Module 3: Unsupervised learning, support vector machines and resampling

Principal Component Analysis, **Clustering Algorithms, practical issues in clustering,** support vector classifiers and support vector machines, resampling methods: cross-validation and bootstrapping

Reference: James, G., Witten, D., Hastie, T., Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R., Springer.)

Clustering Methods

Clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.

GOAL: to partition the data set into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other

For instance, suppose that we have a set of n observations, each with p features. The n observations could correspond to tissue samples for patients with breast cancer, and the p features could correspond to measurements collected for each tissue sample; these could be clinical measurements, such as tumor stage or grade, or they could be gene expression measurements. We may have a reason to believe that there is some heterogeneity among the n tissue samples; for instance, perhaps there are a few different unknown subtypes of breast cancer. Clustering could be used to find these subgroups

Clustering and PCA

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance

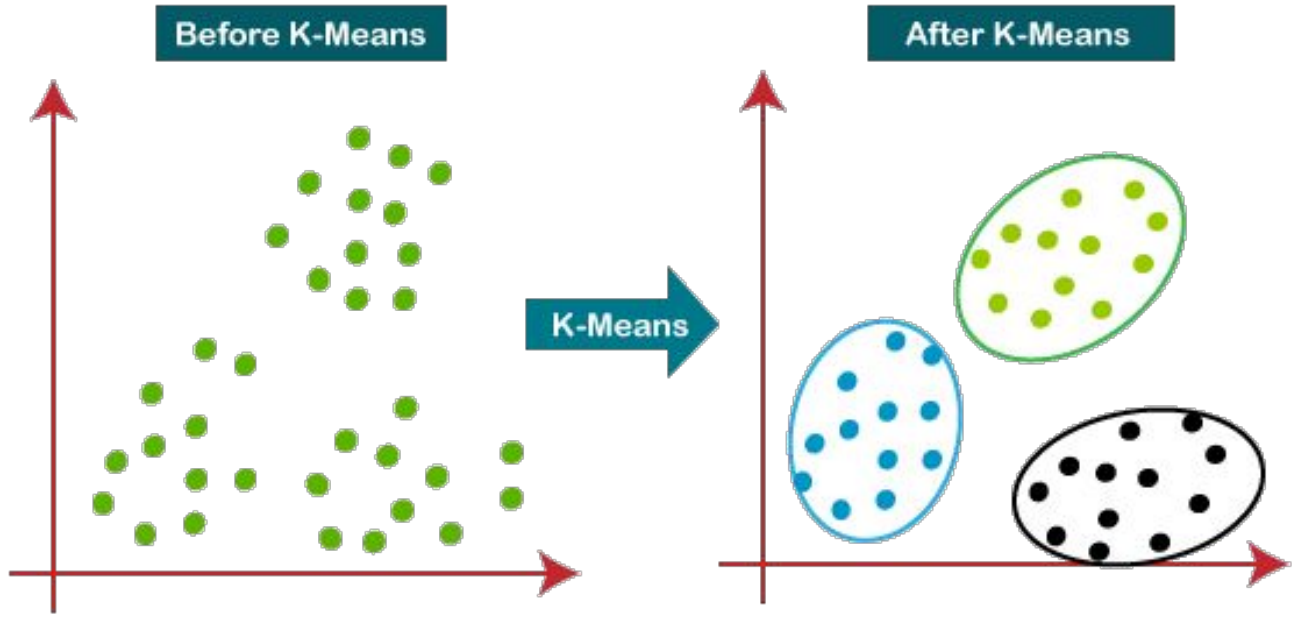
- Clustering looks to find homogeneous subgroups among the observations.

Clustering Methods

- K-means clustering and hierarchical clustering.
- In K-means clustering, we seek to partition the observations into a pre-specified number of clusters.
- On the other hand, in hierarchical clustering, we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a dendrogram, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to n.

K-Means Clustering

- K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, non-overlapping clusters.
- To perform K-means clustering,
  - specify the desired number of clusters K;
  - the K-means algorithm will assign each observation to exactly one of the K clusters



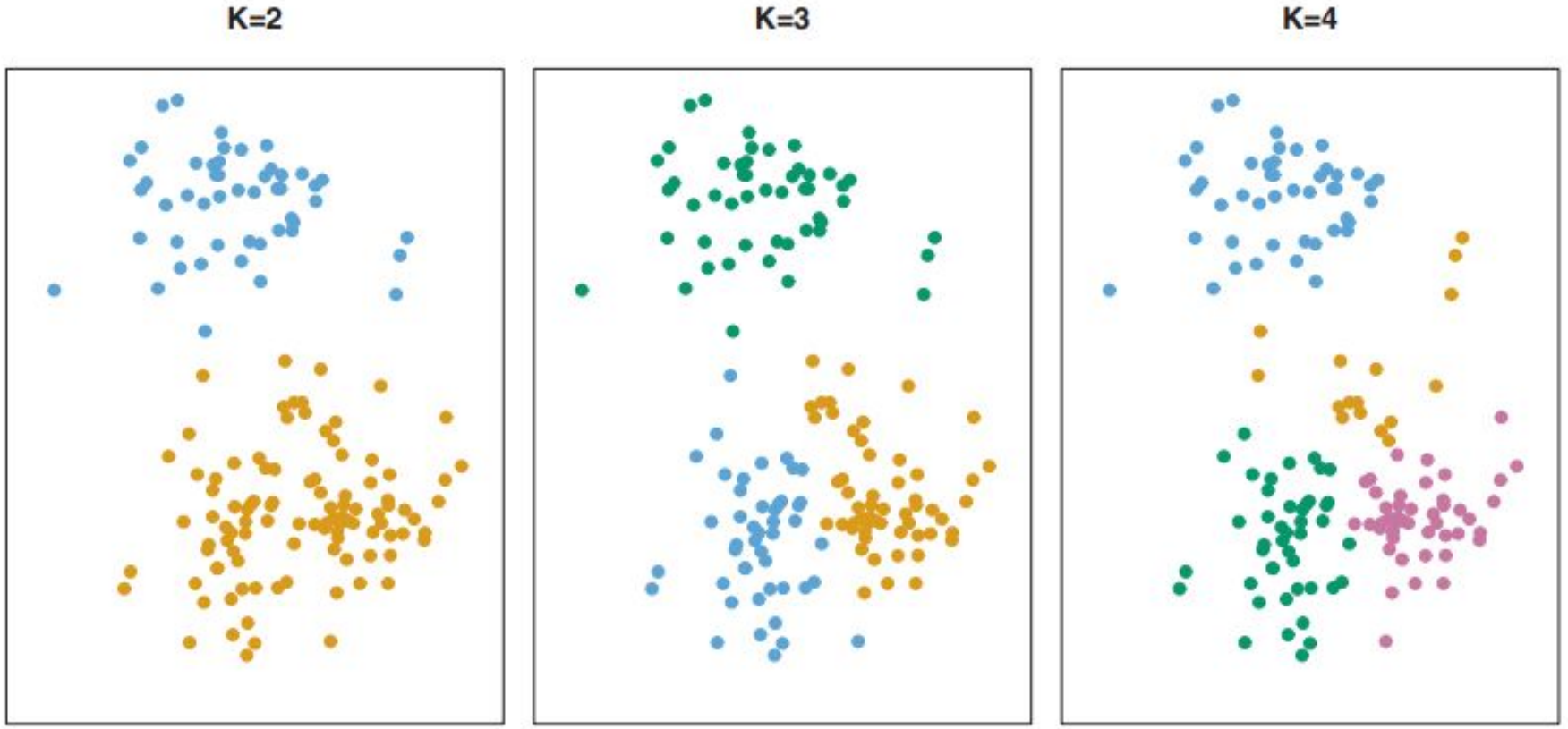


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

K-Means Clustering

- The K-means clustering procedure results from a simple and intuitive mathematical problem
- Let C1,...,CK denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

K-Means Clustering

- The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible.
- The within-cluster variation for cluster Ck is a measure W(Ck) of the amount by which the observations within a cluster differ from each other

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}. \qquad (10.9)$$

Solving (10.9) seems like a reasonable idea, but in order to make it actionable we need to define the within-cluster variation. There are many possible ways to define this concept, but by far the most common choice involves *squared Euclidean distance*. That is, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \tag{10.10}$$

where $|C_k|$ denotes the number of observations in the kth cluster. In other words, the within-cluster variation for the kth cluster is the sum of all of the pairwise squared Euclidean distances between the observations in the kth cluster, divided by the total number of observations in the kth cluster. Combining (10.9) and (10.10) gives the optimization problem that defines K-means clustering,

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}. \tag{10.11}$$

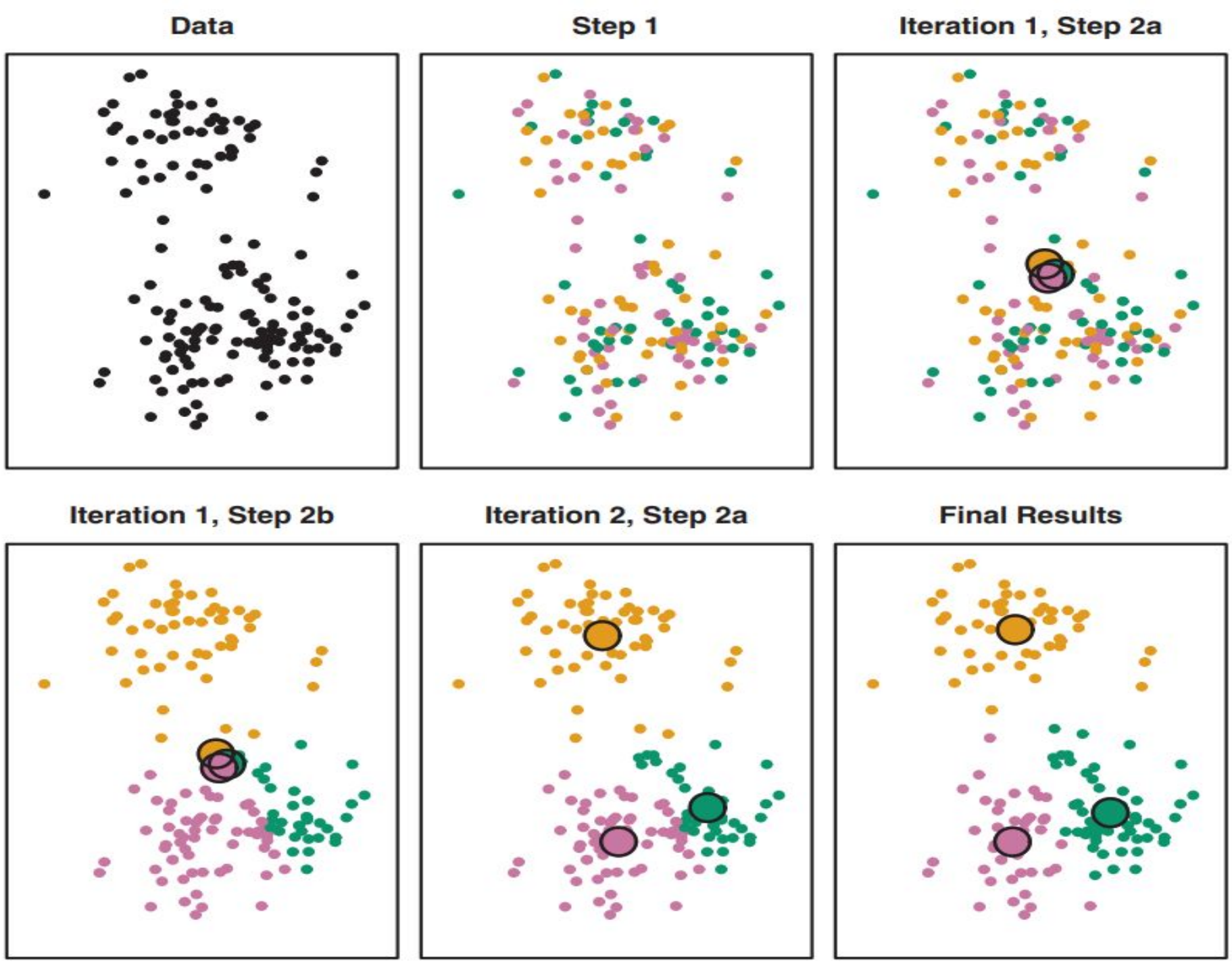
Algorithm 10.1 *K-Means Clustering*

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).

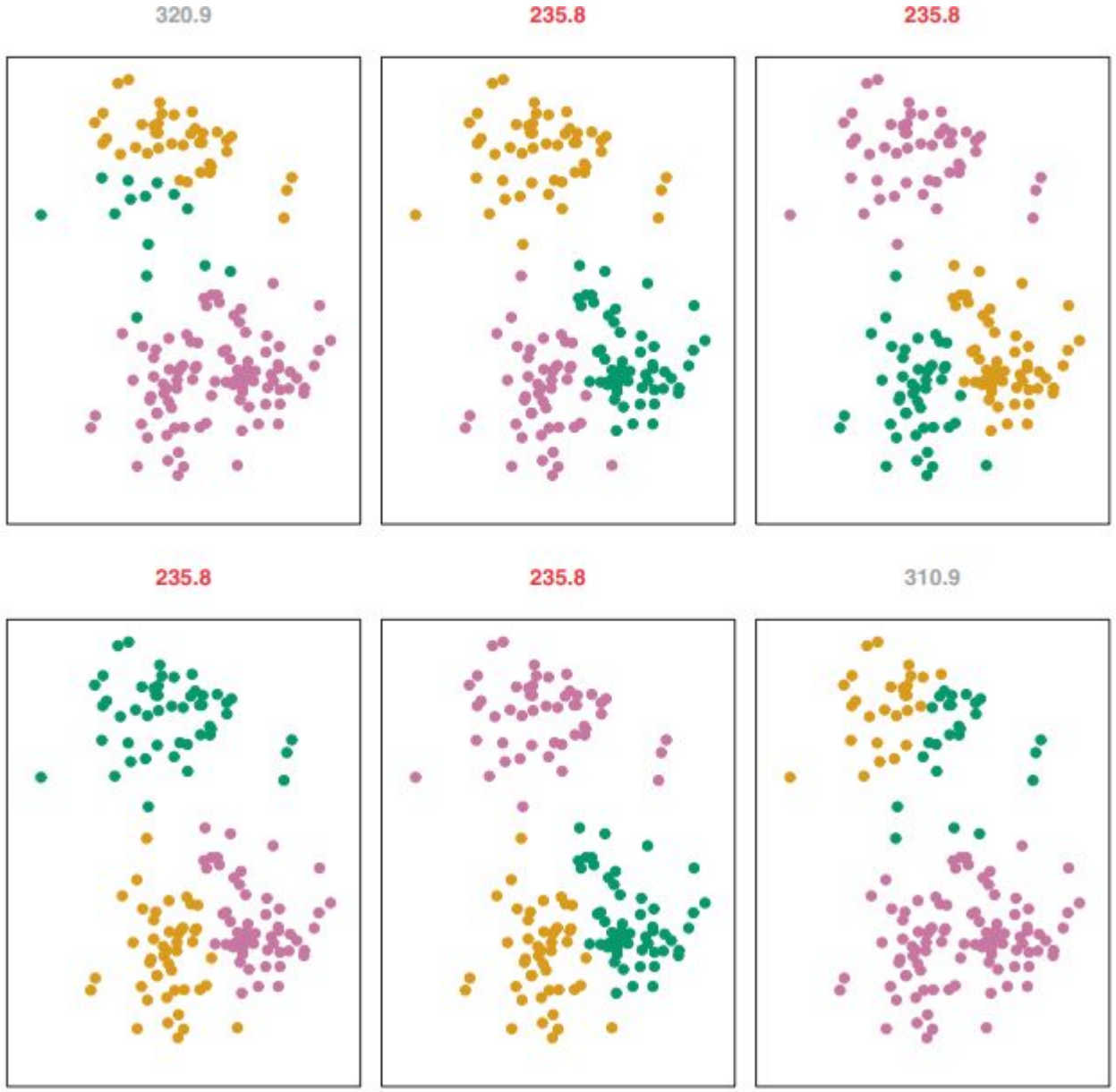

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Algorithm 10.1 is guaranteed to decrease the value of the objective (10.11) at each step. To understand why, the following identity is illuminating:

$$\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2, \qquad (10.12)$$

where $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$ is the mean for feature j in cluster C_k.

Because the K-means algorithm finds a local rather than a global optimum, the results obtain will depend on the initial (random) cluster assignment of each observation in Step 1 of Algorithm 10.1. For this reason, it is important to run the algorithm multiple times from different random initial configurations.



K-means clustering performed six times on the data with K = 3, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.

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
- bottom-up or agglomerative clustering

Hierarchical clustering

Hierarchical clustering is a connectivity-based clustering model that groups the data points together that are close to each other based on the measure of similarity or distance.

The assumption is that data points that are close to each other are more similar or related than data points that are farther apart.

Dendrogram

- A dendrogram, a tree-like figure produced by hierarchical clustering, depicts the hierarchical relationships between groups.
- Individual data points are located at the bottom of the dendrogram, while the largest clusters, which include all the data points, are located at the top.
- In order to generate different numbers of clusters, the dendrogram can be sliced at various heights.
- The dendrogram is created by iteratively merging or splitting clusters based on a measure of similarity or distance between data points.

Dendrogram

- Clusters are divided or merged repeatedly until all data points are contained within a single cluster, or until the predetermined number of clusters is attained.
- We can look at the dendrogram and measure the height at which the branches of the dendrogram form distinct clusters to calculate the ideal number of clusters.
- The dendrogram can be sliced at this height to determine the number of clusters.

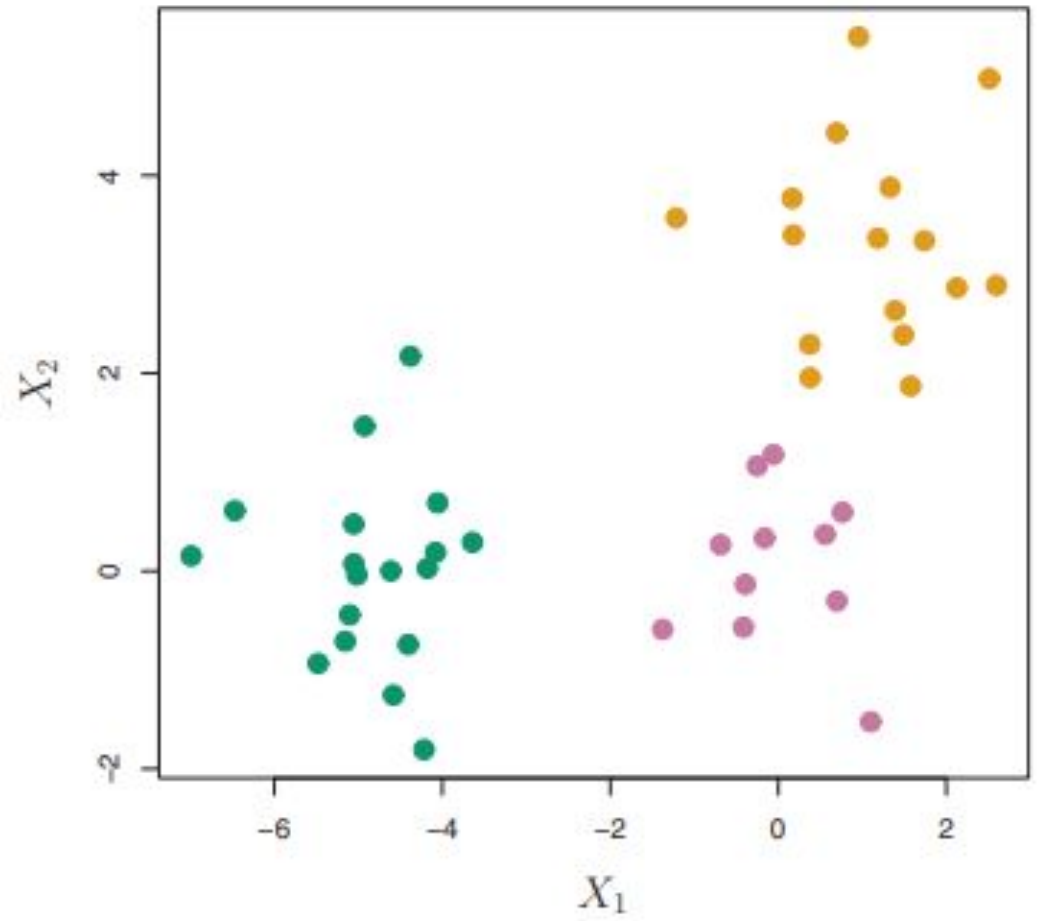
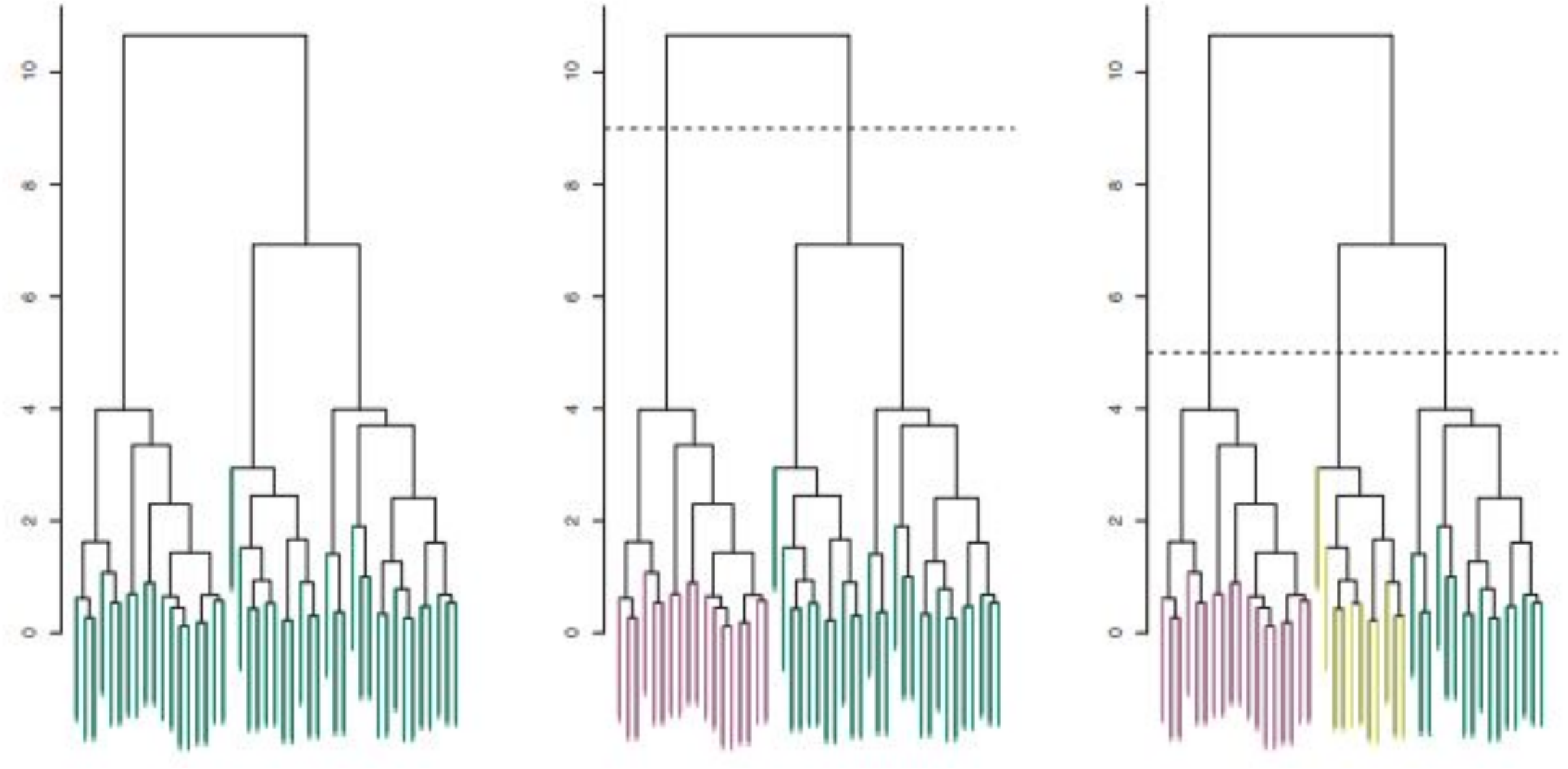


FIGURE 10.8. *Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.*


0
2
4
6
8
10

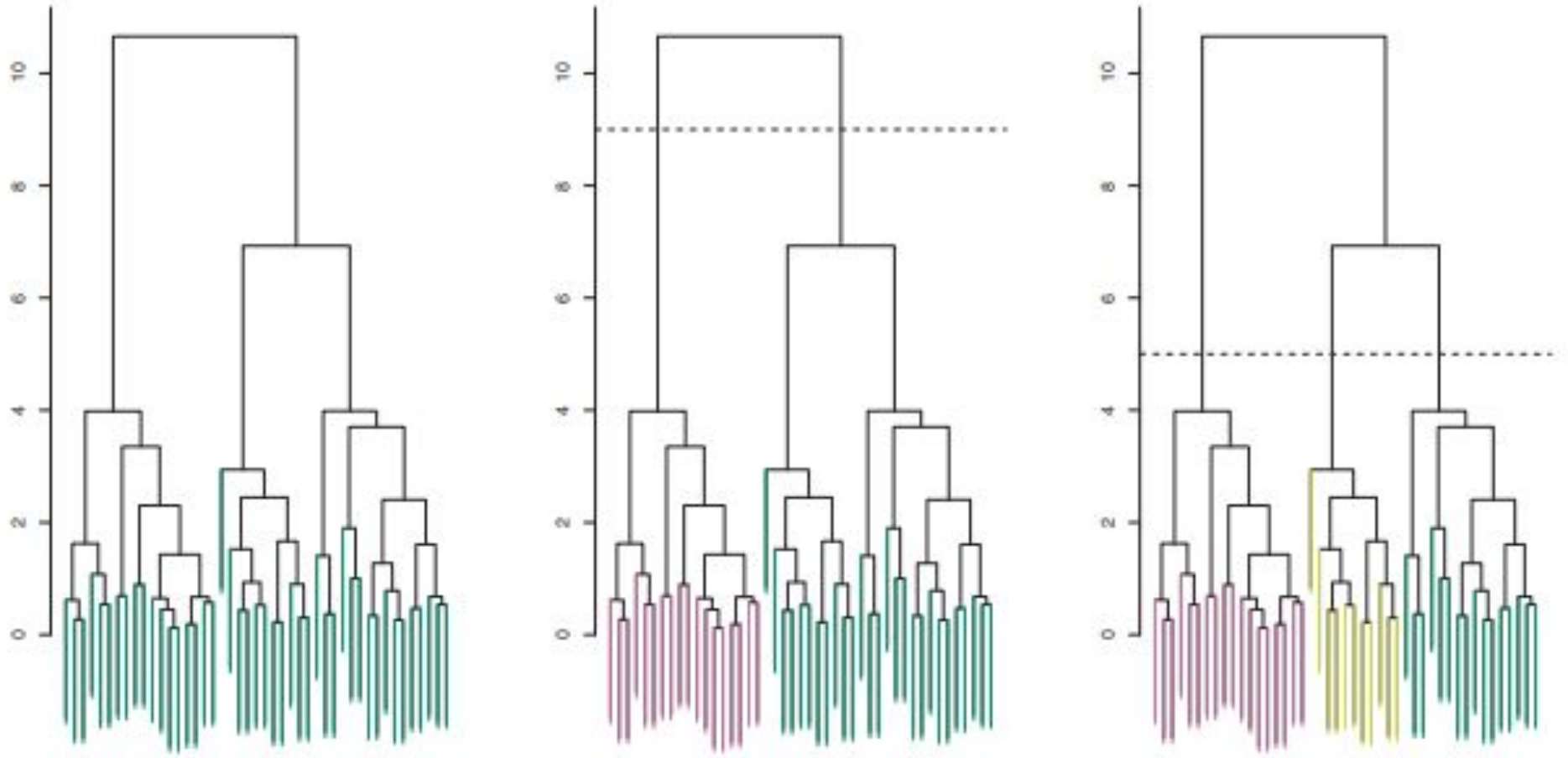


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

Methods of clustering

1. Agglomerative clustering
   - Bottom-up approach
2. Divisive clustering
   - Top-down approach

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*.

Distance between two data points

1. Euclidean distance
2. Squared Euclidean distance
3. Manhattan distance
4. Maximum distance
5. Levenshtein distance

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.

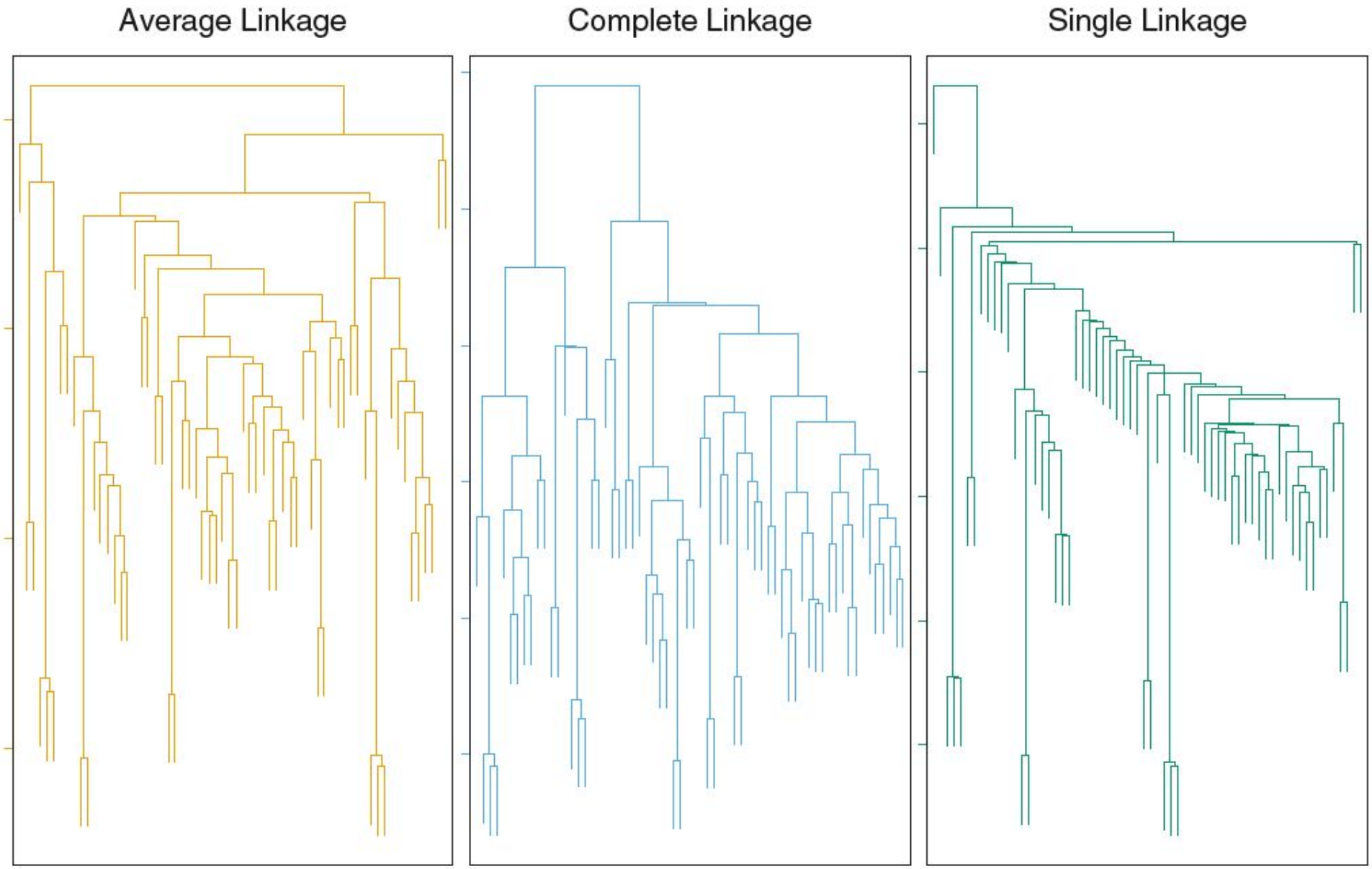


FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

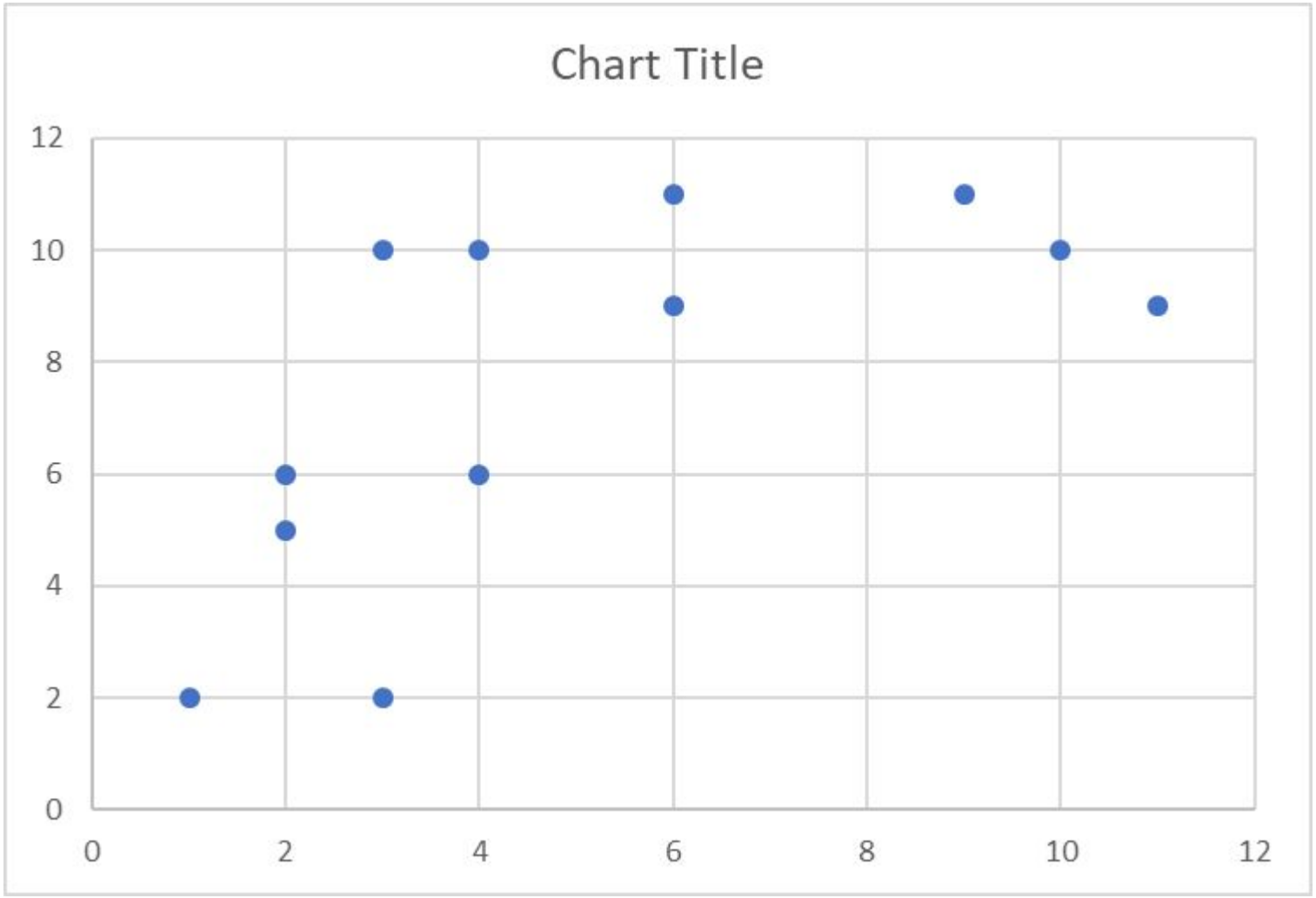
Practical Issues in Clustering

1. Small Decisions with Big Consequences
   - Should the observations or features first be standardized in some way?
   - In the case of hierarchical clustering,
     - What dissimilarity measure should be used?
     - What type of linkage should be used?
     - Where should we cut the dendrogram to obtain clusters?
   - In the case of K-means clustering, how many clusters should we look for in the data?

Practical Issues in Clustering

-
2. Validating Clusters Obtained
   - Noise in the data set
3. Other Considerations in Clustering
   - Is the number of clusters enough?
     - suppose that most of the observations truly belong to a small number of (unknown) subgroups, and a small subset of the observations are quite different from each other and from all other observations.
     - How to accommodate the presence of Outliers
4. A Tempered Approach to Interpreting the Results of Clustering

Random Forest



	x	y
a1	1	2
a2	2	6
a3	9	11
a4	3	10
a5	6	9
a6	10	10
a7	4	6
a8	3	2
a9	4	10
a10	2	5
a11	6	11
a12	11	9

	x	y	Cluster
a1	1	2	C1
a2	2	6	C1
a3	9	11	C1
a4	3	10	C1
a5	6	9	C2
a6	10	10	C2
a7	4	6	C2
a8	3	2	C2
a9	4	10	C3
a10	2	5	C3
a11	6	11	C3
a12	11	9	C3