MONASH University

ETC3250

Business Analytics

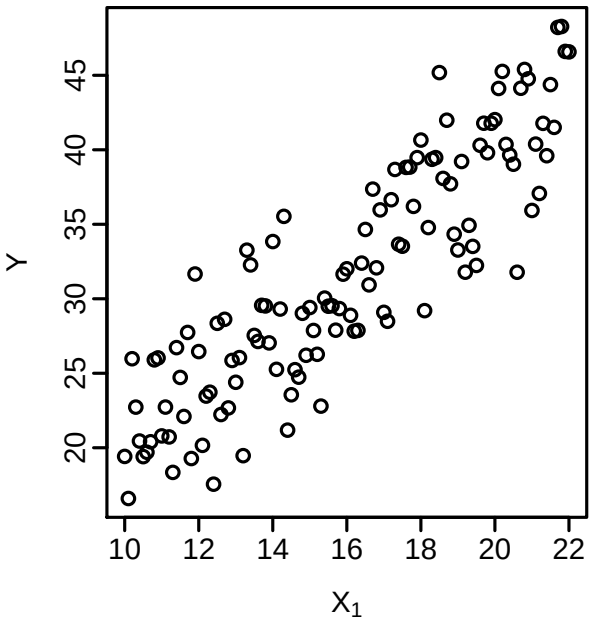
Week 10
Clustering

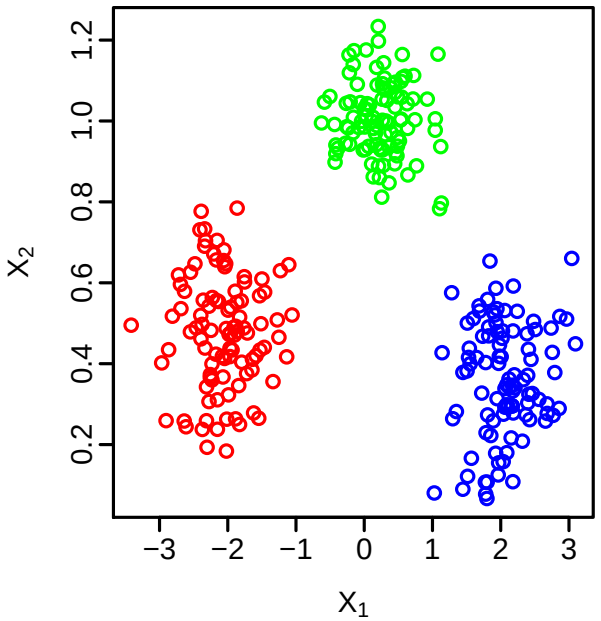
2/5 October 2017

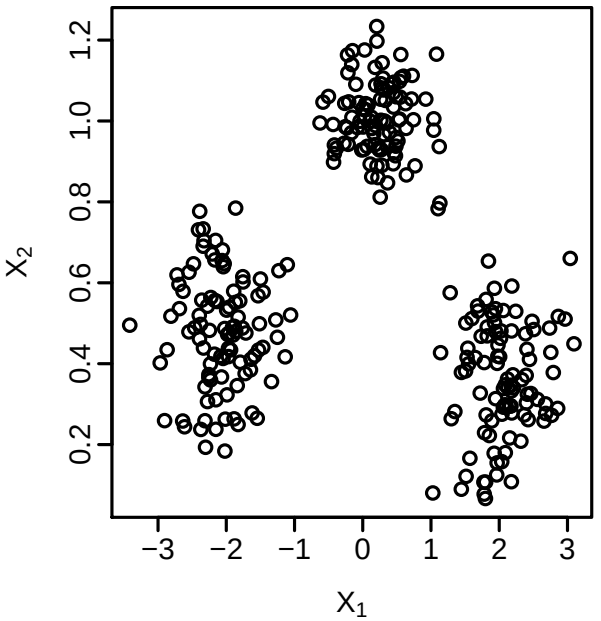
Outline

Week	Topic	Chapter	Lecturer
1	Introduction to business analytics & R	1	Souhaib
2	Statistical learning	2	Souhaib
3	Regression for prediction	3,7	Tas & David
4	Classification	4	Souhaib
5	Classification	4, 9	Souhaib
6	Model selection and resampling methods	5	Souhaib
7	Dimension reduction	6,10	Souhaib
8	Advanced regression	6	Souhaib
9	Tree-based methods	8	Souhaib
	Semester break		
10	Clustering	10	Souhaib
11	Visualization and data wrangling		Souhaib
12	Project presentation		Souhaib

Statistical learning problems







Unsupervised learning

- **Unsupervised learning** is often performed as part of an **exploratory data analysis**.
- Unsupervised learning is often much **more challenging than supervised learning**. The exercise tends to be more **subjective**, and there is no simple goal for the analysis, such as prediction of a response
- **Hard to assess** the results obtained from unsupervised learning methods
- Techniques for unsupervised learning are of **growing importance** in a number of fields
- Examples of unsupervised learning methods?

Unsupervised learning methods

Both **PCA** and **clustering** seek to simplify the data via a small number of summaries, but their mechanisms are different:

- **PCA** (unsupervised dimension reduction method) looks to find a low-dimensional representation of the observations that explain a large fraction of the variance.
- **Clustering** looks to find homogeneous subgroups among the observations.

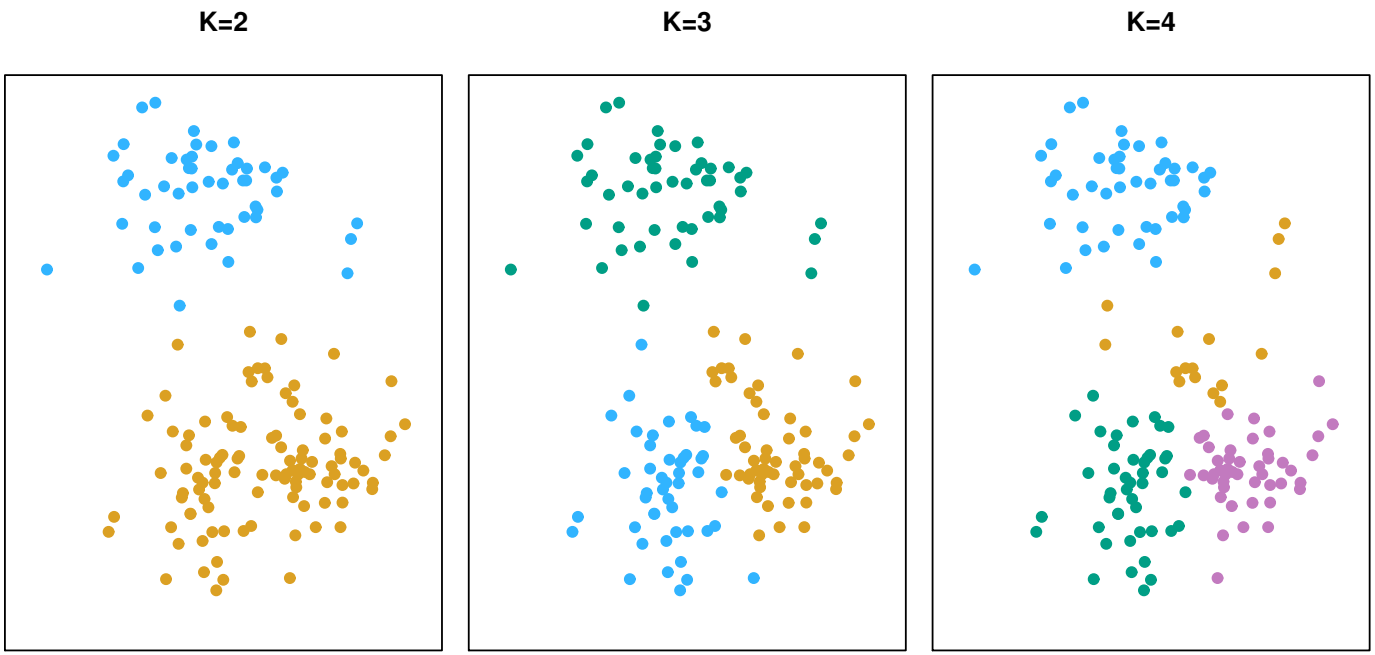
Clustering methods

- **K-means clustering**
  - We seek to partition the observations into K ($K \leq n$) clusters.
- **Hierarchical clustering**
  - We do not know in advance how many clusters we want. We consider all possible number of clusters, from 1 to n.

K-means clustering

Find K clusters $C_1, \ldots, C_K$ where

- $C_1 \cup C_2 \cup C_K = \{1, \ldots, n\}$
- $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$



K-means clustering

For K-means, good clustering means small **total within-cluster variation**:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

where $W(C_k)$ is the within-cluster variation for cluster C_k, i.e. the amount by which the *observations within a cluster* differ from each other.

K-means within-cluster variation

There are many possible ways to define the **within-cluster variation**, but by far the most common choice involves **squared Euclidean distance**:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

K-means optimization problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

The number of possible assignments of n data points into K clusters is

$$S(n,K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n \approx K^n,$$

which is the *Stirling numbers of the second kind*.

- $S(n,K)$ is a huge number unless K and n are tiny.
- Fortunately, a very simple algorithm can be shown to provide a **local optimum**—a pretty good solution—to the K-means optimization problem.

K-means optimization problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

The number of possible assignments of n data points into K clusters is

$$S(n,K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n \approx K^n,$$

which is the *Stirling numbers of the second kind*.

- $S(n,K)$ is a huge number unless K and n are tiny.
- Fortunately, a very simple algorithm can be shown to provide a **local optimum**—a pretty good solution—to the K-means optimization problem.

Rewriting the objective

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

where

$$\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$$

$$\Longrightarrow \underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2 \right\}$$

K-means optimization problem

For any $x_1, \ldots, x_m \in \mathbb{R}^p$, the quantity $\sum_{i=1}^m \|x_i - c\|_2^2$ is minimized by $c = \bar{x} = \sum_{i=1}^m x_i$.

So our problem is the same as

$$\underset{C_1,\ldots,C_K,s_1,\ldots,s_K}{\text{minimize}} \left\{ \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - s_{kj})^2 \right\}$$

The K-means clustering algorithm **approximately** minimizes the previous objective by **alternately minimizing** over $C_1, \ldots, C_K$ and $s_1, \ldots, s_K$.

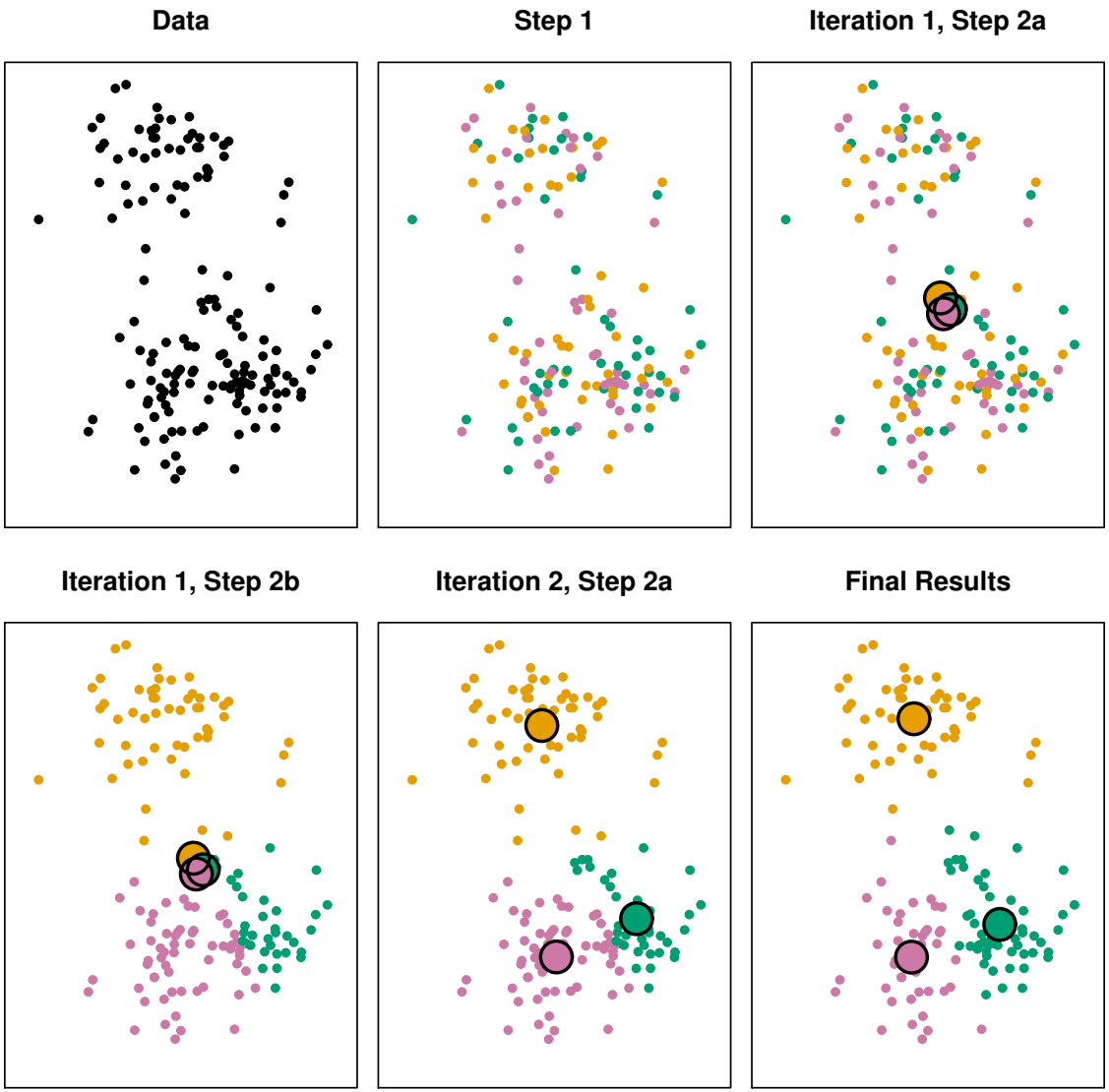
K-means optimization problem

Algorithm 10.1 *K-Means Clustering*

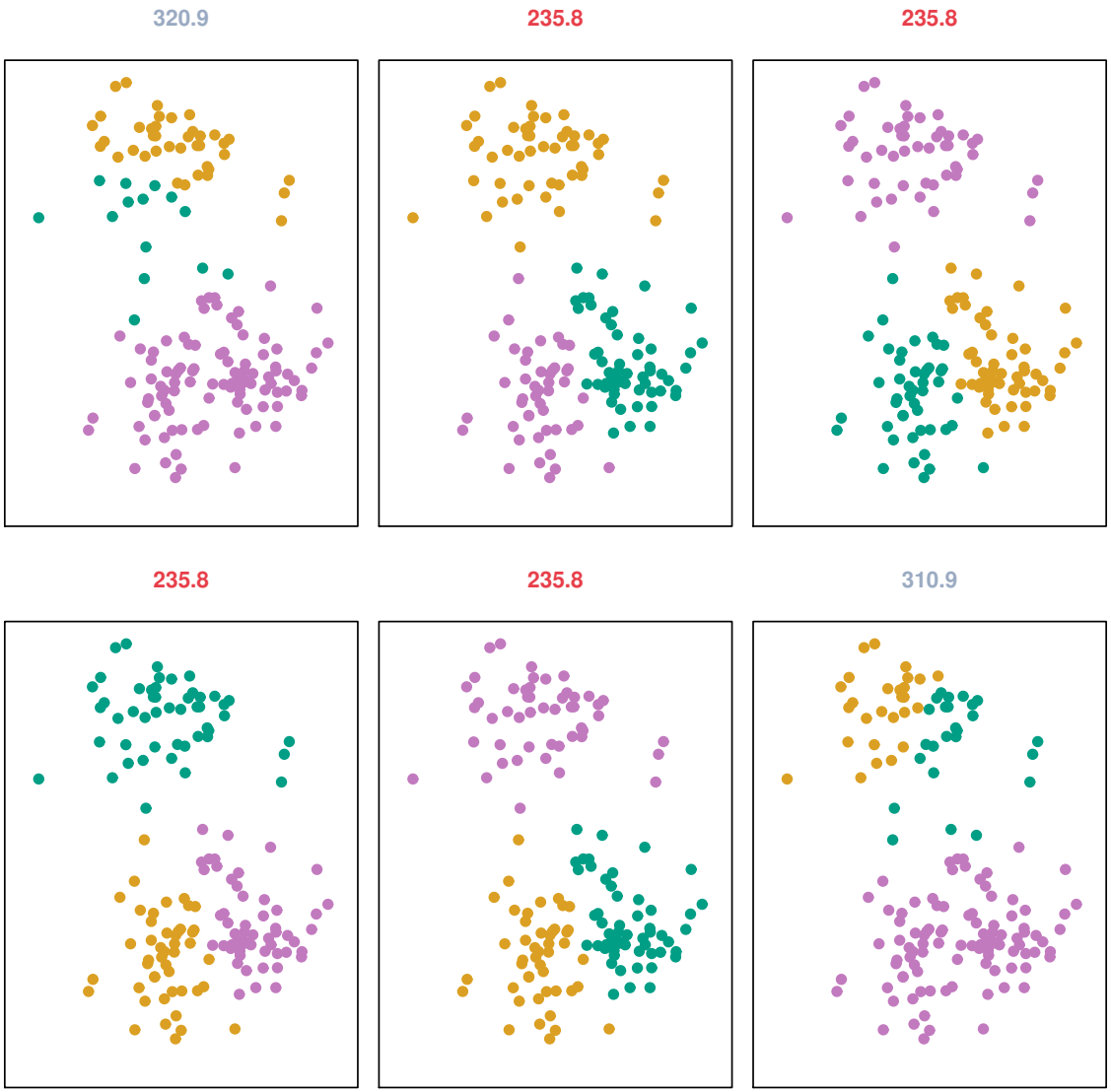
1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

The K-means algorithm is **guaranteed to decrease the value of the objective for each iteration**.

Example


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Random initial clusters



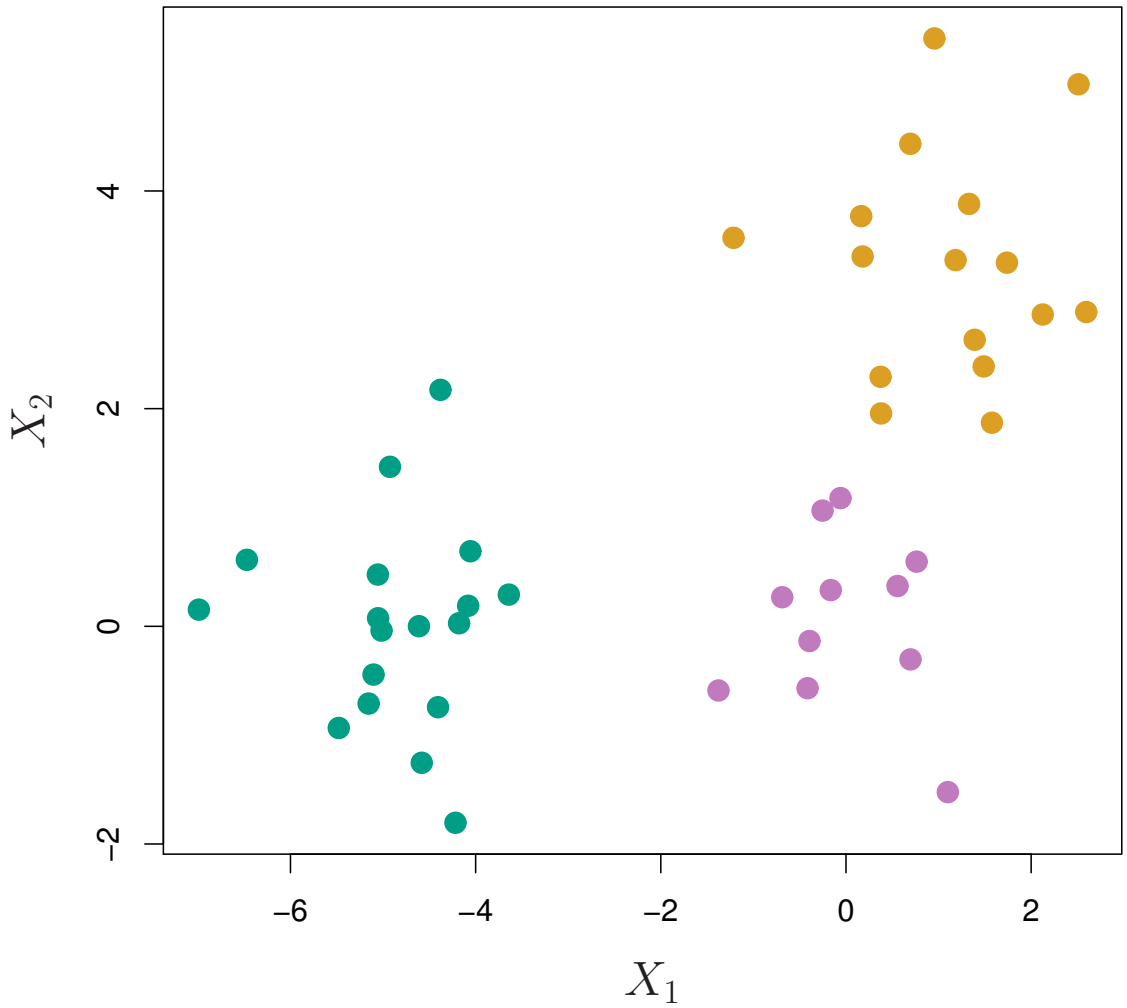
Practical issues with K-means

- Should we standardise the data?
- How many clusters should we use?
  - Objective function is minimal with $K = n$.
- How far from optimal solution?
  - Arbitrarly far with random initial clusters.
  - The clustering quality depends heavily on initial clusters.
  - Better approach: K-means++.

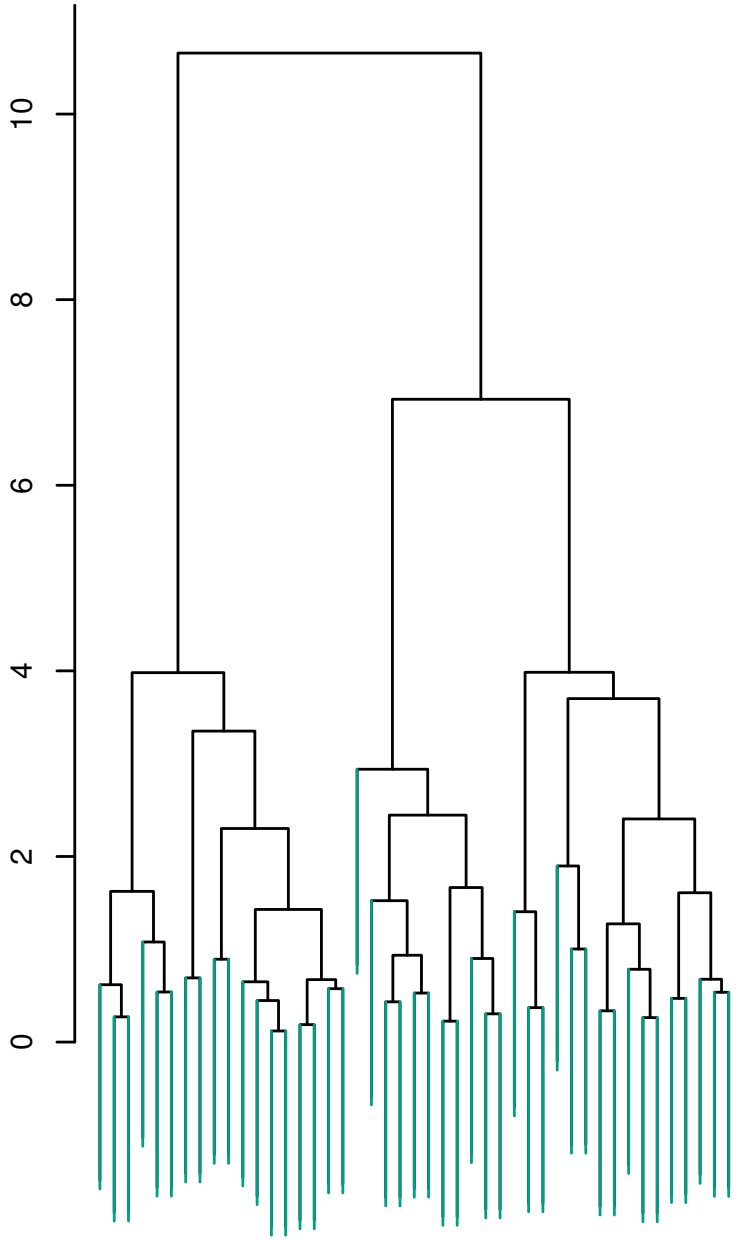
Hierarchical clustering

- One potential disadvantage of K-means clustering is that it requires us to **pre-specify the number of clusters** K.
- Hierarchical clustering is an alternative approach which **does not require** that we commit to a particular choice of K
- Hierarchical clustering has an added advantage over K-means clustering in that it results in an **attractive tree-based representation of the observations**, called a **dendrogram**.

Hierarchical clustering



Dendrogram



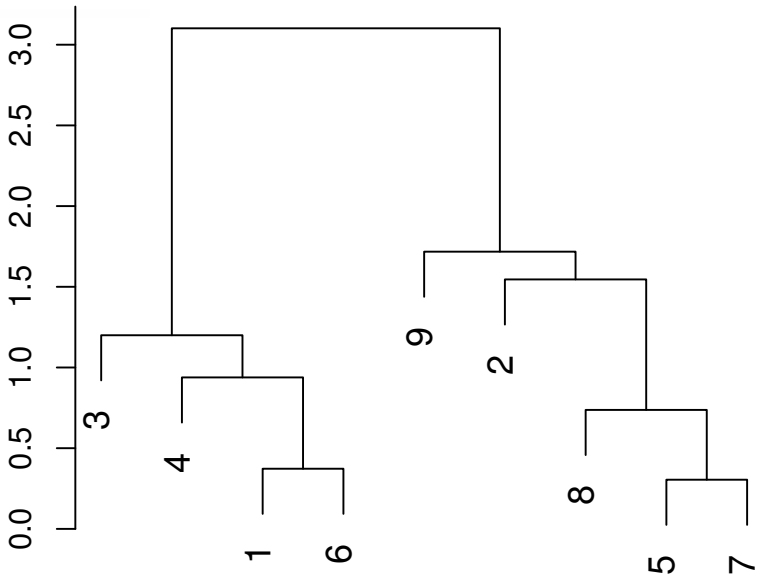
Dendrogram

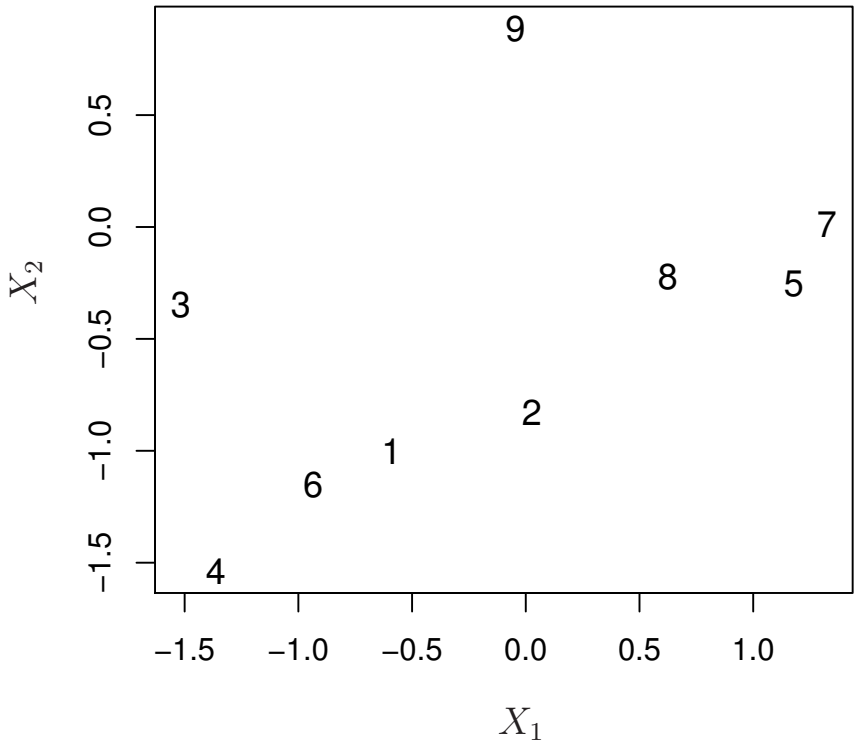
- Each **leaf** of the dendrogram represents one of the 45 observations
- As we move up the tree, some **leaves begin to fuse into branches**. These correspond to observations that are similar to each other.
- As we move higher up the tree, **branches themselves fuse, either with leaves or other branches**. The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other. On the other hand, observations that fuse later (near the top of the tree) can be quite different.

Dendrogram

- For any two observations, we can look for the point in the tree where branches containing those two observations are first fused. **The height of this fusion**, as measured on the *vertical axis*, indicates **how different** the two observations are.
- Thus, observations that fuse at the very **bottom** of the tree are **quite similar** to each other, whereas observations that fuse **close to the top** of the tree will tend to be **quite different**

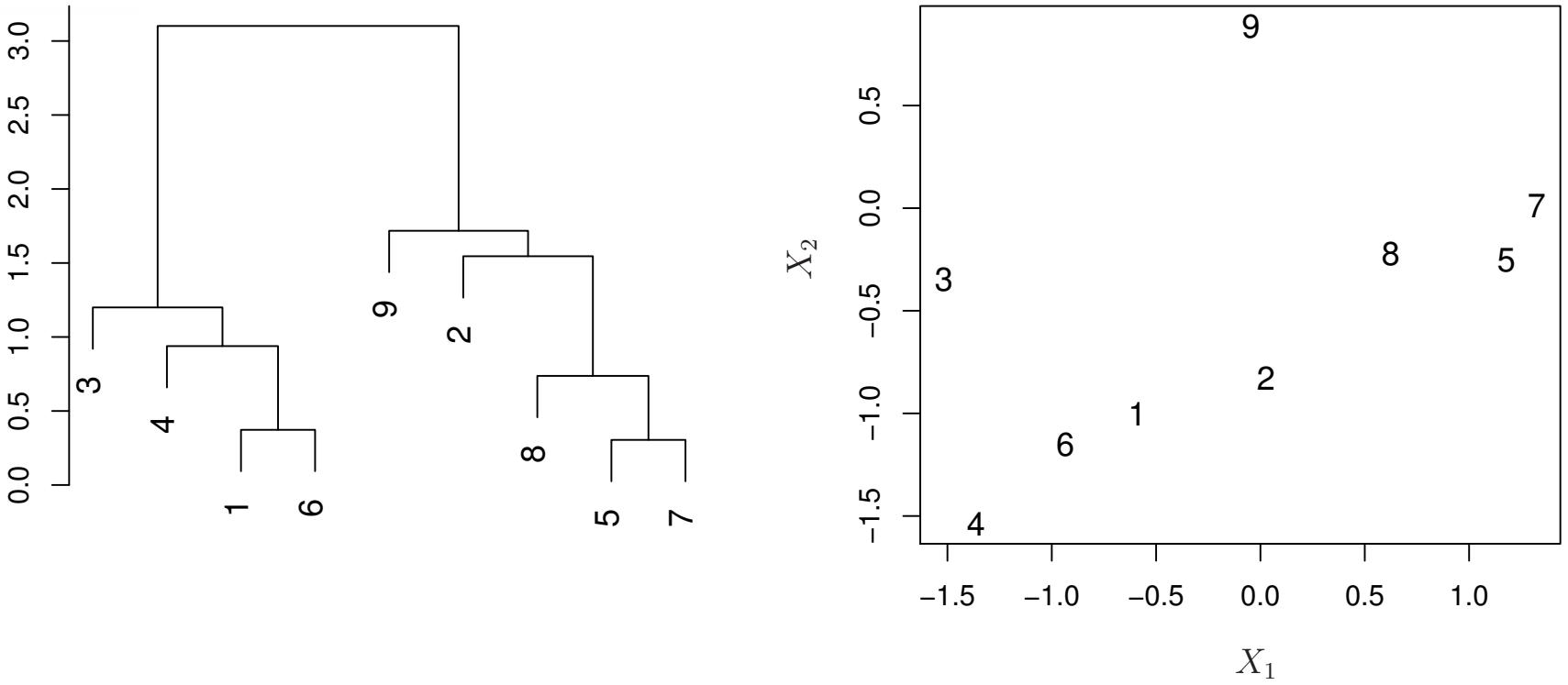
Dendrogram





- Observations 5 and 7, and 1 and 6 are quite similar to each other
- Observations 9 and 2 are located near each other on the dendrogram but incorrect to conclude from the figure that they are quite similar

Dendrogram

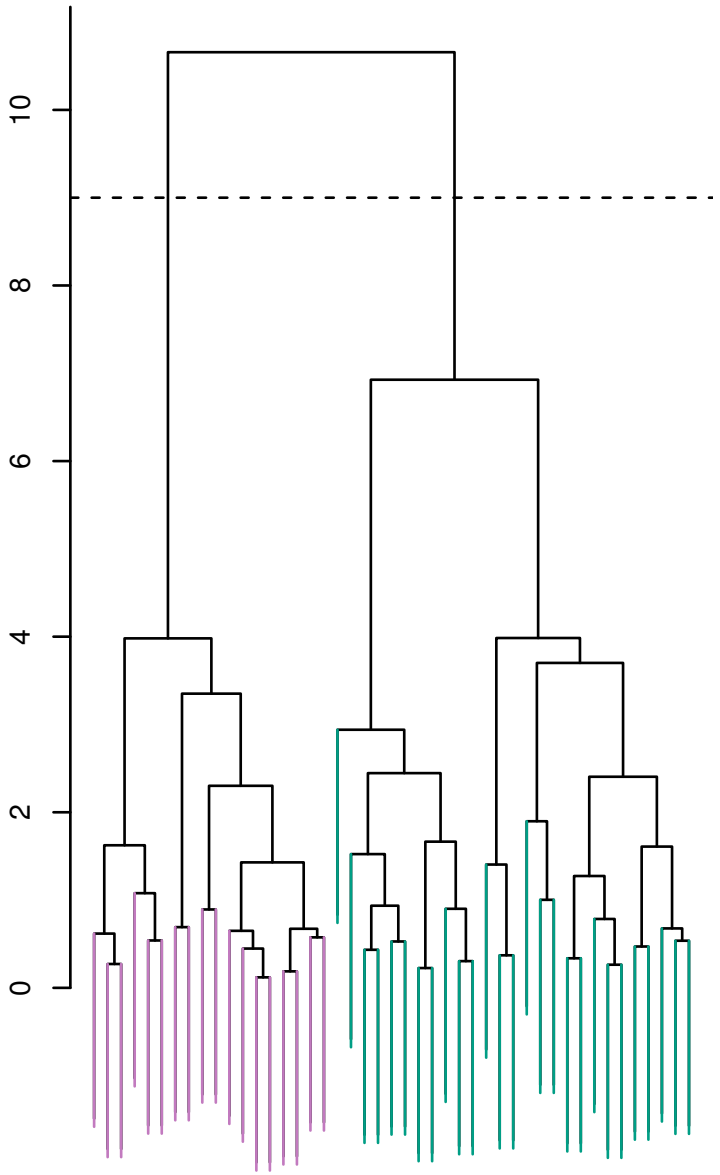


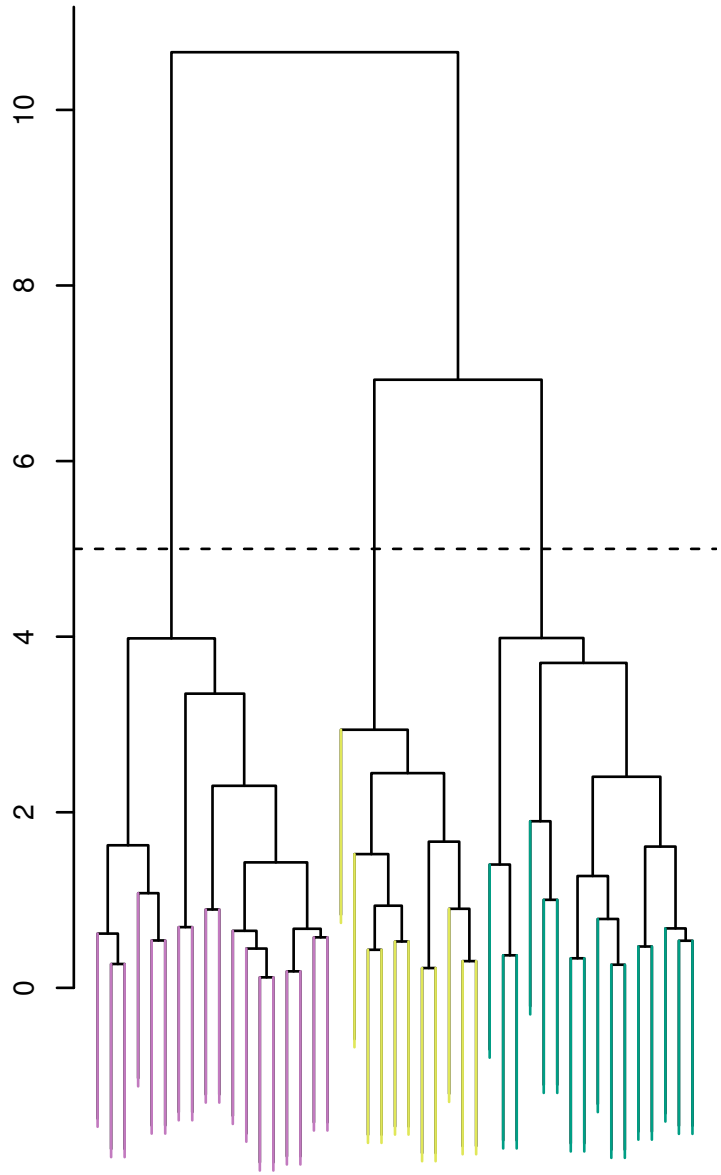
- Observations 5 and 7, and 1 and 6 are quite similar to each other
- Observations 9 and 2 are located near each other on the dendrogram but incorrect to conclude from the figure that they are quite similar

Dendrogram

- Actually, observation 9 is **no more similar** to observation 2 than it is to observations 8, 5, and 7.
- There are 2^{n-1} possible reorderings of the dendrogram, where n is the number of leaves. This is because at each of the $n-1$ points where fusions occur, **the positions of the two fused branches could be swapped without affecting the meaning of the dendrogram**.
- Therefore, we **cannot** draw conclusions about the similarity of two observations based on their proximity along the ***horizontal axis***.
- Rather, we draw conclusions about the similarity of two observations based on the location on the ***vertical axis*** where branches containing those two observations first are fused.

Dendrogram

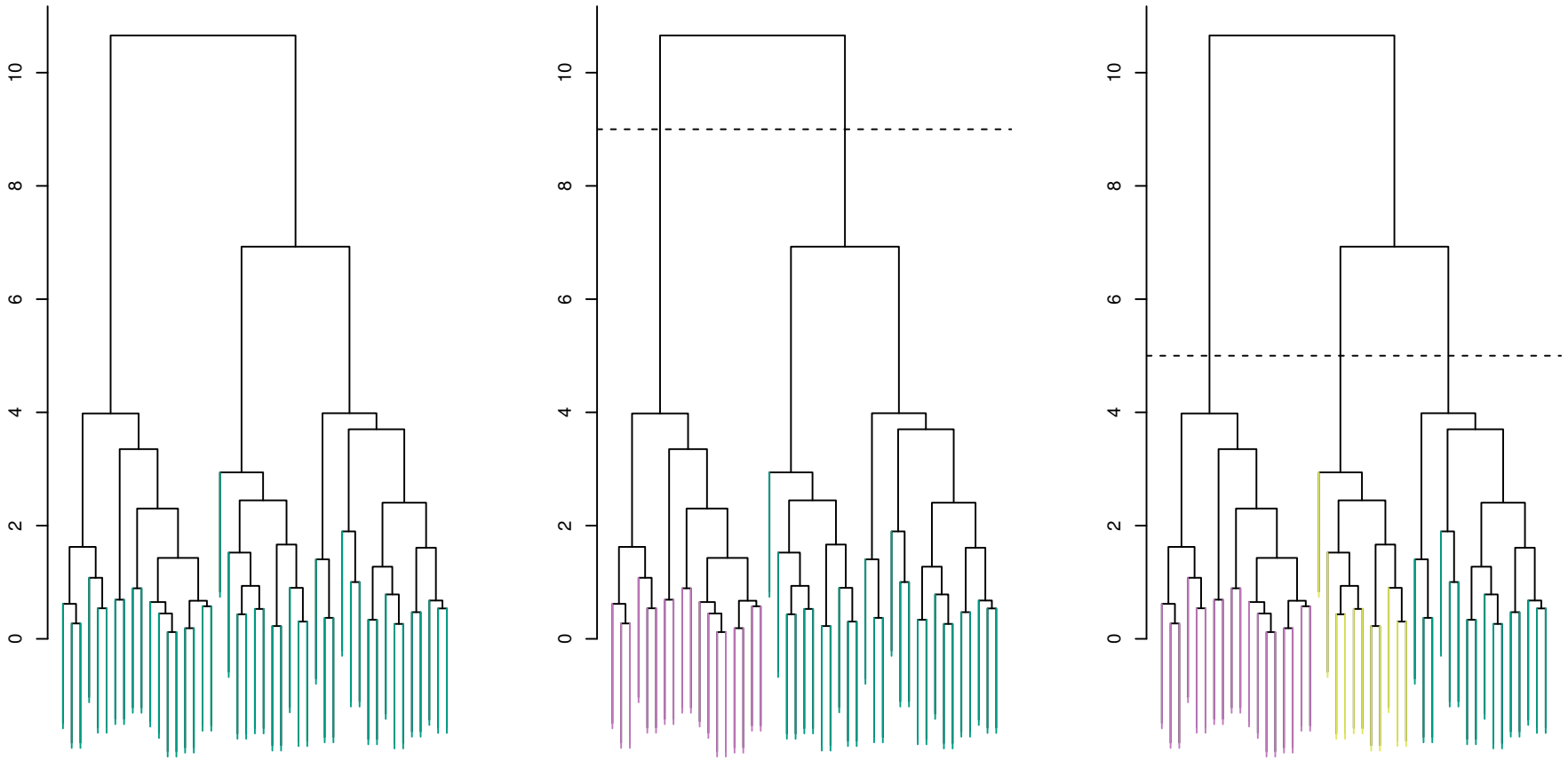




Dendrogram

- The distinct sets of observations **beneath the cut** can be interpreted as clusters
- Cutting the dendrogram **at a height of nine** results in two clusters, shown in distinct colors.
- Cutting the dendrogram **at a height of five** results in three clusters.
- Further cuts can be made as one descends the dendrogram in order to obtain any number of clusters, **between** 1 (corresponding to no cut) and n (corresponding to a cut at height 0, so that each observation is in its own cluster).
- In other words, **the height of the cut** to the dendrogram serves the same role as the K in K-means clustering: it **controls the number of clusters obtained**

Hierarchical clustering



Hierarchical clustering

- In practice, people often look at the dendrogram and **select by eye** a sensible number of clusters, based on the heights of the fusion and the number of clusters desired. However, often the choice of **where to cut** the dendrogram **is not so clear**.
- Also, suppose that our observations correspond to a group of people with a 50–50 split of males and females, evenly split among Americans, Japanese, and French.
- Best division into two groups might split these people by gender, and the best division into three groups might split them by nationality.
- **The true clusters are not nested**, in the sense that the best division into three groups does not result from taking the best division into two groups and splitting up one of those groups.

Hierarchical clustering algorithm

1. We begin by defining some sort of **dissimilarity measure** between each pair of observations. Most often, Euclidean distance is used
2. Each of the n observations is treated as **its own cluster**
3. The two clusters that are most similar to each other are then fused so that there now are $n - 1$ clusters
4. Next the two clusters that are most similar to each other are fused again, so that there now are $n - 2$ clusters
5. The algorithm proceeds in this fashion **until all of the observations belong to one single cluster**, and the dendrogram is complete

Hierarchical clustering algorithm

Algorithm 10.2 *Hierarchical Clustering*

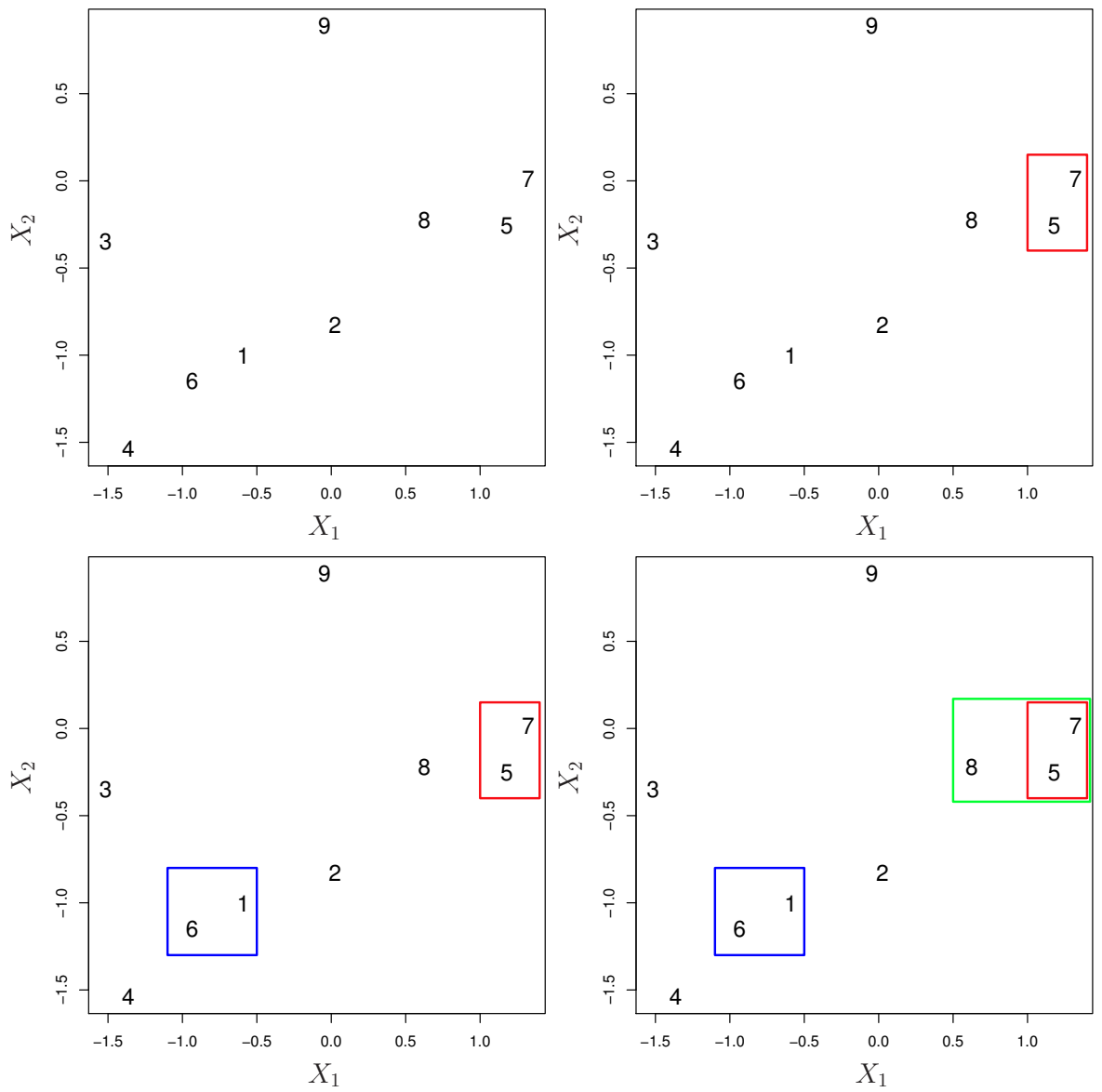
1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Hierarchical clustering



Linkage

We have a concept of the **dissimilarity between pairs of observations**, but how do we define the **dissimilarity between two clusters** if one or both of the clusters contains multiple observations?

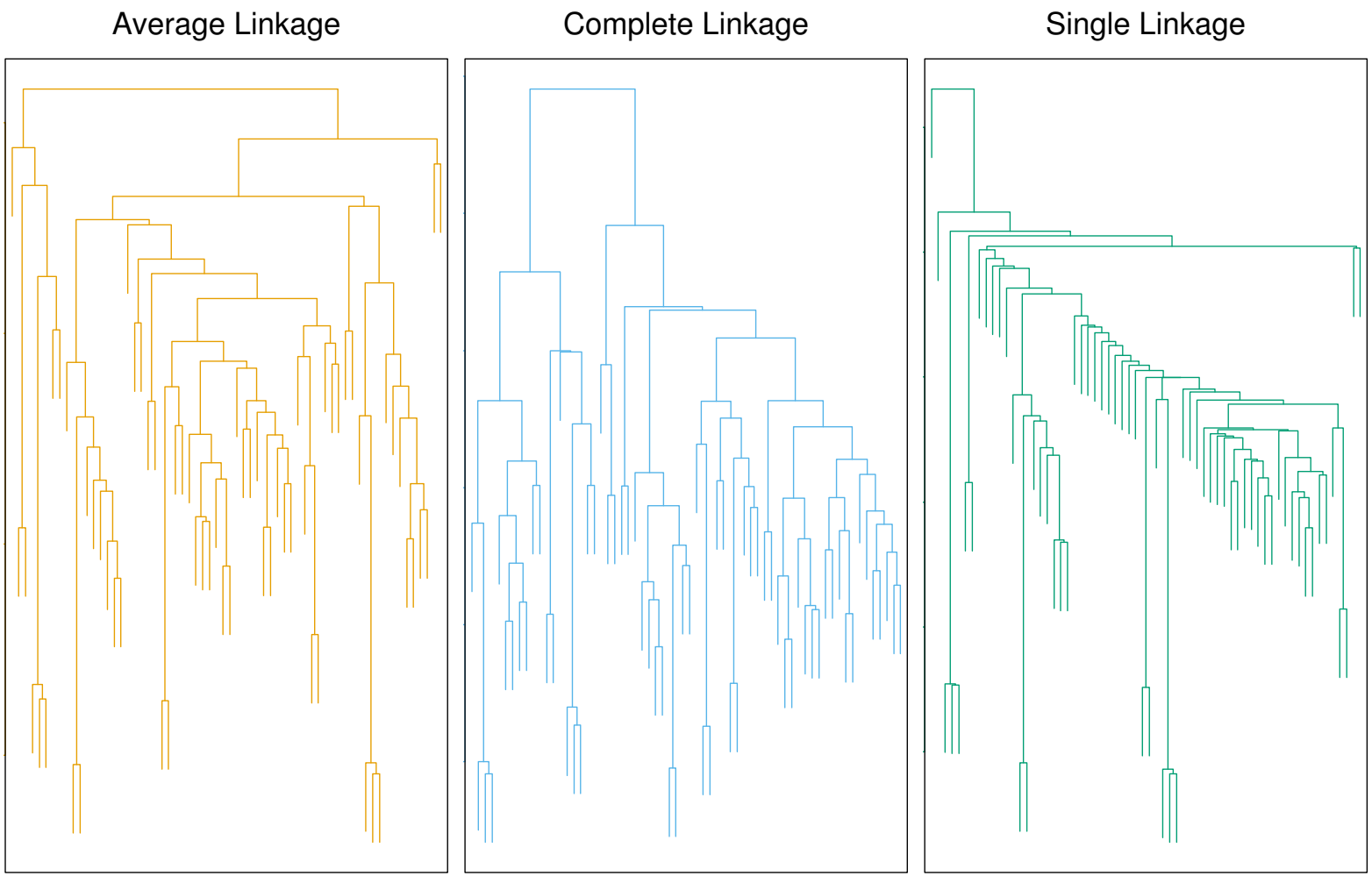
Linkage

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Linkage

- **Average and complete** linkage are generally preferred over single linkage, as they tend to yield more balanced dendrograms.
- **Centroid linkage** is often used in genomics, but suffers from a major drawback in that an inversion can occur, whereby two clusters are fused at a height below either of the individual clusters in the dendrogram. This can lead to difficulties in visualization as well as in interpretation of the dendrogram

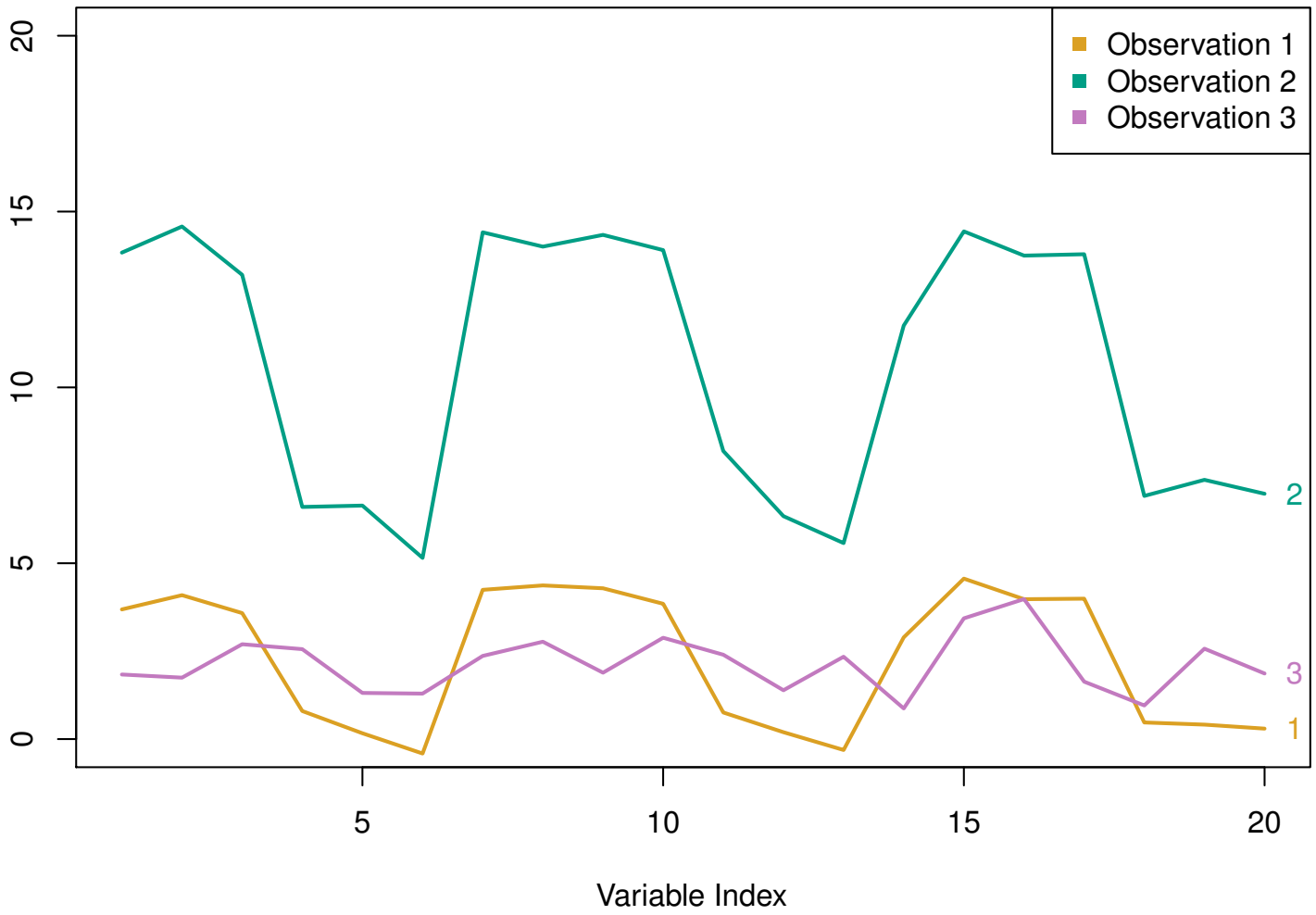
Linkage



Dissimilarity measure

- The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram.
- In general, careful attention should be paid to the type of data being clustered and the scientific question at hand.

Dissimilarity measure



Practical issues in clustering

- Should we standardise the data?
- For K-means clustering:
  - How many clusters should we use?
- For hierarchical clustering:
  - Which dissimilarity measure?
  - What type of linkage?
  - Where should we cut the dendrogram?
- Other considerations
  - Soft clustering (e.g. using mixture models)
  - Clustering high-dimensional data