to $C_1 = \{Denver, Detroit\}$.

In the next step, the distance matrix is calculated for Atlanta, Boston, Chicago, Dallas, and C_1 :

Atlanta	0	536.6	516.4	590.2	693.6
Boston	536.6	0	447.4	833.1	881.1
Chicago	516.4	447.4	0	924.0	971.5
Dallas	590.2	833.1	924.0	0	464.5
C_1	693.6	881.1	971.5	464.5	0

The smallest distance is 447.4 between Boston and Chicago. Therefore $C_2 = \{Boston, Chicago\}$.

Then the distance matrix is calculated for Atlanta, Dallas, C_1 and C_2 :

Atlanta	0	516.4	590.2	693.6
C_2	516.4	0	833.1	881.1
Dallas	590.2	833.1	0	464.5
C_1	693.6	881.1	464.5	0

The smallest distance is 464.5 between Dallas and C_1 , so that $C_3 = \{Dallas, C_1\}$.

The distance matrix for Atlanta, C_2 , and C_3 is given by

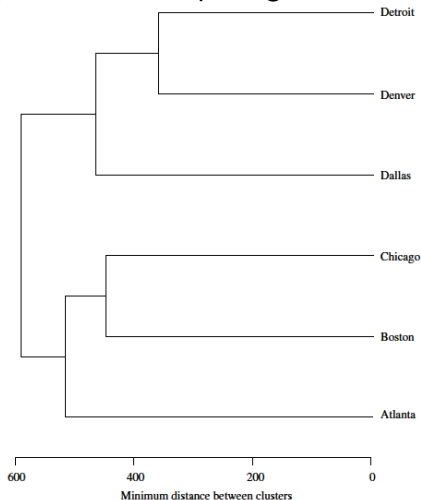
Atlanta	0	516.4	590.2
C_2	516.4	0	833.1
C_3	590.2	833.1	0

The smallest distance is 516.4, which defines the fourth cluster $C_4 = \{Atlanta, C_2\}$. The distance matrix for C_3 and C_4 is

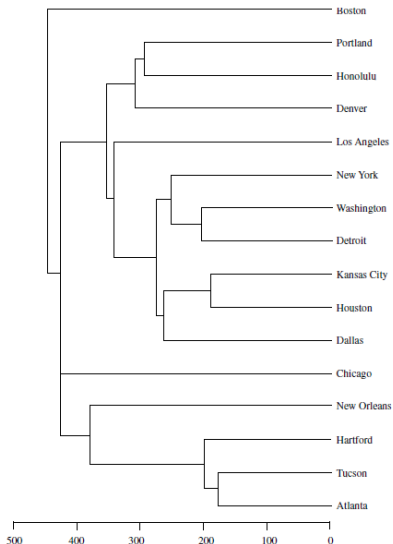
C_3	0	590.2
C_4	590.2	0

And the last cluster is given by $C_5 = C_3, C_4$.

The dendrogram for the steps is given in the figure below:



Using the
complete
city crime
data:



Single Linkage: Remarks

- In the dendrogram, the relative distances can be seen from the horizontal axis, with the scale running from right to left.
- All elements in the subsequent distance matrices are contained in the original distance matrix. This is also characteristic of the complete linkage method to be discussed next.

Complete Linkage (Farthest Neighbor)

In the **complete linkage approach/farthest neighbor method**, the distance between clusters A and B is defined as the maximum distance between a point in A and a point in B :

$$D(A, B) = \max\{d(\mathbf{y}_i, \mathbf{y}_j), \text{ for } \mathbf{y}_i \text{ in } A \text{ and } \mathbf{y}_j \text{ in } B\}.$$

Therefore, we merge the two clusters with the smallest distance defined above.

Complete Linkage: Example

Back to the crime rates example, we still use the first six city to illustrate the complete linkage method. Start from the original distance matrix:

City	Distance between Cities					
Atlanta	0	536.6	516.4	590.2	693.6	716.2
Boston	536.6	0	447.4	833.1	915.0	881.1
Chicago	516.4	447.4	0	924.0	1073.4	971.5
Dallas	590.2	833.1	924.0	0	527.7	464.5
Denver	693.6	915.0	1073.4	527.7	0	358.7
Detroit	716.2	881.1	971.5	464.5	358.7	0

The smallest distance is 358.7 between Denver and Detroit, so $C_1 = \{Denver, Detroit\}$. Since the first cluster is based on the initial distance matrix, it will be the same regardless of which hierarchical clustering method is used.

In the next step, the distance matrix is calculated for Atlanta, Boston, Chicago, Dallas, and C_1 :

Atlanta	0	536.6	516.4	590.2	716.2
Boston	536.6	0	447.4	833.1	915.0
Chicago	516.4	447.4	0	924.0	1073.4
Dallas	590.2	833.1	924.0	0	527.7
C_1	716.2	915.0	1073.4	527.7	0

The smallest distance is 447.4 between Boston and Chicago. Therefore, $C_2 = \{Boston, Chicago\}$. Note that this distance matrix differs from its analog for the single linkage method only in the distances between C_1 and the other cities.

At the next step, distances are calculated for Atlanta, Dallas, C_1 , and C_2 :

Atlanta	0	536.6	590.2	716.2
C_2	536.6	0	924.0	1073.4
Dallas	590.2	924.0	0	527.7
C_1	716.2	1073.4	527.7	0

The smallest distance, 527.7, defines $C_3 = \{Dallas, C_1\}$.

The distance matrix for Atlanta, C_2 , and C_3 is given by

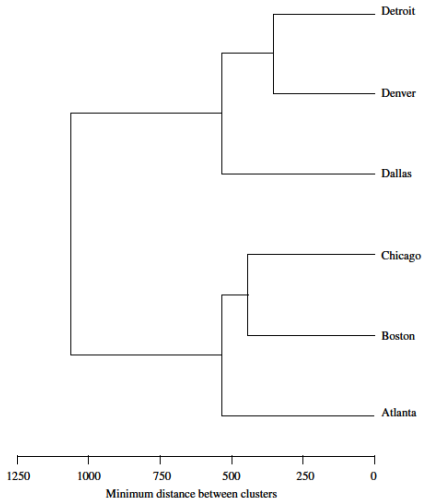
Atlanta	0	536.6	716.2
C_2	536.6	0	1073.4
C_3	716.2	1073.4	0

where the smallest distance 536.6 defines $C_4 = \{Atlanta, C_2\}$.
And the distance matrix for C_3 and C_4 is

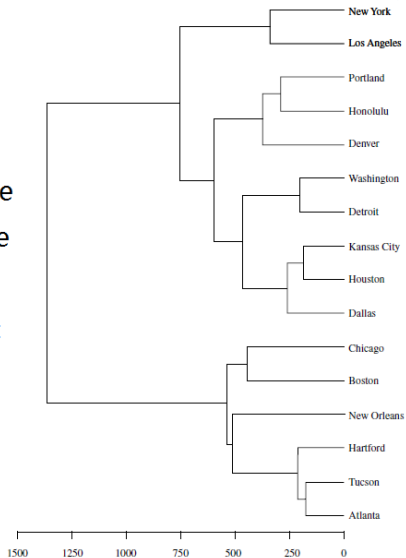
C_3	0	1073.4
C_4	1073.4	0

The last cluster is $C_5 = \{C_3, C_4\}$.

The corresponding dendrogram is as follows:



Using the
complete
data of
16 cities:



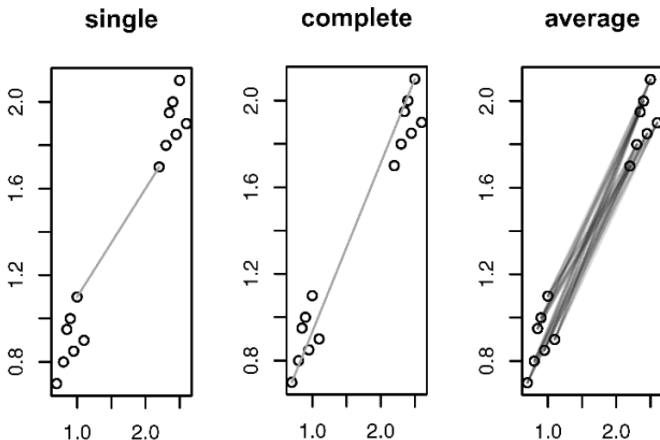
Average Linkage

In the **average linkage approach**, the distance between two clusters A and B is defined as the average of the $n_A n_B$ distances between the n_A points in A and the n_B points in B :

$$D(A, B) = \frac{1}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} d(\mathbf{y}_i, \mathbf{y}_j),$$

where the sum is over all \mathbf{y}_i in A and all \mathbf{y}_j in B . Particularly for the crime data, the average linkage method yields the same result as the complete linkage approach, and only the relative distances between clusters alter.

The following figures illustrate the difference among the single, complete and average linkage:



Centroid Method

In the **centroid method**, the distance between two clusters A and B is defined as the Euclidean distance between the mean vectors (often called centroids) of the two clusters:

$$D(A, B) = d(\bar{\mathbf{y}}_A, \bar{\mathbf{y}}_B),$$

where

$$\bar{\mathbf{y}}_A = \frac{1}{n_A} \sum_{i=1}^{n_A} \mathbf{y}_i, \quad \bar{\mathbf{y}}_B = \frac{1}{n_B} \sum_{j=1}^{n_B} \mathbf{y}_j.$$

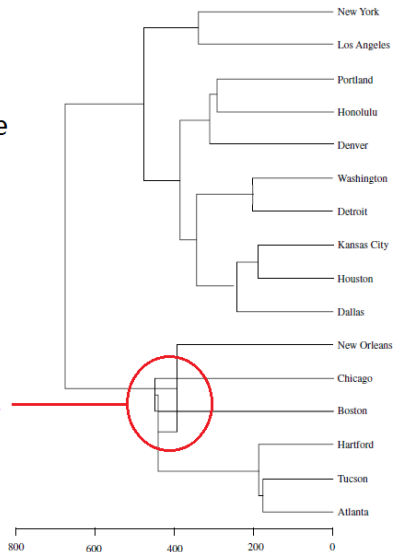
After the two clusters with the smallest distance between centroids are merged at each step, the centroid of the new cluster AB is given by the weighted average

$$\bar{\mathbf{y}}_{AB} = \frac{\sum_{i=1}^{n_A} \mathbf{y}_i + \sum_{j=1}^{n_B} \mathbf{y}_j}{n_A + n_B} = \frac{n_A \bar{\mathbf{y}}_A + n_B \bar{\mathbf{y}}_B}{n_A + n_B}.$$

Centroid Method: Example

Back to the
crime rate
example:
with all 16
cities.

Notice the
crossovers.



Centroid Method: Reversal/Inversion

- If an item or a cluster joins another cluster at a distance that is less than the distance for the previous merger of two clusters, we say that an **reversal** or a **inversion** has occurred. The inversion is represented by a **crossover** in the dendrogram.
- In some hierarchical methods, reversals cannot occur, such as the single linkage, complete linkage and average linkage. These distance measures are called **monotonic** or **ultrametric**.
- Clearly from the previous dendrogram, the centroid method is not monotonic.

Centroid Method with Midpoint

In the centroid method, if A contains a much larger number of items than B , then the new centroid $\bar{\mathbf{y}}_{AB}$ may be much closer to $\bar{\mathbf{y}}_A$ than to $\bar{\mathbf{y}}_B$. To avoid weighting the mean vectors by the cluster size, we can use the midpoint of the line that joins $\bar{\mathbf{y}}_A$ and $\bar{\mathbf{y}}_B$ as the point for computing new distances to other clusters:

$$\mathbf{m}_{AB} = \frac{1}{2}(\bar{\mathbf{y}}_A + \bar{\mathbf{y}}_B).$$

In the crime rate example, this modification does not make a difference for clustering - it only changes the relative distances.

Ward's Method

- **Ward's method/Incremental sum of squares method** uses the within-cluster (squared) distances and the between-cluster (squared) distances.
- If AB is the cluster obtained by combining clusters A and B , then the sum of within-cluster distances are

$$SSE_A = \sum_{i=1}^{n_A} (\mathbf{y}_i - \bar{\mathbf{y}}_A)' (\mathbf{y}_i - \bar{\mathbf{y}}_A) \text{ for } \mathbf{y}_i \in A$$

$$SSE_B = \sum_{i=1}^{n_B} (\mathbf{y}_i - \bar{\mathbf{y}}_B)' (\mathbf{y}_i - \bar{\mathbf{y}}_B) \text{ for } \mathbf{y}_i \in B$$

$$SS_{AB} = \sum_{i=1}^{n_A+n_B} (\mathbf{y}_i - \bar{\mathbf{y}}_{AB})' (\mathbf{y}_i - \bar{\mathbf{y}}_{AB}) \text{ for } \mathbf{y}_i \in AB,$$

where $\bar{\mathbf{y}}_{AB} = (n_A \bar{\mathbf{y}}_A + n_B \bar{\mathbf{y}}_B) / (n_A + n_B)$.

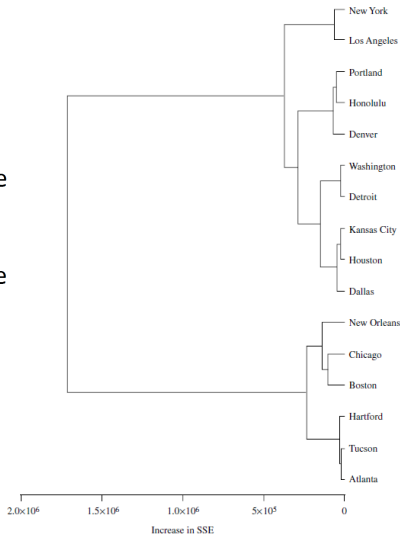
- Ward's method joins the two clusters A and B that minimize the increase in SSE , or equivalently, the between-cluster distance:

$$\begin{aligned} I_{AB} &= SSE_{AB} - (SSE_A + SSE_B) \\ &= \frac{n_A n_B}{n_A + n_B} (\bar{\mathbf{y}}_A - \bar{\mathbf{y}}_B)' (\bar{\mathbf{y}}_A - \bar{\mathbf{y}}_B). \end{aligned}$$

- The only difference between the Ward's method and the centroid method is the coefficient $n_A n_B / (n_A + n_B)$, if the distance in the centroid method is squared. Thus the cluster sizes have an impact on Ward's method but not on the centroid method.
- Compared to the centroid method, Ward's method is more likely to join smaller clusters.

Ward's Method: Example

Back to the
crime rate
example
with all the
16 cities



A Unified Distance

Suppose clusters A and B have been merged to form cluster AB . A general formula for the distance between AB and any other cluster C was given by Lance and Williams (1967):

$$D(C, AB) = \alpha_A D(C, A) + \alpha_B D(C, B) + \beta D(A, B) + \gamma |D(C, A) - D(C, B)|,$$

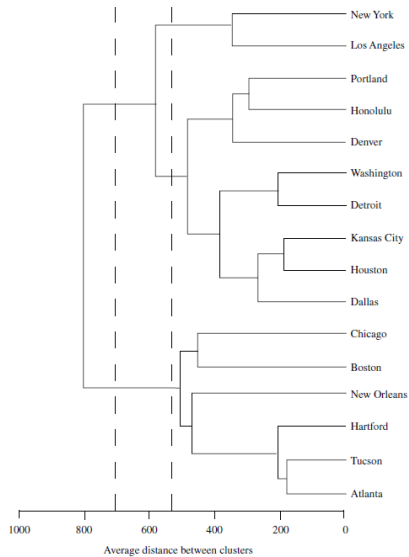
usually with constraints $\beta < 1$ and $\alpha_A + \alpha_B + \beta = 1$. The aforementioned agglomerative hierarchical methods can all be expressed as special cases of this flexible beta method.

Cluster Method	α_A	α_B	β	γ
Single linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Average linkage	$\frac{n_A}{n_A + n_B}$	$\frac{n_B}{n_A + n_B}$	0	0
Centroid	$\frac{n_A}{n_A + n_B}$	$\frac{n_B}{n_A + n_B}$	$\frac{-n_A n_B}{(n_A + n_B)^2}$	0
Centroid (midpoint)	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
Ward's method	$\frac{n_A + n_C}{n_A + n_B + n_C}$	$\frac{n_B + n_C}{n_A + n_B + n_C}$	$\frac{-n_C}{n_A + n_B + n_C}$	0

Outline

2 Hierarchical Clustering

- Introduction to Hierarchical Clustering
- Types of Hierarchical Methods
- **Choosing Number of Clusters**
- Application Example with R Implementation



Choosing Number of Clusters

- In hierarchical clustering, we can select k clusters from the dendrogram by cutting across the branches at a given level of the distance measure used by one of the axes.
- We wish to determine the value of k that provides the best fit to the data. One approach is to look for large changes in distances at which clusters are formed.
- In some applications, the number of clusters might be predetermined for a reasonable interpretation.

Steps of Agglomerative Clustering

We summarize the agglomerative clustering in steps:

- 1 Construct n initial clusters, each with a single item.
- 2 Compute the distance matrix for the n items.
- 3 Join the two clusters with the smallest distance.
- 4 Compute the distance matrix for the new clusters. If the number of clusters is 1, go to step 5, else go to 3.
- 5 Plot the dendrogram.
- 6 Choose the number of clusters and give the conclusion.

Outline

2 Hierarchical Clustering

- Introduction to Hierarchical Clustering
- Types of Hierarchical Methods
- Choosing Number of Clusters
- Application Example with R Implementation

Hierarchical Clustering: R Example

The chest, waist and hip measurements of 20 individuals are tabulated as follows:

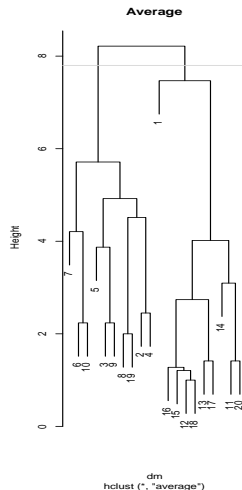
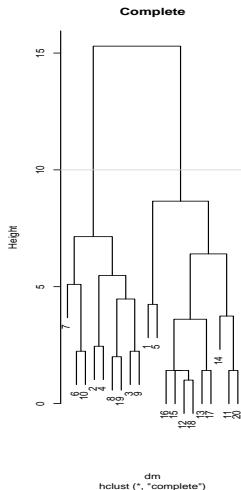
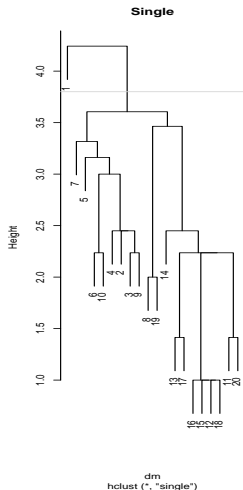
```
> library(HSAUR2)
> measure
      chest waist hips gender
1       34    30   32   male
2       37    32   37   male
3       38    30   36   male
4       36    33   39   male
5       38    29   33   male
6       43    32   38   male
7       40    33   42   male
8       38    30   40   male
9       40    30   37   male
10      41    32   39   male
11      36    24   35 female
12      36    25   37 female
13      34    24   37 female
14      33    22   34 female
15      36    26   38 female
16      37    26   37 female
17      34    25   38 female
18      36    26   37 female
19      38    28   40 female
20      35    23   35 female
```


The distance matrix of chest, waist and hip measurements among the 20 individuals is: (partly shown)

```
> dm <- dist(measure[, c("chest", "waist", "hips")])
> round(dm, 2)
```

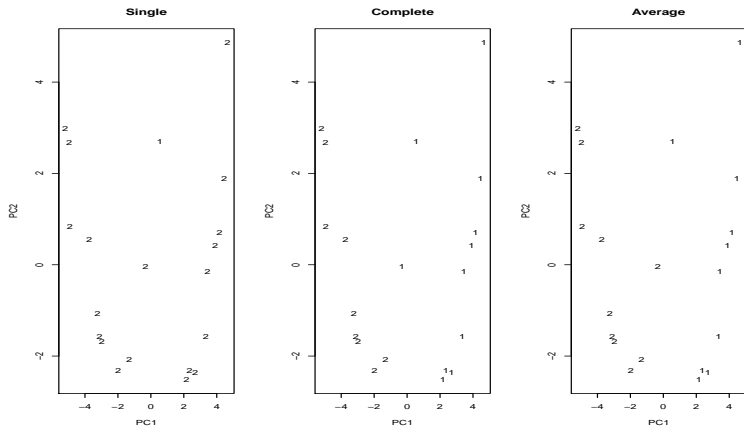
	1	2	3	4	5	6	7	8	9	10	11	12
2	6.16											
3	5.66	2.45										
4	7.87	2.45	4.69									
5	4.24	5.10	3.16	7.48								
6	11.00	6.08	5.74	7.14	7.68							
7	12.04	5.92	7.00	5.00	10.05	5.10						
8	8.94	3.74	4.00	3.74	7.07	5.74	4.12					
9	7.81	3.61	2.24	5.39	4.58	3.74	5.83	3.61				
10	10.10	4.47	4.69	5.10	7.35	2.24	3.32	3.74	3.00			
11	7.00	8.31	6.40	9.85	5.74	11.05	12.08	8.06	7.48	10.25		
12	7.35	7.07	5.48	8.25	6.00	9.95	10.25	6.16	6.40	8.83	2.24	
13	7.81	8.54	7.28	9.43	7.55	12.08	11.92	7.81	8.49	10.82	2.83	2.24
14	8.31	11.18	9.64	12.45	8.66	14.70	15.30	11.18	11.05	13.75	3.74	5.20
15	7.48	6.16	4.90	7.07	6.16	9.22	9.00	4.90	5.74	7.87	3.61	1.41
16	7.07	6.00	4.24	7.35	5.10	8.54	9.11	5.10	5.00	7.48	3.00	1.41
17	7.81	7.68	6.71	8.31	7.55	11.40	10.77	6.71	7.87	9.95	3.74	2.24
18	6.71	6.08	4.58	7.28	5.39	9.27	9.49	5.39	5.66	8.06	2.83	1.00
19	9.17	5.10	4.47	5.48	7.07	6.71	5.74	2.00	4.12	5.10	6.71	4.69
20	7.68	9.43	7.68	10.82	7.00	12.41	13.19	9.11	8.83	11.53	1.41	3.00

We first apply the single, complete and average linkage:



- 
- We could cut the dendrogram with the large size change of relative distance - it generally occurs at $k = 2$.
 - In this particular dataset, since we know that it consists of measurements on ten men and ten women, we are motivated to look at the two-group solution. Thus we could use this “subjective” way to determine the number of clusters.

To check whether the clusterings are reasonable, we could check the plots of the first two principal components labeled by clusters:



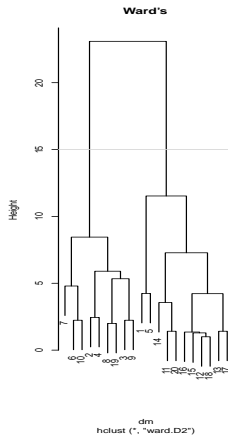
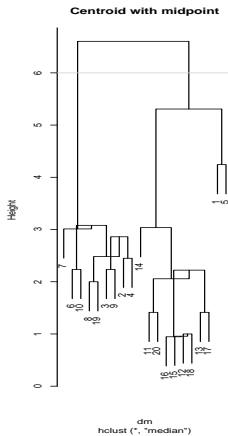
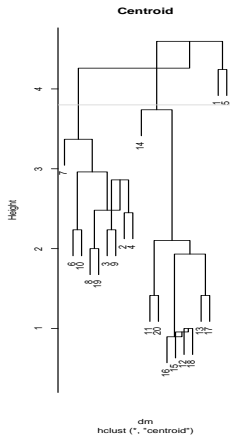
Comments:

- The single linkage plot demonstrates its “chaining” problem, which refers to the tendency to incorporate intermediate points between clusters into an existing cluster rather than initiating a new one.
- Thus single linkage solutions often contain long “straggly” clusters that do not give a useful description of the data.
- The two-group solutions from complete linkage and average linkage are similar and on the whole place the men (observations 1 to 10) together in one cluster and the women (observations 11 to 20) in the other.

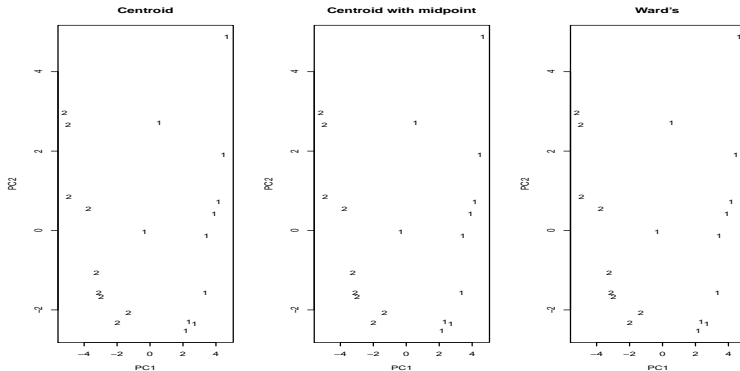
The relevant R code is provided below:

```
> ## conduct the single, complete and average linkage methods
> layout(matrix(1:3, nr = 1), height = c(2, 1))
> plot(cs <- hclust(dm, method = "single"), main = "Single")
> abline(h = 3.8, col = "lightgrey")
>
> plot(cc <- hclust(dm, method = "complete"), main = "Complete")
> abline(h = 10, col = "lightgrey")
>
> plot(ca <- hclust(dm, method = "average"), main = "Average")
> abline(h = 7.8, col = "lightgrey")
>
> ## plot the first two principal components labeled by clusters
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Single")
> lab <- cutree(cs, h = 3.8) # cluster labels of individuals
> text(body_pc$scores[,1:2], labels = lab, cex=1)
>
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Complete")
> lab <- cutree(cc, h = 10)
> text(body_pc$scores[,1:2], labels = lab, cex=1)
>
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Average")
> lab <- cutree(ca, h = 7.8)
> text(body_pc$scores[,1:2], labels = lab, cex=1)
```

The last three hierarchical methods can also be conducted:



The principal component plots with clustering labels are:



Thus the last three method yield identical clustering.

The relevant R code is provided below:

```
> ## conduct the centroid, midpoint, and Ward's methods
> layout(matrix(1:3, nr = 1), height = c(2, 1))
> plot(ct <- hclust(dm, method = "centroid"), main = "Centroid")
> abline(h = 3.8, col = "lightgrey")
>
> plot(ctm <- hclust(dm, method = "median"), main = "Centroid with midpoint")
> abline(h = 6, col = "lightgrey")
>
> plot(ward <- hclust(dm, method = "ward.D2"), main = "Ward's")
> abline(h = 15, col = "lightgrey")
>
> ## plot the first two principal components labeled by clusters
> layout(matrix(1:3, nr = 1), height = c(2, 1))
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Centroid")
> lab <- cutree(ct, k=2) # cluster labels of individuals
> text(body_pc$scores[,1:2], labels = lab, cex=1)
>
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Centroid with midpoint")
> lab <- cutree(ctm, k=2)
> text(body_pc$scores[,1:2], labels = lab, cex=1)
>
> plot(body_pc$scores[,1:2], type = "n", xlab = "PC1", ylab = "PC2", main = "Ward's")
> lab <- cutree(ward, k=2)
> text(body_pc$scores[,1:2], labels = lab, cex=1)
```

Outline

3 Nonhierarchical Methods: Partitioning

- k -Means Method

- Application Example with R Implementation

Introduction to Partitioning

- With hierarchical techniques, once a grouping is made, an item cannot be moved into another group. Thus we now introduce a nonhierarchical approaches - partitioning.
- Using partitioning, the observations are separated into k clusters without using a hierarchical approach.
- The most commonly used partitioning approach is the k -means method.

k -Means Method: Definition

The **k -means clustering** seeks to partition the n items $\mathbf{y}_i = (y_{i1}, \dots, y_{ip})'$, $i = 1, \dots, n$, in a set of k groups, (G_1, \dots, G_k) , that minimizes the within-group sum of squares (WGSS) over all variables, i.e.

$$WGSS = \sum_{j=1}^p \sum_{l=1}^k \sum_{i \in G_l} (y_{ij} - \bar{y}_j^{(l)})^2,$$

where $\bar{y}_j^{(l)} = \sum_{i \in G_l} y_{ij} / n_l$ is the mean of the individuals in group G_l on variable j .

k -Means Method

- An attractive strategy would be to examine all possible ways to partition n items into k clusters and find the optimal clustering according to some criterion.
- However, the number of possible partitions is prohibitively large for even moderate values of n and k . Thus we seek simpler techniques.

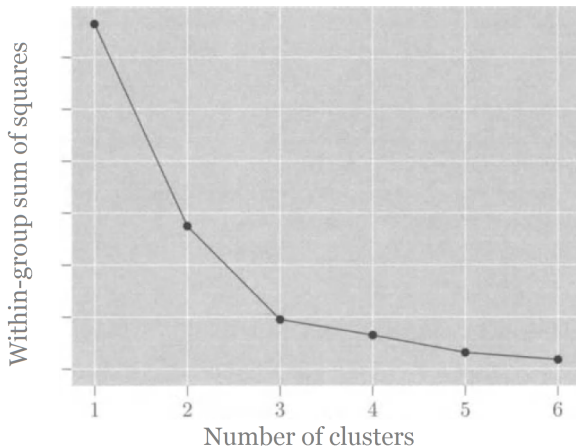
Steps of k -Means Method

The procedure of k -means method to cluster n items into $k < n$ groups, with a predetermined k , is as follows:

- 1 Select k items to serve as “cluster seeds”.
- 2 Each point in the data set is assigned to the cluster with the nearest seed, and the cluster seeds are updated by the centroid of the new cluster.
- 3 After all items are assigned to clusters, each item is examined to see if it is closer to the centroid of another cluster than to the centroid of its own cluster. If so, the item is moved to the new cluster and the two cluster centroids are updated.
- 4 Continue 3 until no reassignment is possible.

k -Means Method: Choosing k

- To determine the number of clusters k in the hierarchical clustering, we could find the “large” change of relative distance in the dendrogram.
- In the k -means method, however, there is no dendrogram and distances available.
- As the k -means methods concerns about the within-group sum of squares (WGSS), we choose k for a reasonably small (but not the smallest) WGSS.
- Graphically, we could search the “scree graph” of WGSS against k for a knee point to serve as the optimal k .



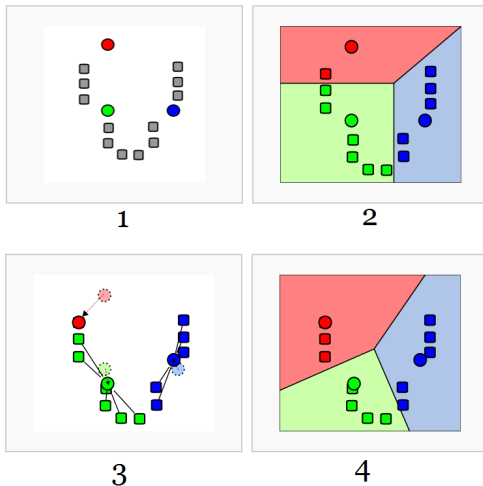
The above scree graph suggests 3 clusters, as 3 is the knee point in the plot.

k -Means Method: Choosing Initial Seeds

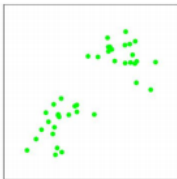
There are various ways to choose seeds:

- Select k items at random separated by a specified minimum distance.
- Choose the first k points in the data set subject to a minimum distance requirement.
- Select the k points that are mutually farthest apart.
- Specify k regularly spaced points in a gridlike pattern - these would not be actual data points.

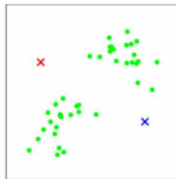
Illustration of k -Means Method



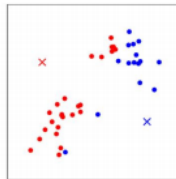
Another example:



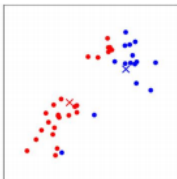
(a)



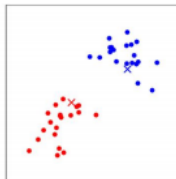
(b)



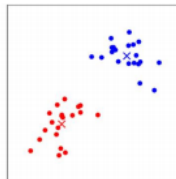
(c)



(d)



(e)



(f)

k -Means Method: Remarks

- The k -means procedure is somewhat sensitive to the initial seeds. It might be advisable to try the procedure again with another choice of seeds. If different initial choices of seeds produce widely different final clusters, or if convergence is extremely slow, there may be no natural clusters in the data.
- The k -means partitioning method can also be used as a possible improvement on hierarchical techniques. We first cluster the items using a hierarchical method and then use the centroids of these clusters as seeds for a k -means approach, which will allow points to be reallocated from one cluster to another.

Outline


3 Nonhierarchical Methods: Partitioning

- k -Means Method


- Application Example with R Implementation

k -Means Method: R Example


Example: A study was conducted for Romano-British pottery made in three different regions (region 1 contains kiln 1, region 2 contains kilns 2 and 3, and region 3 contains kilns 4 and 5). The dataset consists of the chemical analysis results on 45 pots. One question that might be posed is whether the chemical profiles of each pot suggest different types of pots and if any such types are related to kiln or region.



Al2O3	Fe2O3	MgO	CaO	Na2O	K2O	TiO2	MnO	BaO	kiln
18.8	9.52	2.00	0.79	0.40	3.20	1.01	0.077	0.015	1
16.9	7.33	1.65	0.84	0.40	3.05	0.99	0.067	0.018	1
18.2	7.64	1.82	0.77	0.40	3.07	0.98	0.087	0.014	1
16.9	7.29	1.56	0.76	0.40	3.05	1.00	0.063	0.019	1
17.8	7.24	1.83	0.92	0.43	3.12	0.93	0.061	0.019	1
18.8	7.45	2.06	0.87	0.25	3.26	0.98	0.072	0.017	1
16.5	7.05	1.81	1.73	0.33	3.20	0.95	0.066	0.019	1
18.0	7.42	2.06	1.00	0.28	3.37	0.96	0.072	0.017	1
15.8	7.15	1.62	0.71	0.38	3.25	0.93	0.062	0.017	1
14.6	6.87	1.67	0.76	0.33	3.06	0.91	0.055	0.012	1
13.7	5.83	1.50	0.66	0.13	2.25	0.75	0.034	0.012	1
14.6	6.76	1.63	1.48	0.20	3.02	0.87	0.055	0.016	1
14.8	7.07	1.62	1.44	0.24	3.03	0.86	0.080	0.016	1
17.1	7.79	1.99	0.83	0.46	3.13	0.93	0.090	0.020	1
16.8	7.86	1.86	0.84	0.46	2.93	0.94	0.094	0.020	1
15.8	7.65	1.94	0.81	0.83	3.33	0.96	0.112	0.019	1



Al2O3	Fe2O3	MgO	CaO	Na2O	K2O	TiO2	MnO	BaO	kiln
18.6	7.85	2.33	0.87	0.38	3.17	0.98	0.081	0.018	1
16.9	7.87	1.83	1.31	0.53	3.09	0.95	0.092	0.023	1
18.9	7.58	2.05	0.83	0.13	3.29	0.98	0.072	0.015	1
18.0	7.50	1.94	0.69	0.12	3.14	0.93	0.035	0.017	1
17.8	7.28	1.92	0.81	0.18	3.15	0.90	0.067	0.017	1
14.4	7.00	4.30	0.15	0.51	4.25	0.79	0.160	0.019	2
13.8	7.08	3.43	0.12	0.17	4.14	0.77	0.144	0.020	2
14.6	7.09	3.88	0.13	0.20	4.36	0.81	0.124	0.019	2
11.5	6.37	5.64	0.16	0.14	3.89	0.69	0.087	0.009	2
13.8	7.06	5.34	0.20	0.20	4.31	0.71	0.101	0.021	2
10.9	6.26	3.47	0.17	0.22	3.40	0.66	0.109	0.010	2
10.1	4.26	4.26	0.20	0.18	3.32	0.59	0.149	0.017	2
11.6	5.78	5.91	0.18	0.16	3.70	0.65	0.082	0.015	2
11.1	5.49	4.52	0.29	0.30	4.03	0.63	0.080	0.016	2
13.4	6.92	7.23	0.28	0.20	4.54	0.69	0.163	0.017	2
12.4	6.13	5.69	0.22	0.54	4.65	0.70	0.159	0.015	2
13.1	6.64	5.51	0.31	0.24	4.89	0.72	0.094	0.017	2



Al2O3	Fe2O3	MgO	CaO	Na2O	K2O	TiO2	MnO	BaO	kiln
11.6	5.39	3.77	0.29	0.06	4.51	0.56	0.110	0.015	3
11.8	5.44	3.94	0.30	0.04	4.64	0.59	0.085	0.013	3
18.3	1.28	0.67	0.03	0.03	1.96	0.65	0.001	0.014	4
15.8	2.39	0.63	0.01	0.04	1.94	1.29	0.001	0.014	4
18.0	1.50	0.67	0.01	0.06	2.11	0.92	0.001	0.016	4
18.0	1.88	0.68	0.01	0.04	2.00	1.11	0.006	0.022	4
20.8	1.51	0.72	0.07	0.10	2.37	1.26	0.002	0.016	4
17.7	1.12	0.56	0.06	0.06	2.06	0.79	0.001	0.013	5
18.3	1.14	0.67	0.06	0.05	2.11	0.89	0.006	0.019	5
16.7	0.92	0.53	0.01	0.05	1.76	0.91	0.004	0.013	5
14.8	2.74	0.67	0.03	0.05	2.15	1.34	0.003	0.015	5
19.1	1.64	0.60	0.10	0.03	1.75	1.04	0.007	0.018	5

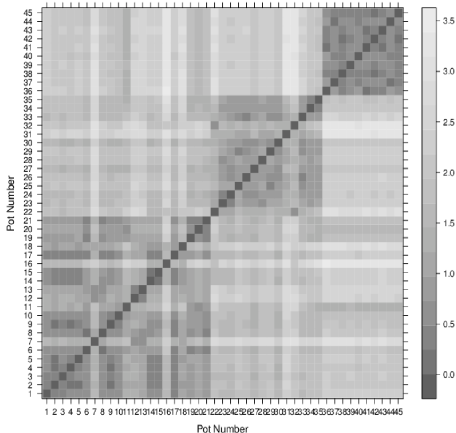
To conduct the cluster analysis, we first check the distance matrix (partly shown):

```
> pottery_dist <- dist(pots <- scale(pottery[, colnames(pottery) != "kiln"], center = FALSE))
> pottery_dist
```

	1	2	3	4	5	6	7
2	0.44792698						
3	0.33852414	0.36477265					
4	0.49647084	0.14193206	0.42850238				
5	0.53143290	0.20798180	0.49475386	0.28544740			
6	0.62037802	0.53791475	0.58237680	0.58343192	0.63479583		
7	1.47096656	1.31522603	1.46933715	1.42985863	1.22416990	1.29393983	
8	0.62124388	0.50007272	0.58796867	0.58756246	0.54914197	0.22209057	1.08939448
9	0.51215588	0.24554901	0.40932286	0.19680282	0.39450489	0.55825964	1.49505655
10	0.65070948	0.49511604	0.53039078	0.51800136	0.61969954	0.56896487	1.47774056
11	1.32122209	1.16636848	1.23006557	1.15128161	1.27407627	0.94332022	1.83234944
12	1.34374953	1.17242219	1.31758442	1.26960823	1.15067249	0.98564299	0.62678891
13	1.19518172	1.05388823	1.14723804	1.16736340	1.04287526	0.89666811	0.59184300
14	0.49739638	0.38562830	0.42976723	0.43105501	0.39763302	0.75967051	1.40961709
15	0.52000136	0.41153238	0.44032273	0.46001759	0.43972497	0.78363302	1.40791406
16	1.53623890	1.52167016	1.48790489	1.54111173	1.46492847	1.98287704	2.19026430
17	0.36676824	0.31919530	0.34101725	0.40232534	0.36422334	0.45911216	1.29543450
18	1.04684827	0.91278055	1.04399169	1.00474884	0.79337053	1.21062199	0.98590335
19	0.94618089	0.93044287	0.91944694	0.95775524	1.04151768	0.41732421	1.49445203
20	1.11462907	1.03061182	1.12838982	1.00550521	1.12448730	0.66941405	1.70673717
21	0.83664573	0.74336926	0.79274423	0.75953860	0.85141722	0.28243473	1.43125907
22	1.71787742	1.81040549	1.62758791	1.79340022	1.87007136	1.90997572	2.76013088

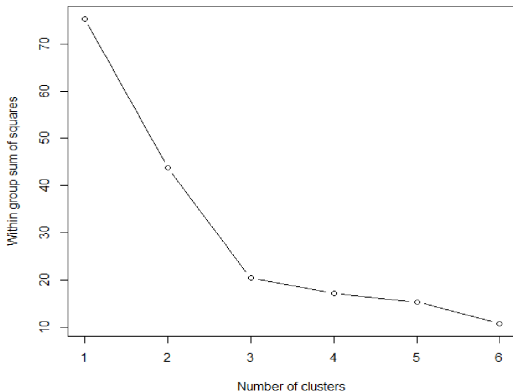
If we suspect the clusters are related to the regions or kilns, we may inspect the image plot of the distance matrix:

```
> library("lattice")  
> levelplot(as.matrix(pottery_dist), xlab = "Pot Number", ylab = "Pot Number")
```



To determine the number of clusters, we may plot the “scree graph” for the within-group sum of squares:

```
> n <- nrow(pots)
> wgss <- rep(0, 6)
> for (i in 1:6) {wgss[i] <- sum(kmeans(pots, centers = i)$withinss)}
> plot(1:6, wgss, type = "b", xlab = "Number of clusters", ylab = "Within group sum of squares")
```



Both the image plot and the scree graph suggests 3 clusters.
Thus we conduct the corresponding *k*-means approach:

```
> (kmeans_3<-kmeans(pots, centers=3)) #k-means method with 3 clusters
K-means clustering with 3 clusters of sizes 10, 21, 14

Cluster means:
      Al2O3      Fe2O3      MgO      CaO      Na2O      K2O      TiO2      MnO      BaO
1 1.1014781 0.2559212 0.2091007 0.0565216 0.1680445 0.6038928 1.1275412 0.03751705 0.9432930
2 1.0499133 1.1793602 0.6019426 1.3609353 1.1391251 0.9271614 1.0364747 0.83408451 1.0106711
3 0.7716996 0.9855595 1.5610211 0.3105583 0.7437263 1.2513690 0.7548525 1.37925420 0.9390819

Clustering vector:
[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1

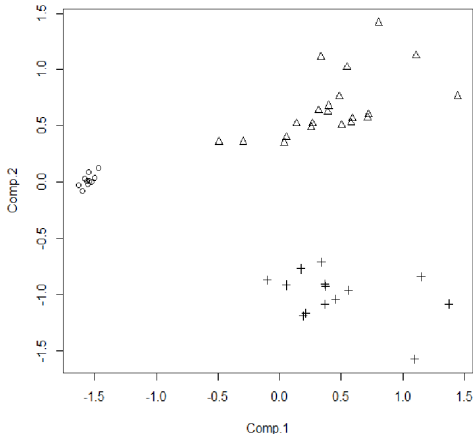
Within cluster sum of squares by cluster:
[1] 1.110246 11.494967 7.775663
(between_SS / total_SS = 73.0 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"      "withinss"      "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

From the plot the first two principal components of the data, we can see this clustering performs well:

```
> pots_pca <- princomp(pots)
> plot(pots_pca$scores[, 1:2], pch = kmeans_3$cluster)
```



Furthermore, by observing the cluster each pot belongs to, the clusters found actually correspond to pots from three different regions, thus the region where the pot is found might be used to categorize the pot.

[illegible]

Summary and Take-home Messages

- What is cluster analysis and how does it differ from the classification?
- What is the idea of hierarchical cluster analysis?
- What types of clustering do we have? What are the pros and cons of each type?
- How to conduct the k -means method?
- How to choose the number of clusters?