Lecture 4: Finish PCA, Clustering

Reading: Sections 10.3, 10.5

STATS 202: Data mining and analysis

Lester Mackey
September 28, 2015

(Slide credits: Sergio Bacallado)

Announcements

- If you haven't yet signed off on the Kaggle competition terms and conditions, please review the terms as soon as possible

Recap: Principal components analysis

- PCA provides a way to compress and visualize high dimensional data, summarizing the most "important" information
- First PC is the direction along which our **data varies the most**

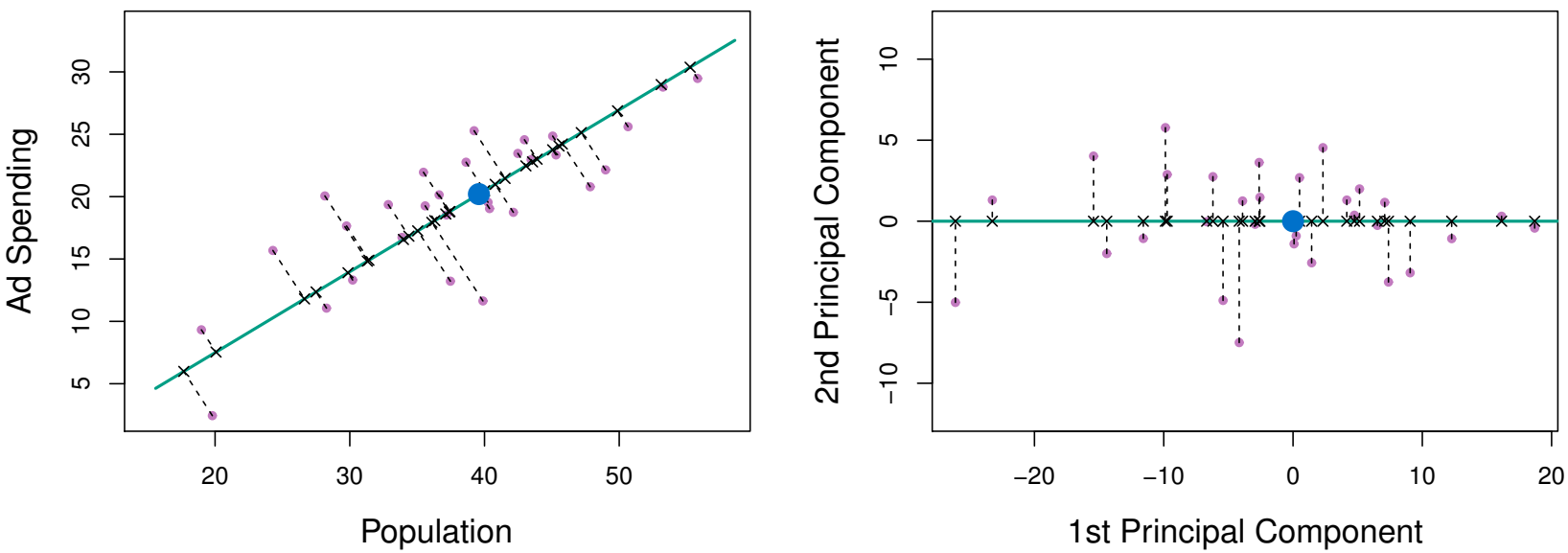


Figure 6.15

Intuition: High variance directions are often interesting directions

Finding PCs with optimization

Let $\mathbf{X}$ be a data matrix with n samples and p variables.

- We assume that each column is **centered**, i.e., that the mean of the column has been subtracted away from each entry

Finding PCs with optimization

Let $\mathbf{X}$ be a data matrix with n samples and p variables.

- We assume that each column is **centered**, i.e., that the mean of the column has been subtracted away from each entry

To find the first principal component $\phi_1 = (\phi_{11}, \ldots, \phi_{p1})$, we solve the following optimization problem

Finding PCs with optimization

Let $\mathbf{X}$ be a data matrix with n samples and p variables.

- We assume that each column is **centered**, i.e., that the mean of the column has been subtracted away from each entry

To find the first principal component $\phi_1 = (\phi_{11}, \ldots, \phi_{p1})$, we solve the following optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\}$$

$$\text{subject to} \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Finding PCs with optimization

Let $\mathbf{X}$ be a data matrix with n samples and p variables.

- We assume that each column is **centered**, i.e., that the mean of the column has been subtracted away from each entry

To find the first principal component $\phi_1 = (\phi_{11}, \ldots, \phi_{p1})$, we solve the following optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\}$$

$$\text{subject to} \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Scalar projection of x_i onto ϕ_1. Also known as **the score z_{i1}**.

Sample mean of the scores is 0:

$\frac{1}{n} \sum_{i=1}^{n} z_{i1} = \frac{1}{n} \sum_{i=1}^{n} (\sum_{j=1}^{p} \phi_{j1} x_{ij}) = \sum_{j=1}^{p} \phi_{j1} \frac{1}{n} \sum_{i=1}^{n} x_{ij} = 0$

Finding PCs with optimization

Let $\mathbf{X}$ be a data matrix with n samples and p variables.

- We assume that each column is **centered**, i.e., that the mean of the column has been subtracted away from each entry

To find the first principal component $\phi_1 = (\phi_{11}, \ldots, \phi_{p1})$, we solve the following optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\}$$

$$\text{subject to} \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Sample variance of the scores z_{i1}
(i.e., sample variance of projections of datapoints onto ϕ_1).

Finding PCs with optimization

To find the second principal component $\phi_2 = (\phi_{12}, \ldots, \phi_{p2})$, we solve the following optimization problem

Finding PCs with optimization

To find the second principal component $\phi_2 = (\phi_{12}, \ldots, \phi_{p2})$, we solve the following optimization problem

$$\max_{\phi_{12},\ldots,\phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j2} x_{ij} \right)^2 \right\}$$

$$\text{subject to } \sum_{j=1}^{p} \phi_{j2}^2 = 1 \quad \text{and} \quad \sum_{j=1}^{p} \phi_{j1}\phi_{j2} = 0.$$

Finding PCs with optimization

To find the second principal component $\phi_2 = (\phi_{12}, \ldots, \phi_{p2})$, we solve the following optimization problem

$$\max_{\phi_{12},\ldots,\phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j2} x_{ij} \right)^2 \right\}$$

$$\text{subject to } \sum_{j=1}^{p} \phi_{j2}^2 = 1 \quad \text{and} \quad \sum_{j=1}^{p} \phi_{j1}\phi_{j2} = 0.$$

First and second principal components must be orthogonal.

Finding PCs with optimization

To find the second principal component $\phi_2 = (\phi_{12}, \ldots, \phi_{p2})$, we solve the following optimization problem

$$\max_{\phi_{12},\ldots,\phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j2} x_{ij} \right)^2 \right\}$$

$$\text{subject to } \sum_{j=1}^{p} \phi_{j2}^2 = 1 \quad \text{and} \quad \sum_{j=1}^{p} \phi_{j1}\phi_{j2} = 0.$$

First and second principal components must be orthogonal.

Equivalent to saying that the scores $(z_{11}, \ldots, z_{n1})$ and $(z_{12}, \ldots, z_{n2})$ are uncorrelated.

Solving the optimization problems

These optimization problems are standard in linear algebra. They are solved by:

Solving the optimization problems

These optimization problems are standard in linear algebra. They are solved by:

- The singular value decomposition (SVD) of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$$

where the ith column of $\mathbf{\Phi}$ is the ith principal component ϕ_i, and the ith column of $\mathbf{U}\mathbf{\Sigma}$ is the ith vector of scores $(z_{1i}, \ldots, z_{ni})$.

Solving the optimization problems

These optimization problems are standard in linear algebra. They are solved by:

- The singular value decomposition (SVD) of $\mathbf{X}$:

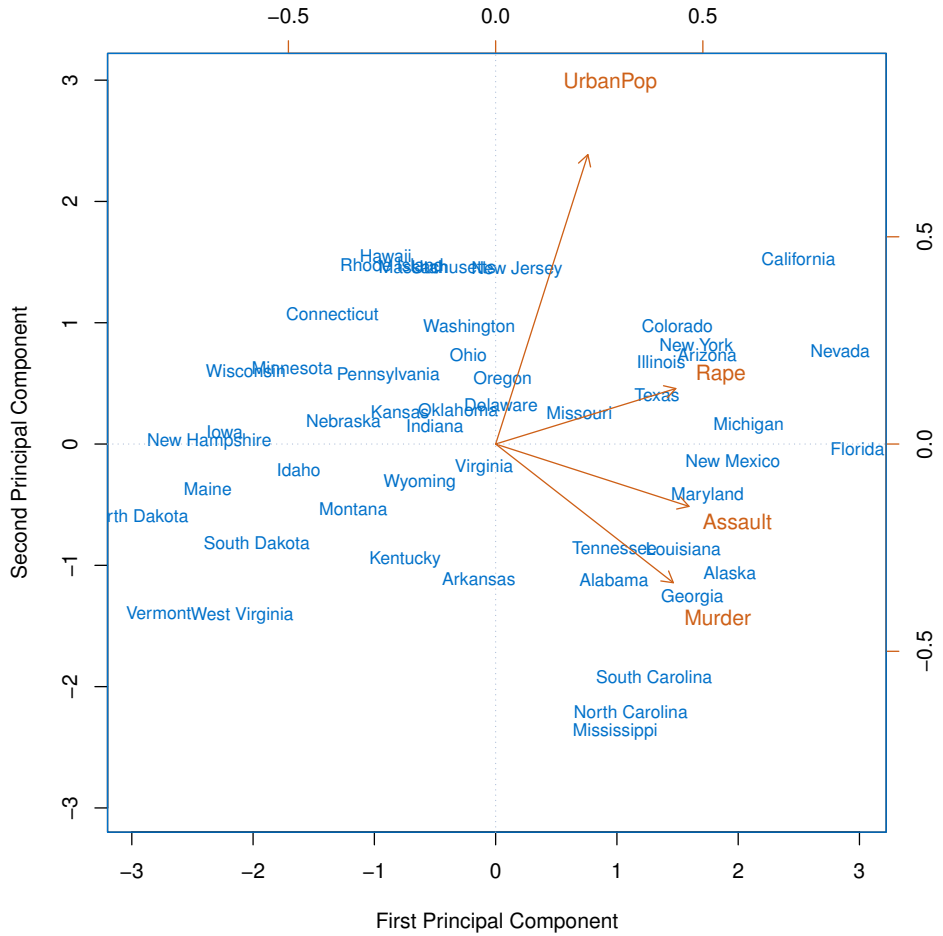
$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$$

  where the ith column of $\mathbf{\Phi}$ is the ith principal component ϕ_i, and the ith column of $\mathbf{U}\mathbf{\Sigma}$ is the ith vector of scores $(z_{1i}, \ldots, z_{ni})$.

- The eigendecomposition of $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{\Phi}\mathbf{\Sigma}^2\mathbf{\Phi}^T$$

PCA in practice: The biplot



- j-th variable represented by its PC loadings (ϕ_{1j}, ϕ_{2j})
- i-th datapoint represented by its scores (z_{i1}, z_{i2})

Scaling the variables

We have already discussed **centering** our variables to have mean 0.

Scaling the variables

We have already discussed **centering** our variables to have mean 0.

In some instances we will also want to *rescale* our variables prior to performing PCA

- e.g., if different variables are measured in different units
- Or if we're principally interested in the correlations between variables with very different scales

Scaling the variables

We have already discussed **centering** our variables to have mean 0.

In some instances we will also want to *rescale* our variables prior to performing PCA

- e.g., if different variables are measured in different units
- Or if we're principally interested in the correlations between variables with very different scales

When a variable is centered and rescaled to have standard deviation 1, we say it has been **standardized**.

Scaling the variables

We have already discussed **centering** our variables to have mean 0.

In some instances we will also want to *rescale* our variables prior to performing PCA

- e.g., if different variables are measured in different units
- Or if we're principally interested in the correlations between variables with very different scales

When a variable is centered and rescaled to have standard deviation 1, we say it has been **standardized**.

In other instances, initial variable scalings are meaningful and should be preserved

- e.g., Gene expression levels for different genes all measured in the same units

Example: scaled vs. unscaled PCA

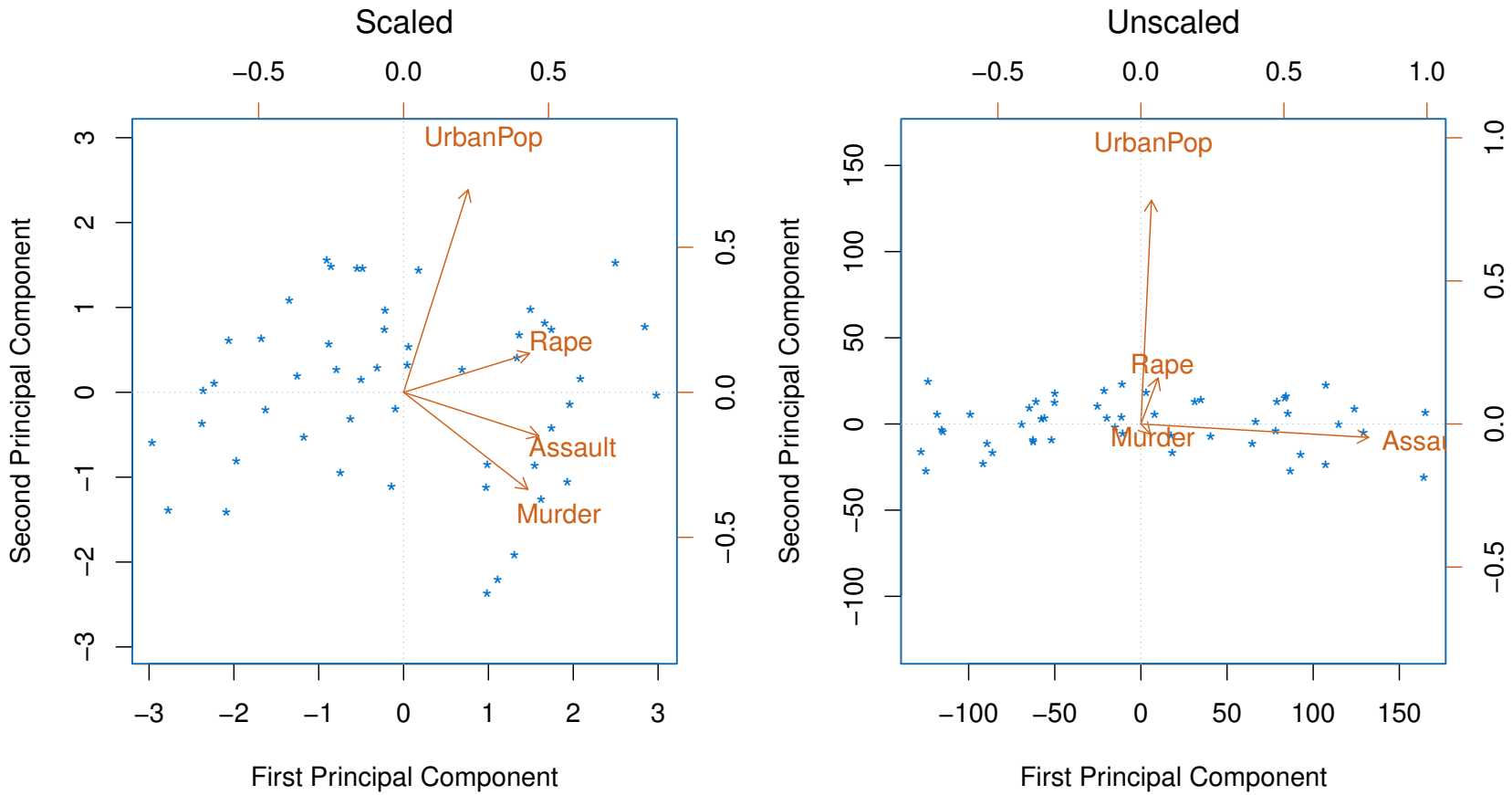
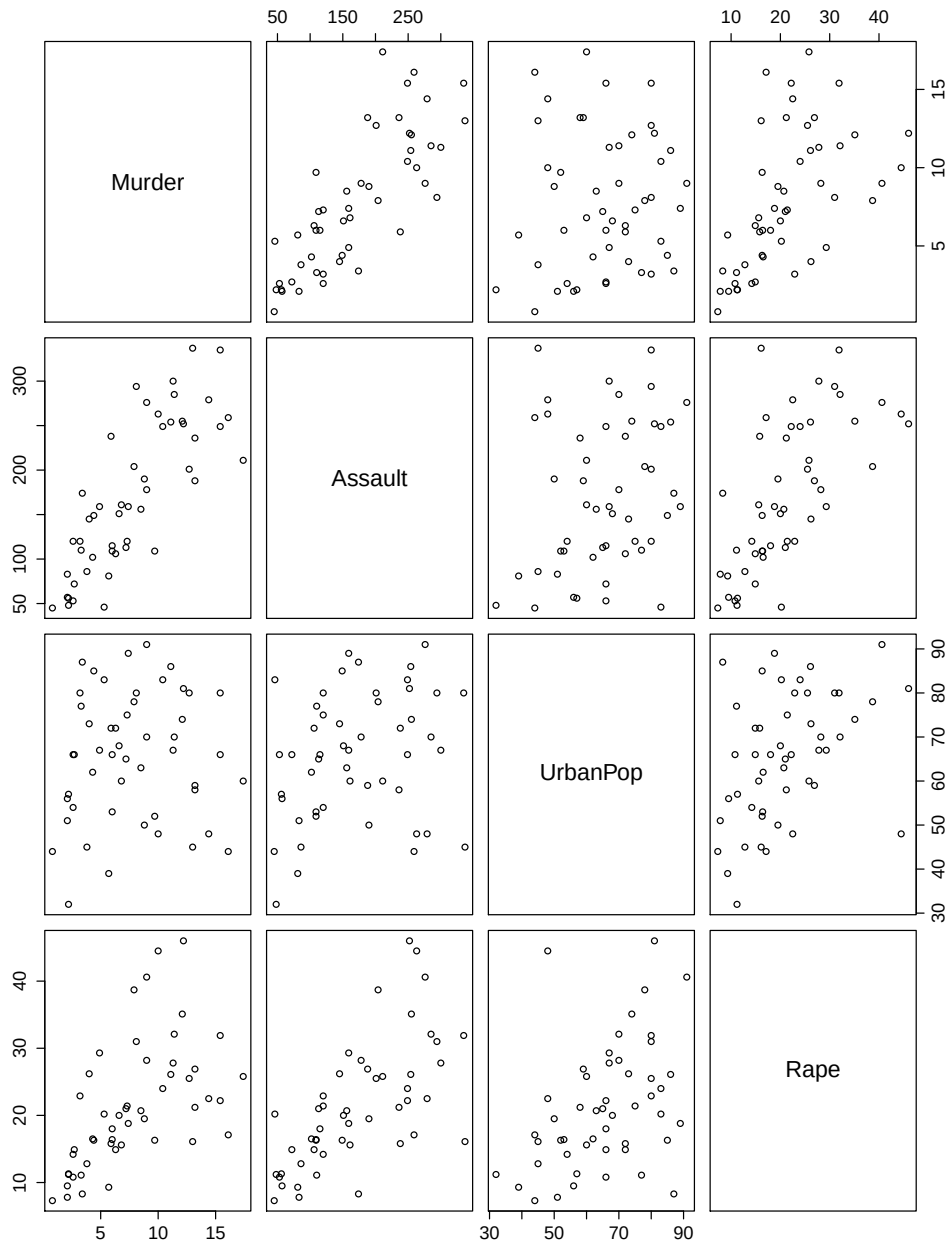
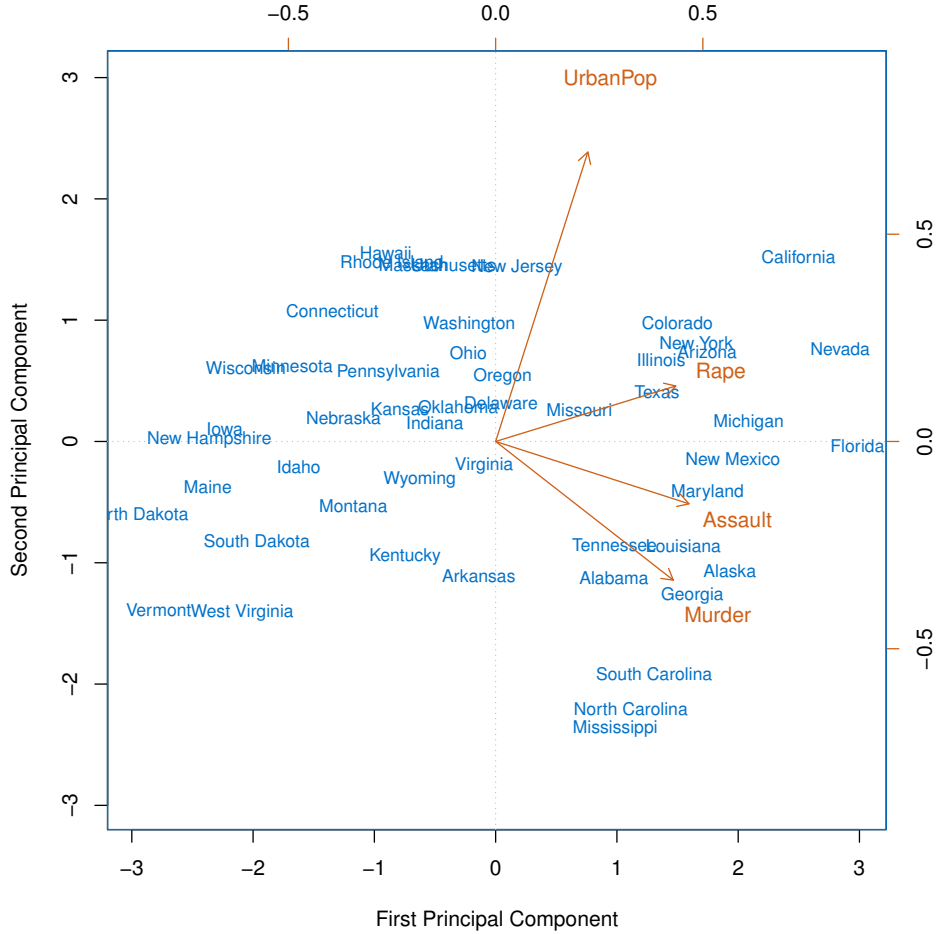


Figure 10.3

How many principal components are enough?



How many principal components are enough?



How can we tell if 2 principal components capture most of the relevant information?

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

Recall that datapoint x_i has associated score $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$, induced by the mth principal component.

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

Recall that datapoint x_i has associated score $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$, induced by the mth principal component.

The mth **score vector** $(z_{1m}, \ldots, z_{nm})$ can be interpreted as a *new* variable. The (sample) variance of this variable decreases as we take m from 1 to p.

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

Recall that datapoint x_i has associated score $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$, induced by the mth principal component.

The mth **score vector** $(z_{1m}, \ldots, z_{nm})$ can be interpreted as a *new* variable. The (sample) variance of this variable decreases as we take m from 1 to p. However, the total variance of the score vectors is the same as the total variance of the original variables:

$$\sum_{m=1}^{p} \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \sum_{m=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{im}^2.$$

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

Recall that datapoint x_i has associated score $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$, induced by the mth principal component.

The mth **score vector** $(z_{1m}, \ldots, z_{nm})$ can be interpreted as a *new* variable. The (sample) variance of this variable decreases as we take m from 1 to p. However, the total variance of the score vectors is the same as the total variance of the original variables:

$$\sum_{m=1}^{p} \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \sum_{m=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{im}^2.$$

We can quantify how much of the variance is captured by the first m principal components/score variables.

The proportion of variance explained

The variance of the mth score variable is (for $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$)

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2$$

The proportion of variance explained

The variance of the mth score variable is (for $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$)

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2$$

The proportion of variance explained

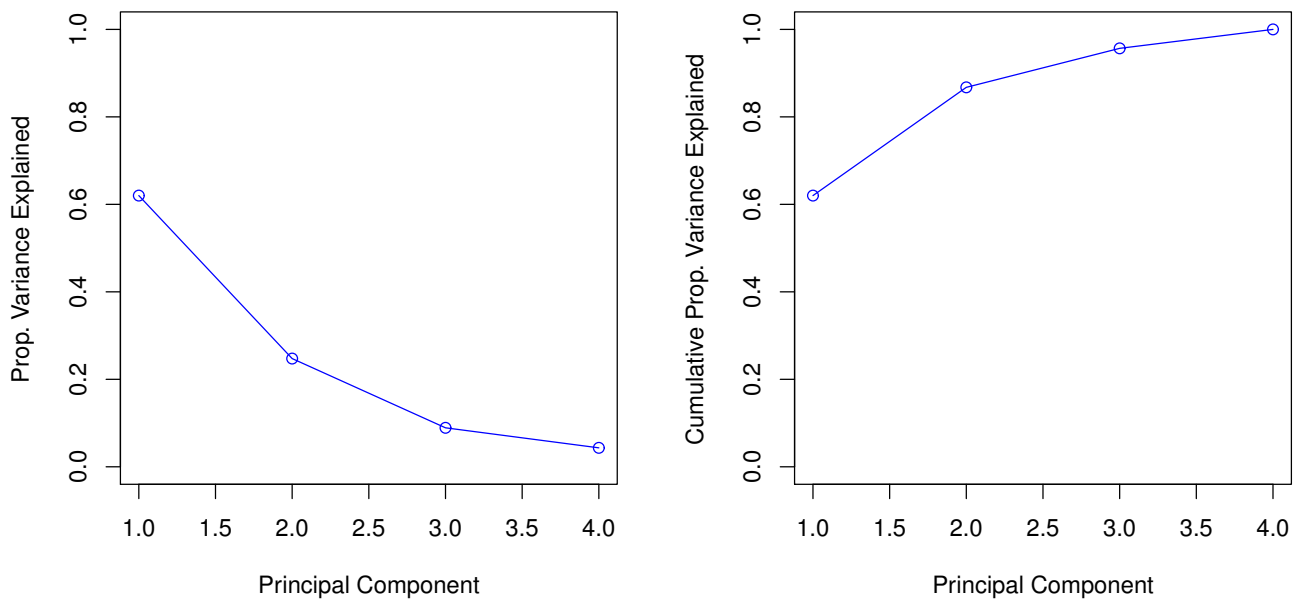
The variance of the mth score variable is (for $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$)

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2 = \frac{1}{n}\mathbf{\Sigma}_{mm}^2.$$

The proportion of variance explained

The variance of the mth score variable is (for $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Phi}^T$)

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2 = \frac{1}{n}\mathbf{\Sigma}_{mm}^2.$$



Scree plot

Clustering

Goal: Find groups of datapoints that are similar to one another

Clustering

Goal: Find groups of datapoints that are similar to one another

- As in classification, we **assign** a class or **cluster label to each datapoint** in the data matrix.

Clustering

Goal: Find groups of datapoints that are similar to one another

- As in classification, we **assign** a class or **cluster label to each datapoint** in the data matrix.
- Unlike in classification, we do not have access to labeled training points: **we only observe the input variables**
- Clustering is an **unsupervised** learning task

Clustering

Goal: Find groups of datapoints that are similar to one another

- As in classification, we **assign** a class or **cluster label to each datapoint** in the data matrix.
- Unlike in classification, we do not have access to labeled training points: **we only observe the input variables**
- Clustering is an **unsupervised** learning task

We will discuss two algorithms:

Clustering

Goal: Find groups of datapoints that are similar to one another

- As in classification, we **assign** a class or **cluster label to each datapoint** in the data matrix.
- Unlike in classification, we do not have access to labeled training points: **we only observe the input variables**
- Clustering is an **unsupervised** learning task

We will discuss two algorithms:

- K-means clustering

Clustering

Goal: Find groups of datapoints that are similar to one another

- As in classification, we **assign** a class or **cluster label to each datapoint** in the data matrix.
- Unlike in classification, we do not have access to labeled training points: **we only observe the input variables**
- Clustering is an **unsupervised** learning task

We will discuss two algorithms:

- K-means clustering
- Hierarchical clustering

K-means clustering

- K is the number of clusters and must be fixed in advance.

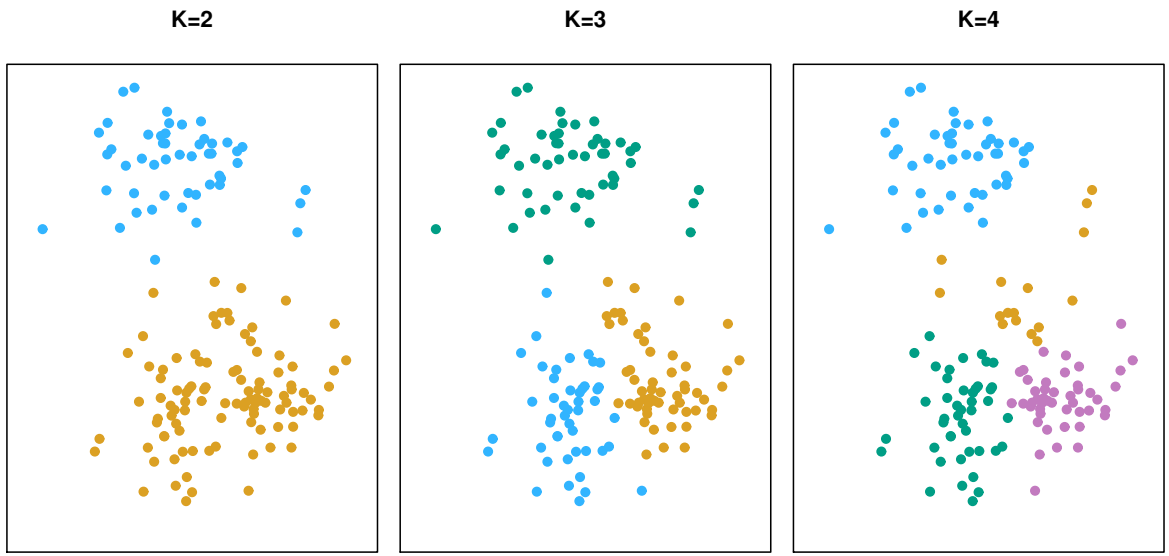


Figure 10.5

K-means clustering

- K is the number of clusters and must be fixed in advance.
- The goal of this method is to minimize the dissimilarity of samples $W(C_\ell)$ within each cluster C_ℓ:

$$\min_{C_1,\ldots,C_K} \sum_{\ell=1}^{K} W(C_\ell)$$

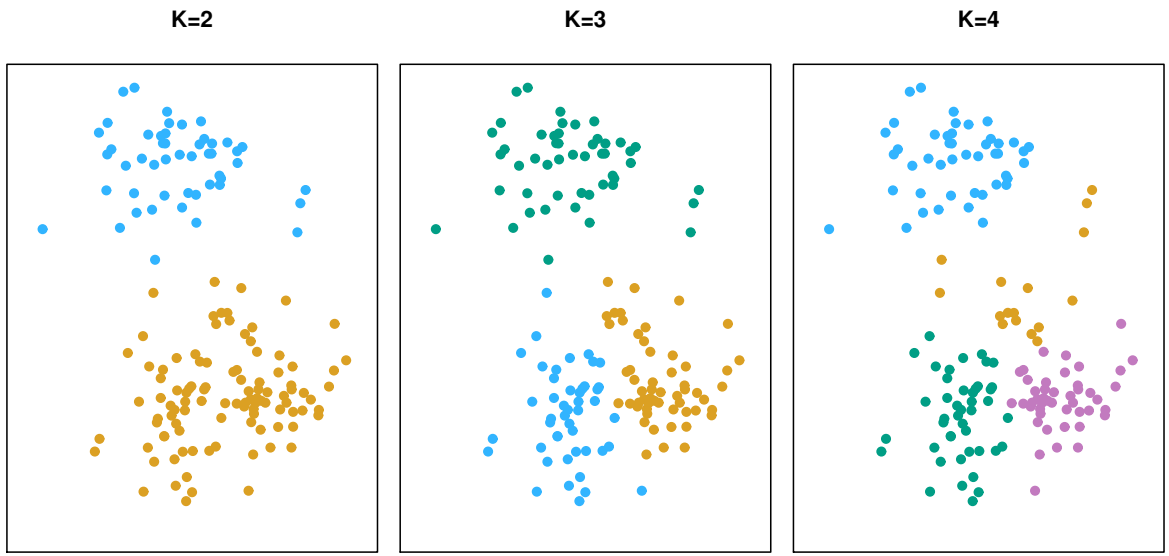


Figure 10.5

K-means clustering

- K is the number of clusters and must be fixed in advance.
- The goal of this method is to minimize the dissimilarity of samples $W(C_\ell)$ within each cluster C_ℓ:

$$\min_{C_1,\ldots,C_K} \sum_{\ell=1}^{K} W(C_\ell) \quad ; \quad W(C_\ell) = \frac{1}{|C_\ell|} \sum_{ij \in C_\ell} \text{Distance}^2(x_i, x_j).$$

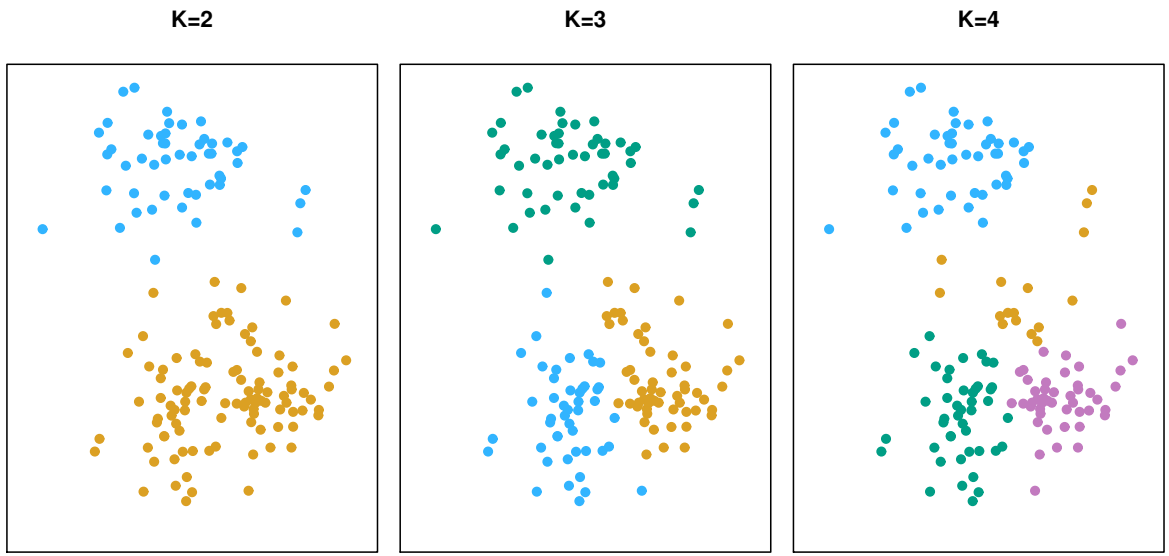


Figure 10.5

Lloyd's algorithm for K-means clustering

1. Assign each sample a cluster label from 1 to K arbitrarily, e.g., uniformly at random.

Lloyd's algorithm for K-means clustering

1. Assign each sample a cluster label from 1 to K arbitrarily, e.g., uniformly at random.
2. Iterate these two steps until the clustering is constant:

Lloyd's algorithm for K-means clustering

1. Assign each sample a cluster label from 1 to K arbitrarily, e.g., uniformly at random.

2. Iterate these two steps until the clustering is constant:

   - Find the *centroid* of each cluster ℓ; i.e., the average $\overline{x}_\ell$ of all the samples in the cluster:

$$\overline{x}_{\ell j} = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} x_{ij} \quad \text{for } j = 1, \ldots, p.$$

Lloyd's algorithm for K-means clustering

1. Assign each sample a cluster label from 1 to K arbitrarily, e.g., uniformly at random.

2. Iterate these two steps until the clustering is constant:

   - Find the *centroid* of each cluster ℓ; i.e., the average $\overline{x}_\ell$ of all the samples in the cluster:

$$\overline{x}_{\ell j} = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} x_{ij} \quad \text{for } j = 1, \ldots, p.$$

   - Reassign each sample to the nearest centroid.

Lloyd's algorithm for K-means clustering

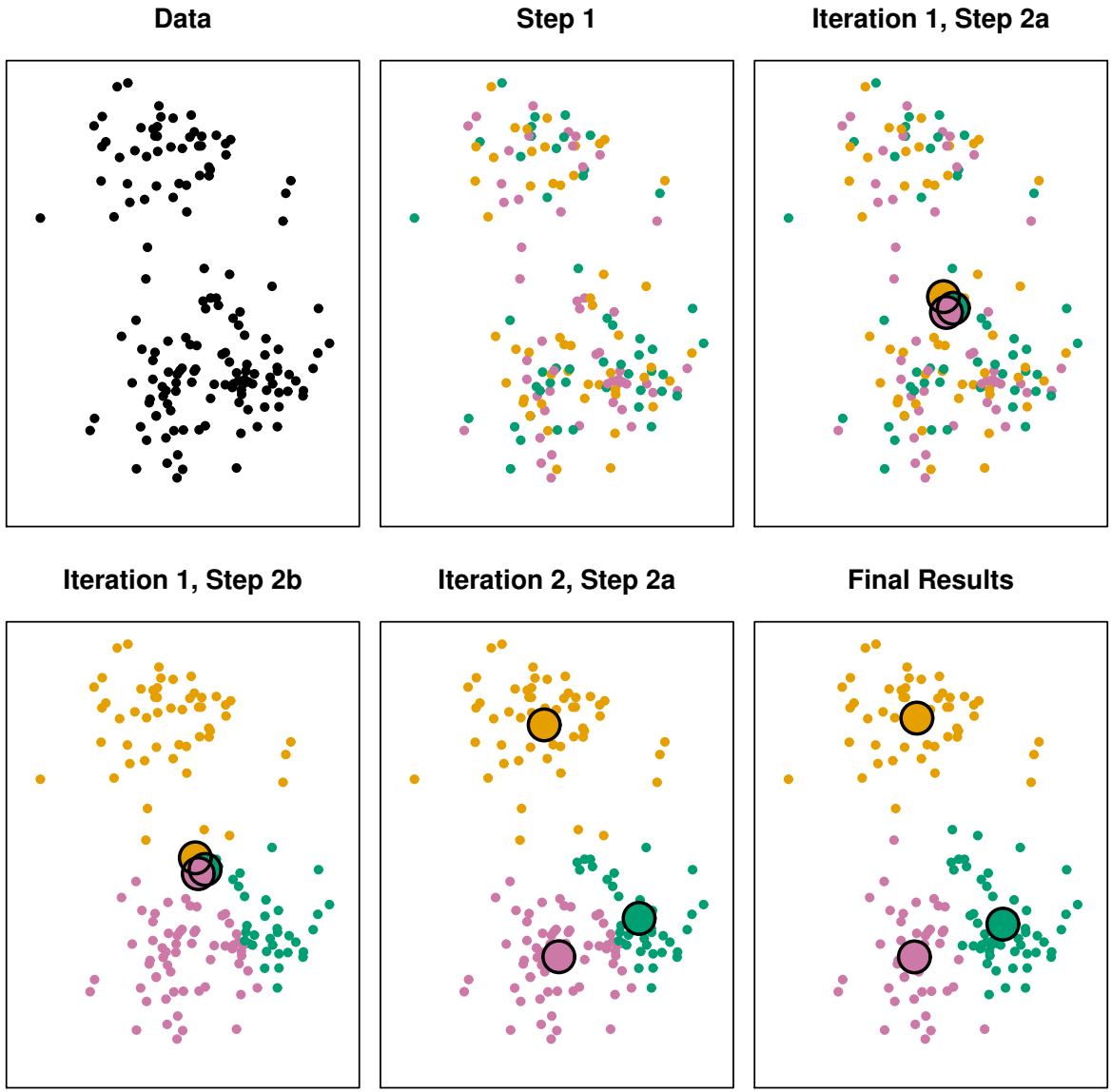


Figure 10.6

Properties of K-means clustering

- Lloyd's algorithm always terminates (if we break ties consistently)

Properties of K-means clustering

- Lloyd's algorithm always terminates (if we break ties consistently)

  Why?

$$\min_{C_1,\ldots,C_K} \sum_{\ell=1}^{K} W(C_\ell)$$

$$W(C_\ell) = \frac{1}{|C_\ell|} \sum_{ij \in C_\ell} \text{Distance}^2(x_i, x_j) = 2 \sum_{i \in C_\ell} \text{Distance}^2(x_i, \overline{x}_\ell)$$

Properties of K-means clustering

- Lloyd's algorithm always terminates (if we break ties consistently)

  Why?

$$\min_{C_1,\ldots,C_K} \sum_{\ell=1}^{K} W(C_\ell)$$

$$W(C_\ell) = \frac{1}{|C_\ell|} \sum_{ij \in C_\ell} \text{Distance}^2(x_i, x_j) = 2 \sum_{i \in C_\ell} \text{Distance}^2(x_i, \overline{x}_\ell)$$

  After each step of Lloyd's algorithm, K-means objective decreases or stays the same.

Properties of K-means clustering

- Lloyd's algorithm always terminates (if we break ties consistently)

  Why?

$$\min_{C_1,\ldots,C_K} \sum_{\ell=1}^{K} W(C_\ell)$$

$$W(C_\ell) = \frac{1}{|C_\ell|} \sum_{ij \in C_\ell} \text{Distance}^2(x_i, x_j) = 2 \sum_{i \in C_\ell} \text{Distance}^2(x_i, \overline{x}_\ell)$$

  After each step of Lloyd's algorithm, K-means objective decreases or stays the same.

- Different initializations can yield different clusterings (solution returned not guaranteed to be globally optimal)

Example: K-means with random restarts



Figure 10.7

Example: K-means with random restarts

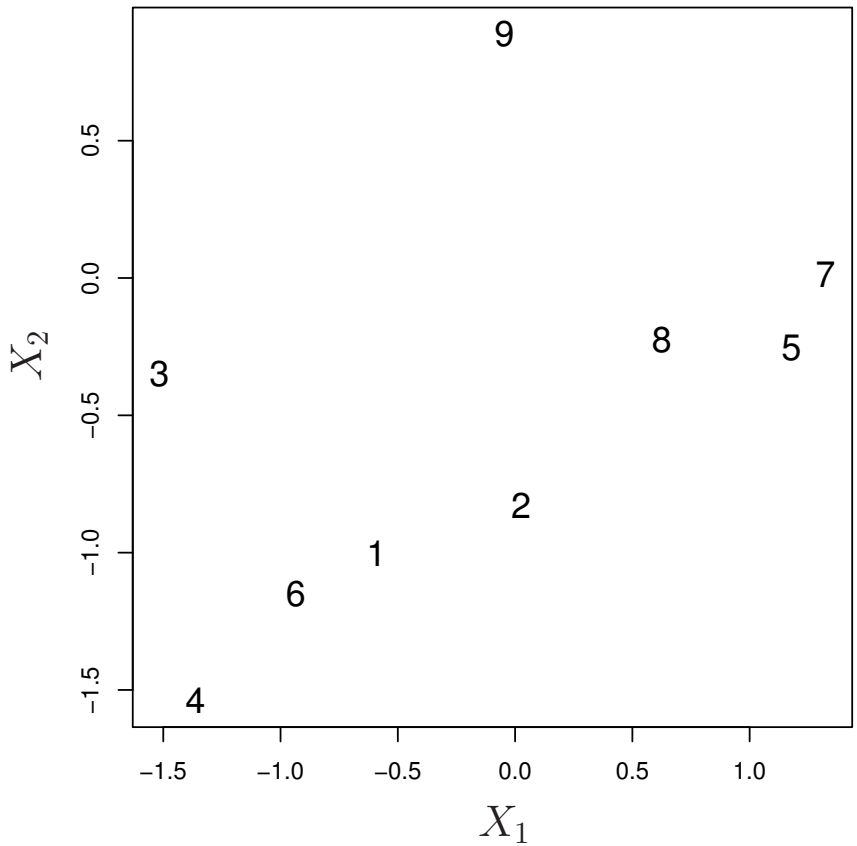


Figure 10.7

In practice, we start from many random initializations (called **random restarts**) and choose the output which minimizes the objective function.

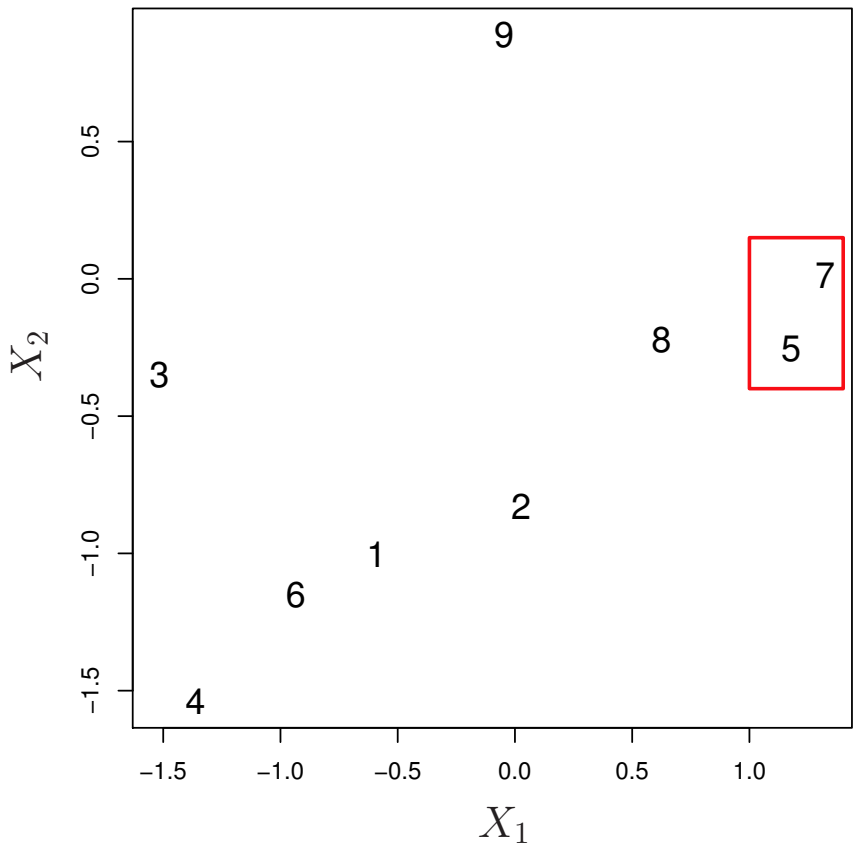
Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.



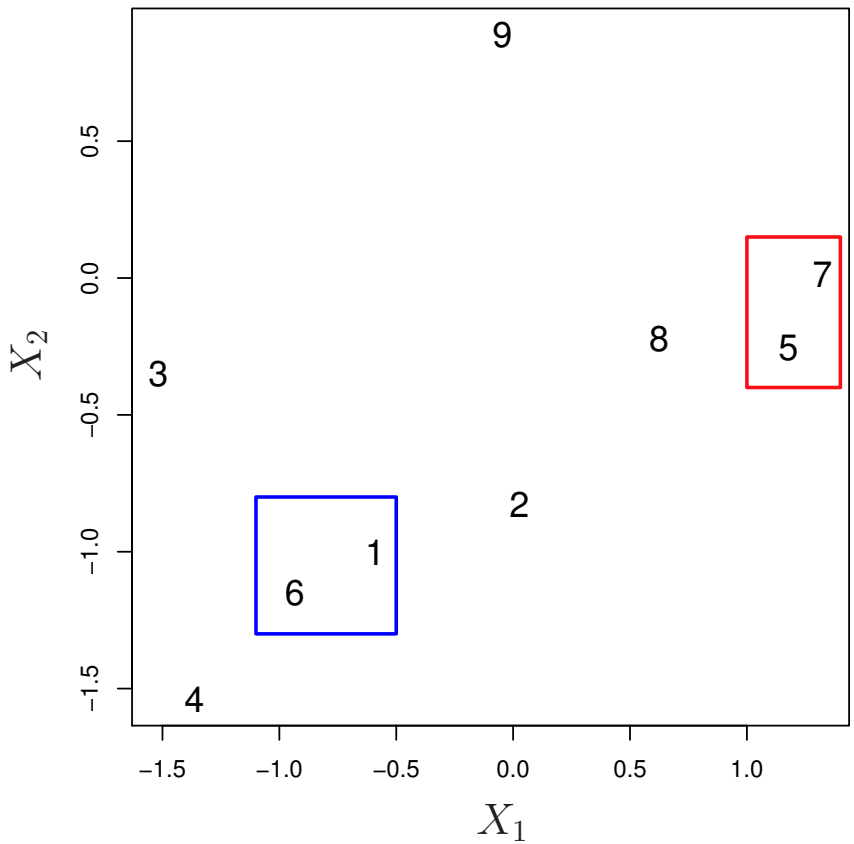
Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.



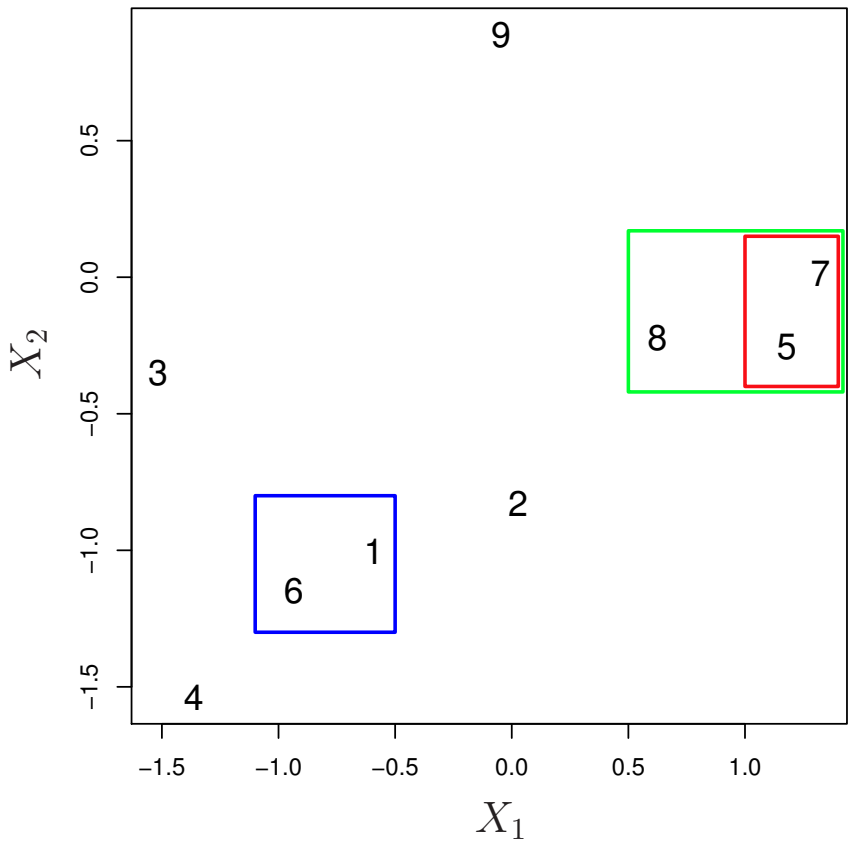
Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.



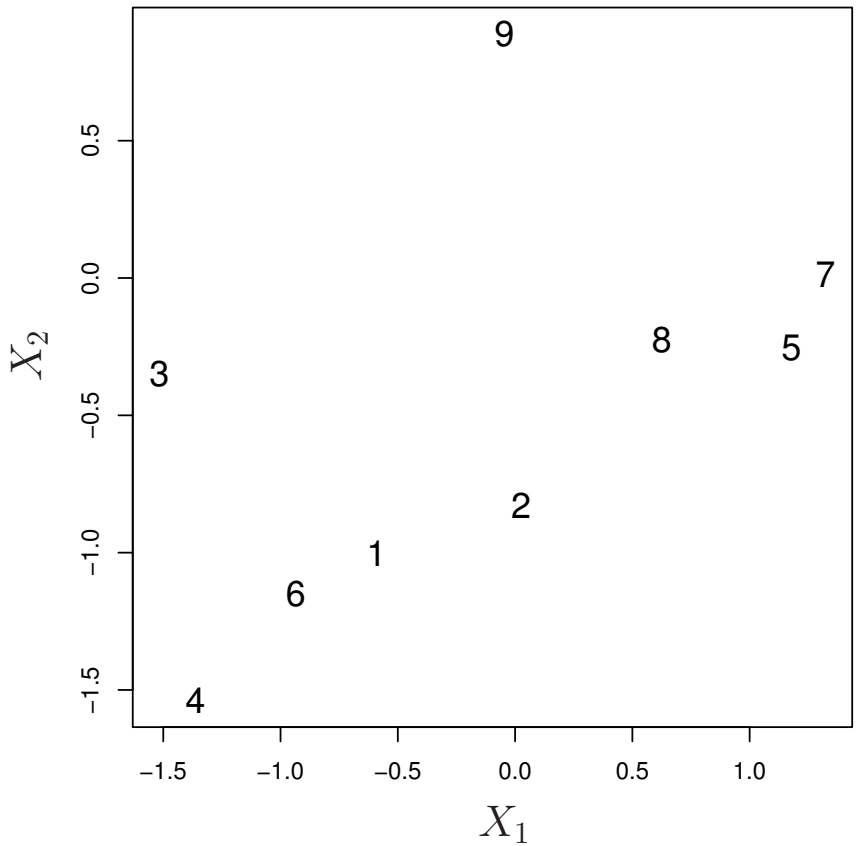
Hierarchical clustering

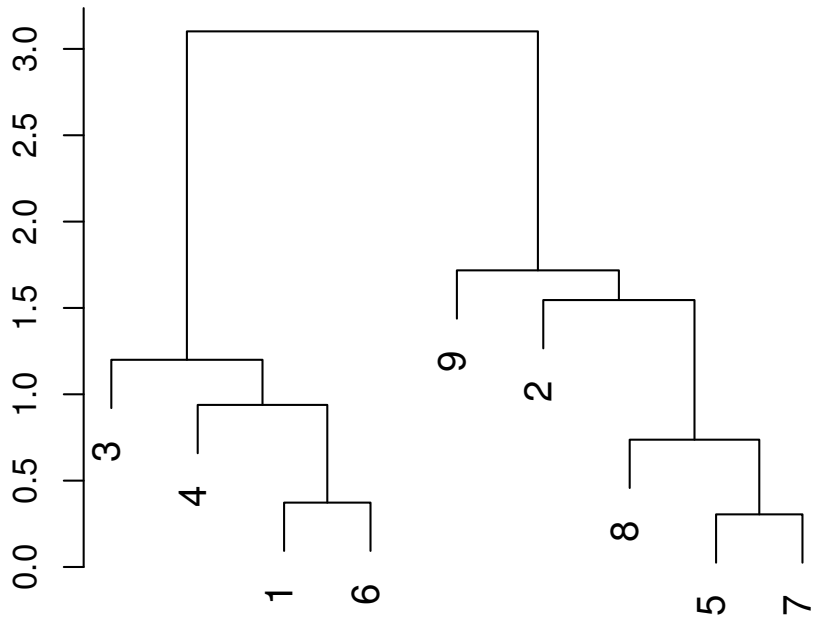
Most algorithms for hierarchical clustering are *agglomerative*.



Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

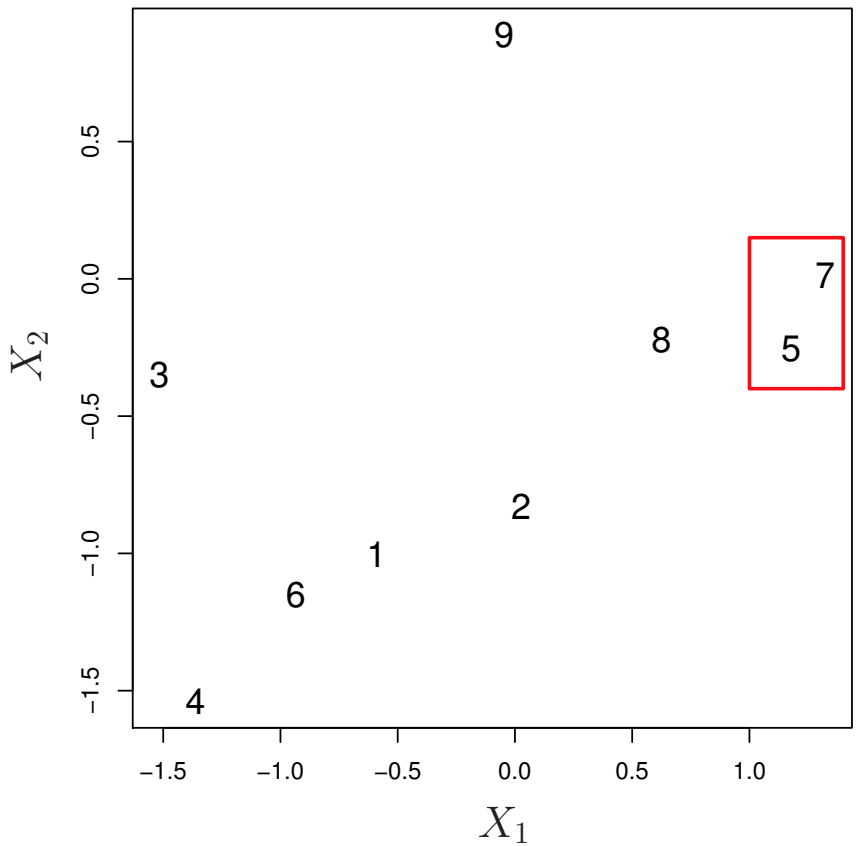


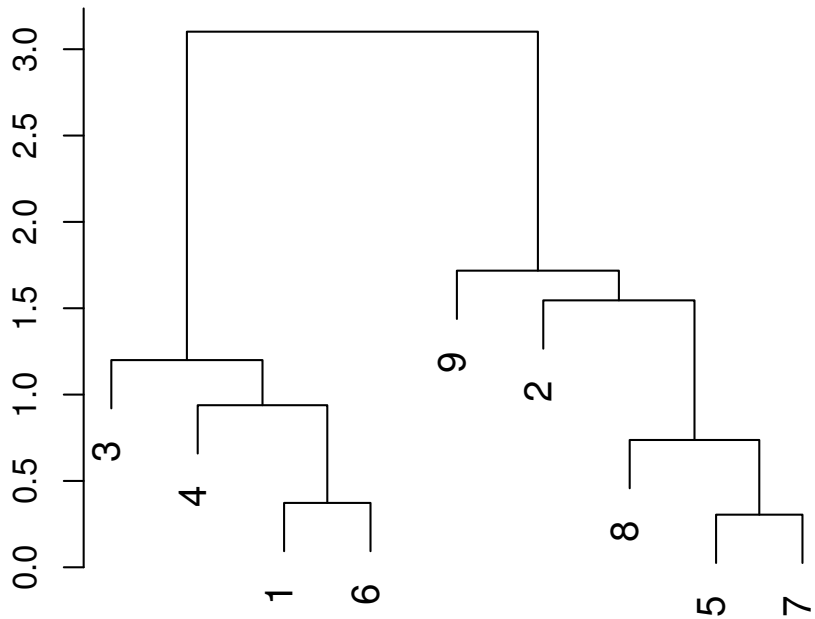


The output of the algorithm is a *dendogram*.

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

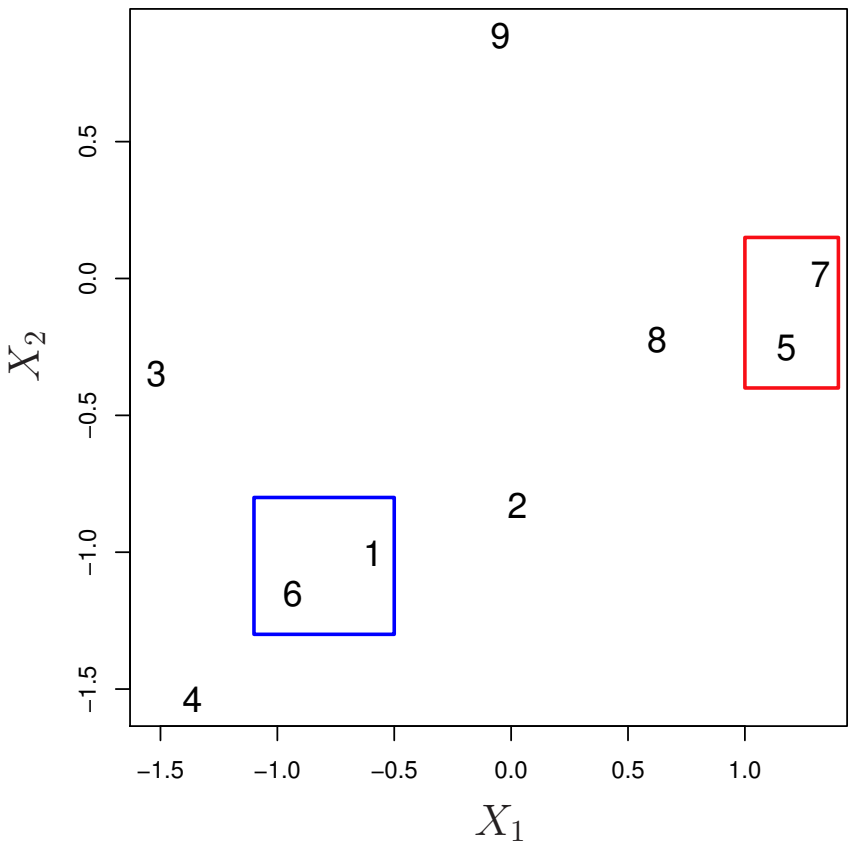


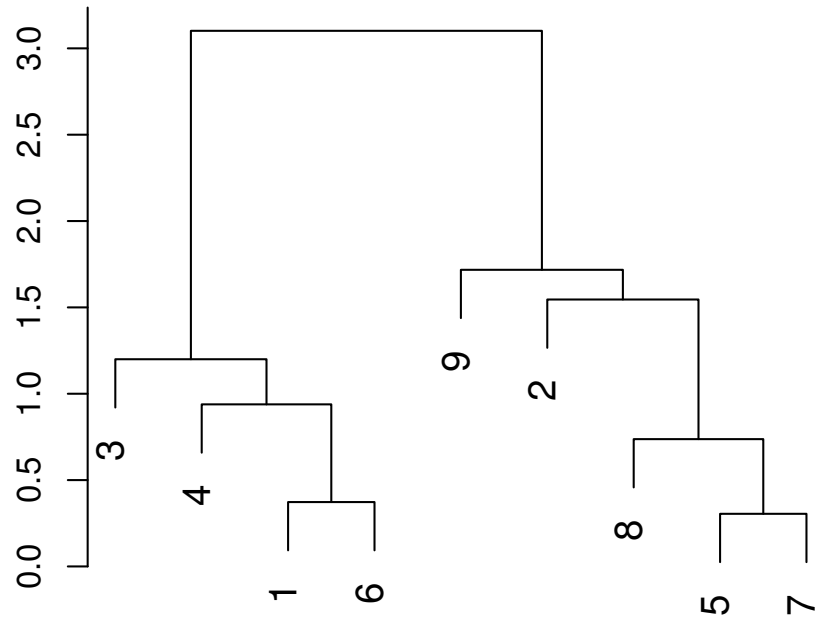


The output of the algorithm is a *dendogram*.

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

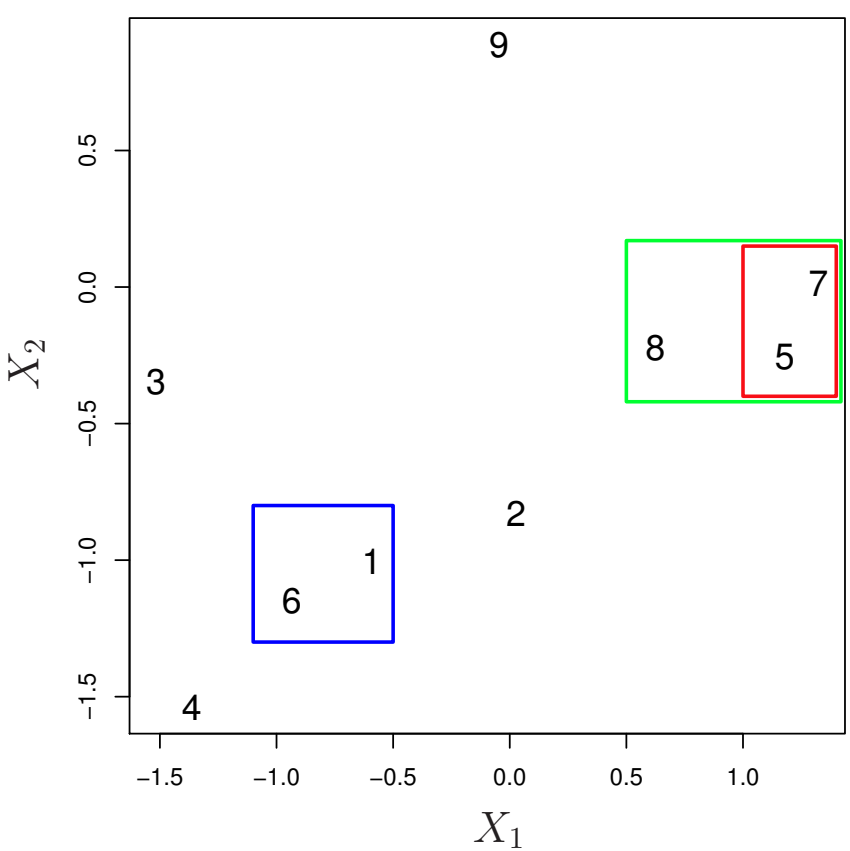


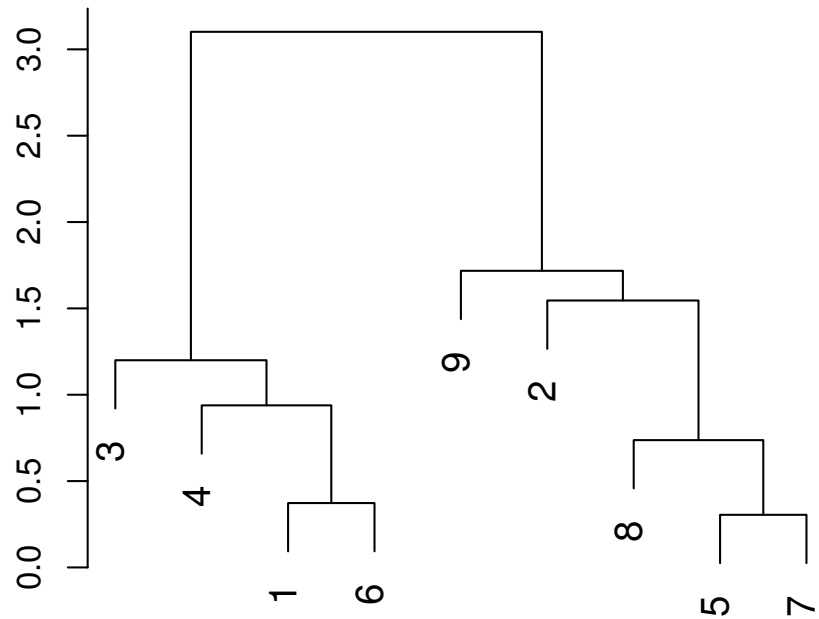


The output of the algorithm is a *dendogram*.

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

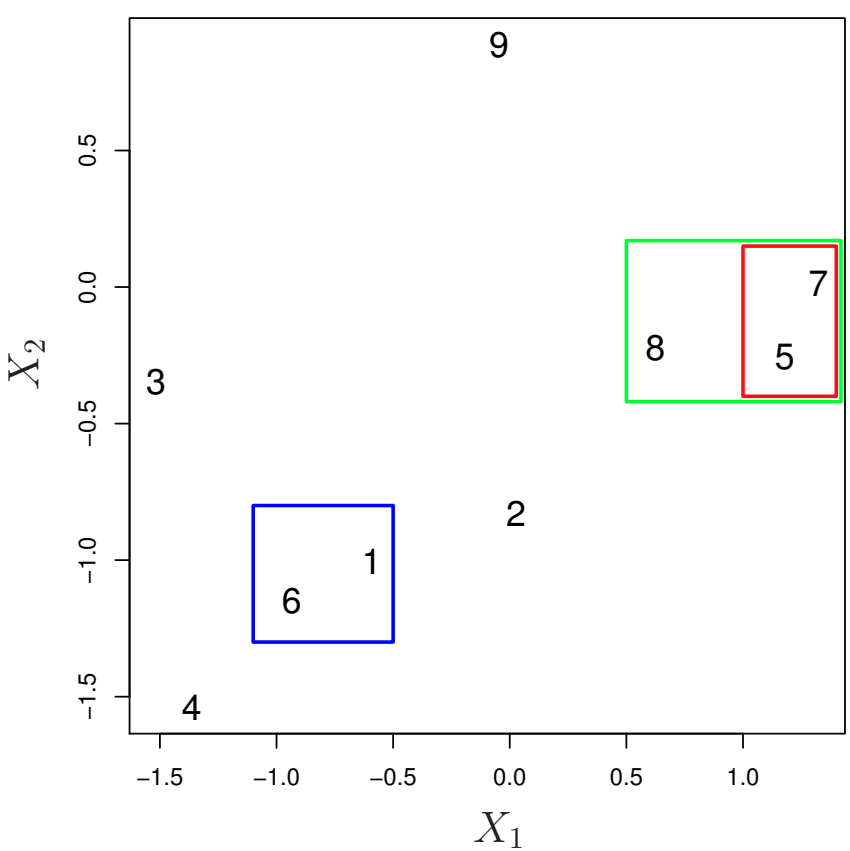


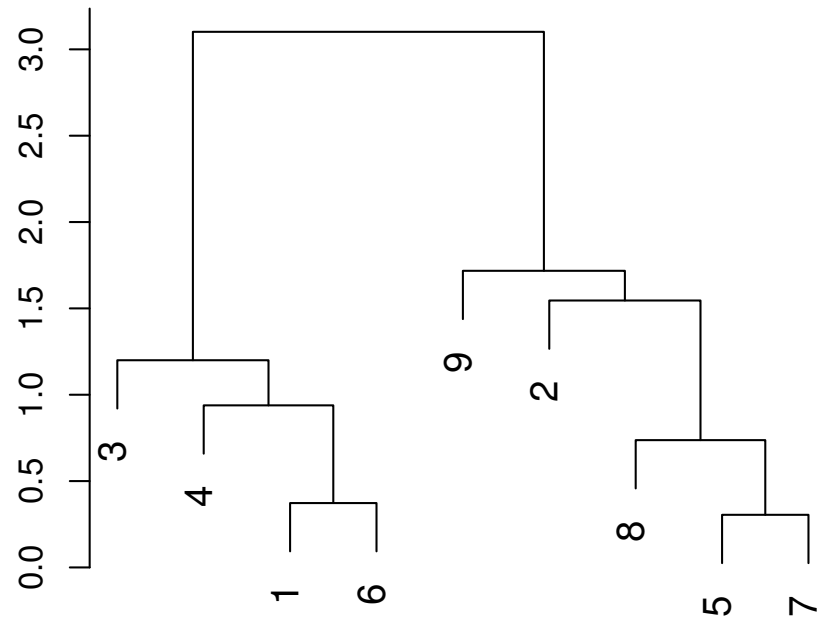


The output of the algorithm is a *dendogram*.

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.





We must be careful about how we interpret the dendogram.

Hierarchical clustering

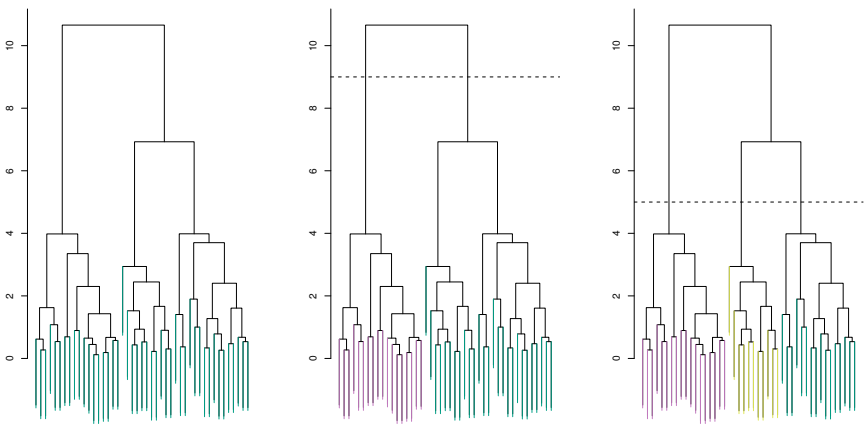


Figure 10.9

- Number of clusters not fixed
- Clusterings are nested
- Dendrogram summarizes relationships amongst clusters

Hierarchical clustering

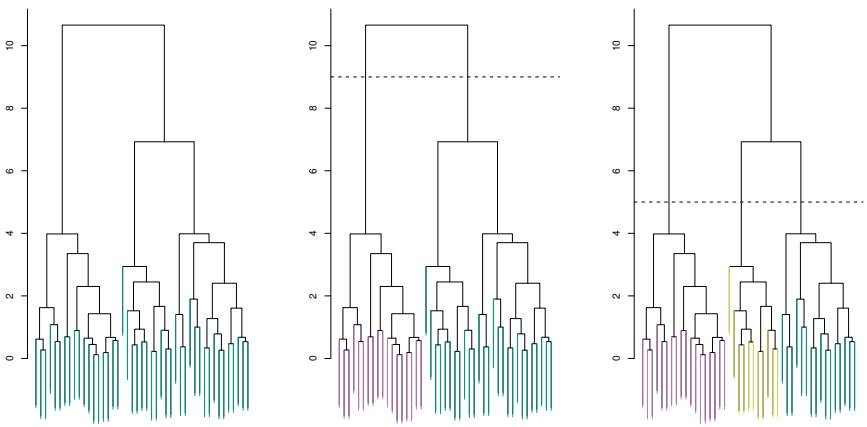


Figure 10.9

- Number of clusters not fixed
- Clusterings are nested
- Dendrogram summarizes relationships amongst clusters
- Hierarchical clustering is not always appropriate.

Hierarchical clustering

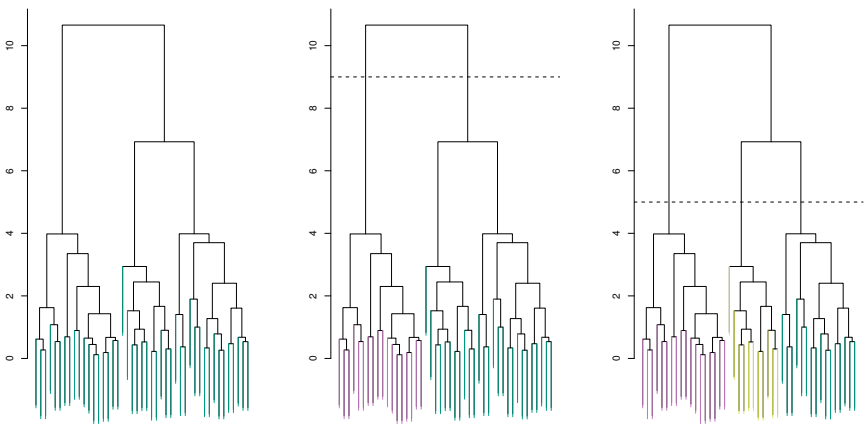


Figure 10.9

- Number of clusters not fixed
- Clusterings are nested
- Dendrogram summarizes relationships amongst clusters
- Hierarchical clustering is not always appropriate.

  e.g., Market segmentation for consumers of 3 different nationalities.
  - Natural 2 clusters: gender
  - Natural 3 clusters: nationality

  These clusterings are not nested / hierarchical.

Notion of distance between clusters

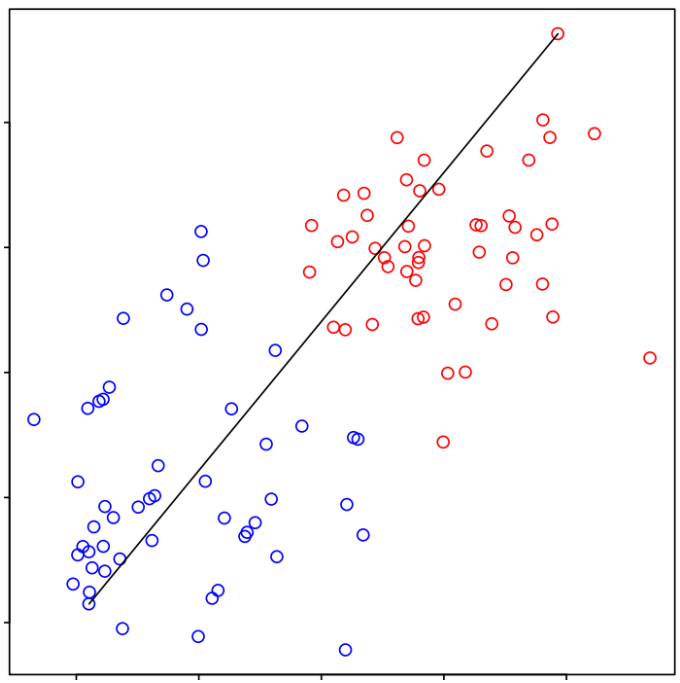
At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

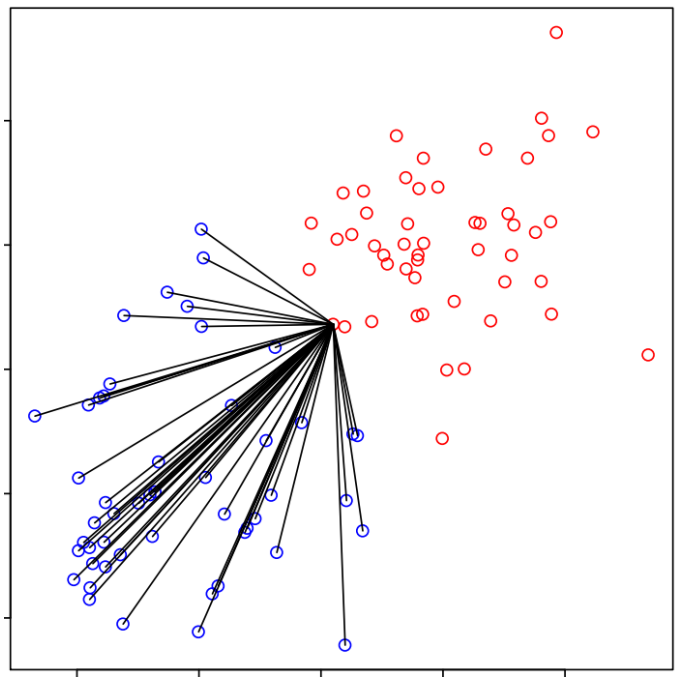


Complete linkage:
The distance between 2 clusters is the *maximum* distance between any pair of samples, one in each cluster.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

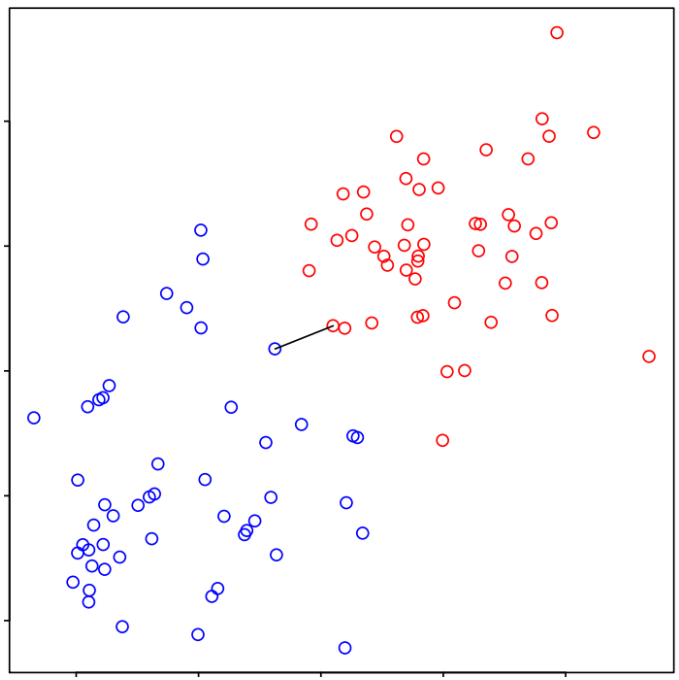


Average linkage:
The distance between 2 clusters is the average of all pairwise distances.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

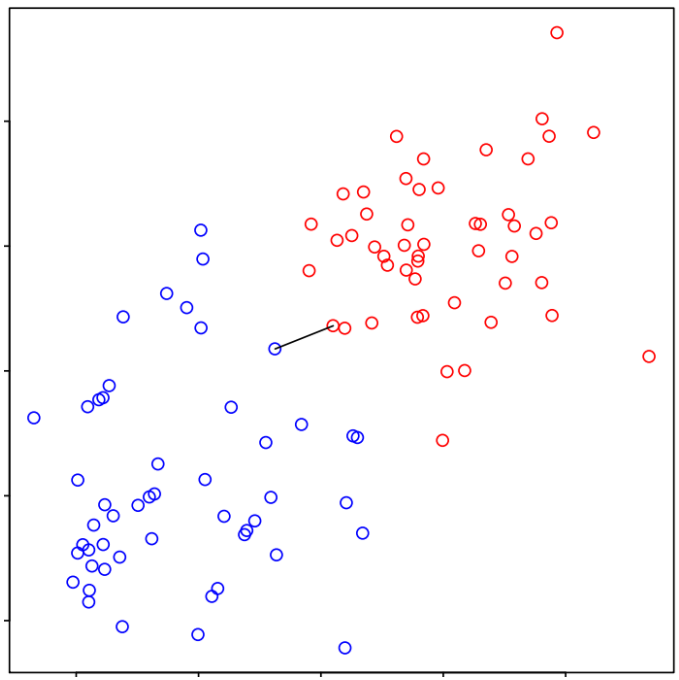


Single linkage:
The distance between 2 clusters is the *minimum* distance between any pair of samples, one in each cluster.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.



Single linkage:
The distance between 2 clusters is the *minimum* distance between any pair of samples, one in each cluster.

Suffers from the *chaining phenomenon*

- Produces long, stringy, or spread out clusters with dissimilar points in same cluster

Example

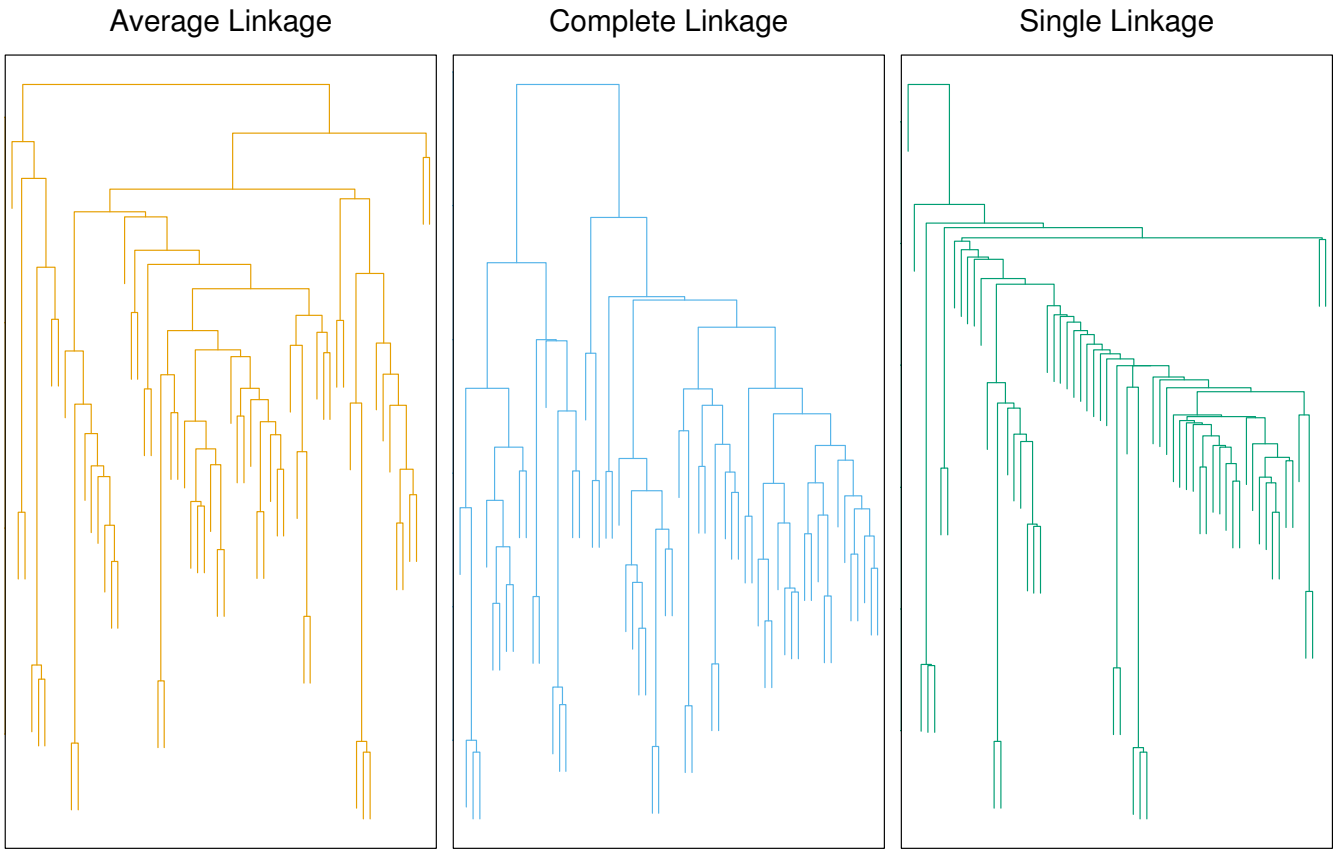


Figure 10.12

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)
  - Heuristics (e.g., the elbow rule)
  - Formal methods (e.g., stability, gap statistics)

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)
  - Heuristics (e.g., the elbow rule)
  - Formal methods (e.g., stability, gap statistics)
- Are the clusters robust?

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)
  - Heuristics (e.g., the elbow rule)
  - Formal methods (e.g., stability, gap statistics)
- Are the clusters robust?
  - Run the clustering on different random subsets of the data. Is the structure preserved?

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)
  - Heuristics (e.g., the elbow rule)
  - Formal methods (e.g., stability, gap statistics)
- Are the clusters robust?
  - Run the clustering on different random subsets of the data. Is the structure preserved?
  - Try different clustering algorithms. Are the conclusions consistent?

Clustering is riddled with questions and choices

- Is (hard) clustering appropriate? Could a sample belong to more than one cluster?
  - Soft clustering, mixture models, feature models, topic models
- How many clusters are appropriate? (No single agreed-upon solution!)
  - Application-dependent choices (based on budget, target level of compression)
  - Heuristics (e.g., the elbow rule)
  - Formal methods (e.g., stability, gap statistics)
- Are the clusters robust?
  - Run the clustering on different random subsets of the data. Is the structure preserved?
  - Try different clustering algorithms. Are the conclusions consistent?
  - Most important: temper your conclusions.

Clustering is riddled with questions and choices

- Should we scale the variables before doing the clustering?

Clustering is riddled with questions and choices

- Should we scale the variables before doing the clustering?
  - Variables with larger variance have a larger effect on the Euclidean distance between two samples.

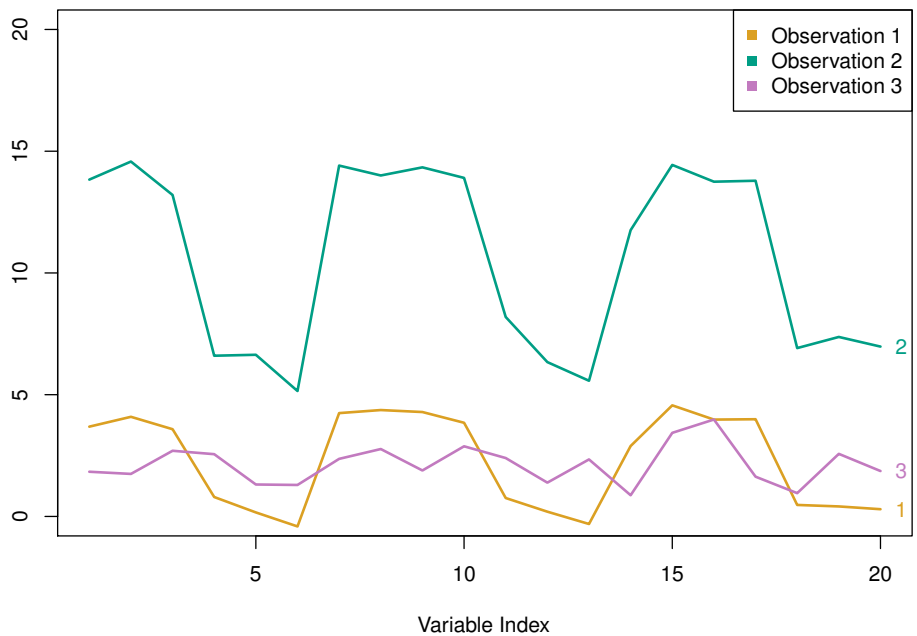
Clustering is riddled with questions and choices

- Should we scale the variables before doing the clustering?
  - Variables with larger variance have a larger effect on the Euclidean distance between two samples.
- Does Euclidean distance capture dissimilarity between samples?

Correlation-based dissimilarity

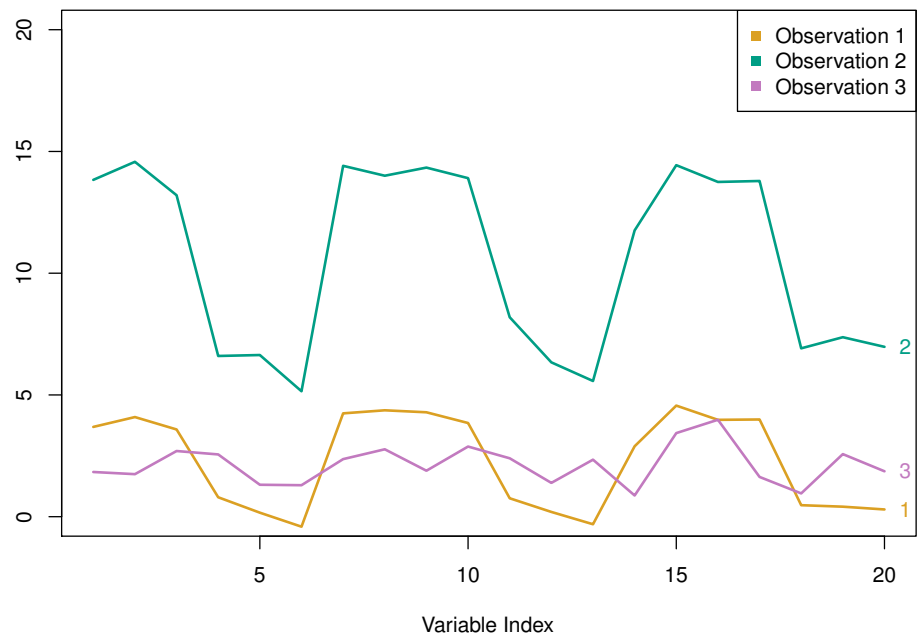
Example: Suppose that we want to cluster customers at a store for market segmentation.

- Samples are customers
- Each variable corresponds to a specific product and measures the number of items bought by the customer during a year.



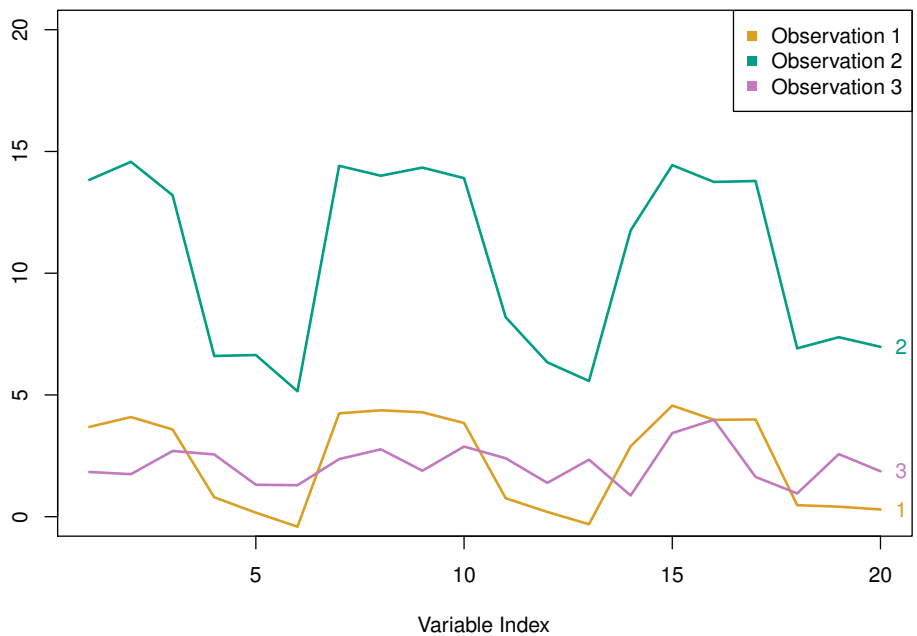
Correlation-based dissimilarity

- Euclidean distance would cluster all customers who purchase few things (orange and purple).



Correlation-based dissimilarity

- Euclidean distance would cluster all customers who purchase few things (orange and purple).
- Perhaps we want to cluster customers who purchase *similar* things (orange and teal).



Correlation-based dissimilarity

- Euclidean distance would cluster all customers who purchase few things (orange and purple).
- Perhaps we want to cluster customers who purchase *similar* things (orange and teal).
- Then, the **correlation distance** may be a more appropriate measure of dissimilarity between samples.

