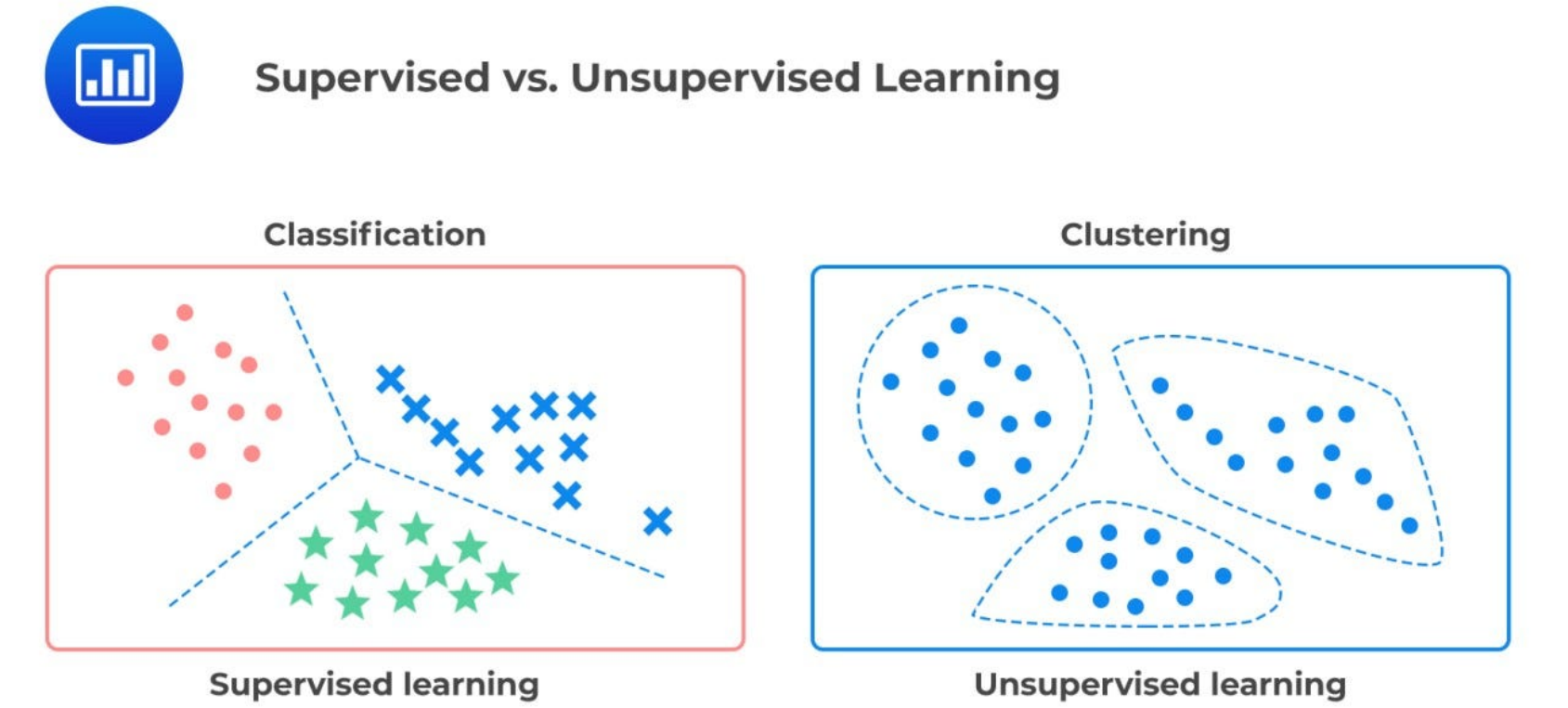
Chapter 12: Unsupervised Learning

Yonghyun Kwon

Department of Mathematics, Korea Military Academy

Unsupervised Learning

- Most of this course focuses on *supervised learning* methods such as regression and classification.
- In that setting we observe both a set of features $X_1, X_2, \ldots, X_p$ for each object, as well as a response variable Y. The goal is to **predict** Y using $X_1, \ldots, X_p$.
- Here we instead focus on *unsupervised learning*, where we observe only the features $X_1, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable Y.



https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/

The Goals of Unsupervised Learning

- Discover interesting **structure** in the data:
  - Is there an informative way to visualize the data?
  - Can we discover **subgroups** among the variables or among the observations?
- We discuss two important methods:
  - *Principal Components Analysis (PCA)* for visualization and dimensionality reduction.
  - *Clustering* for discovering unknown subgroups in data.
- Unsupervised learning is more **subjective** than supervised learning, since there is no simple goal like predicting Y.

Principal Components Analysis

Principal Components Analysis (PCA)

- *Principal Components Analysis (PCA)* produces a **low-dimensional representation** of a dataset.
- It finds a sequence of linear combinations of the variables that have **maximal variance** and are mutually uncorrelated.

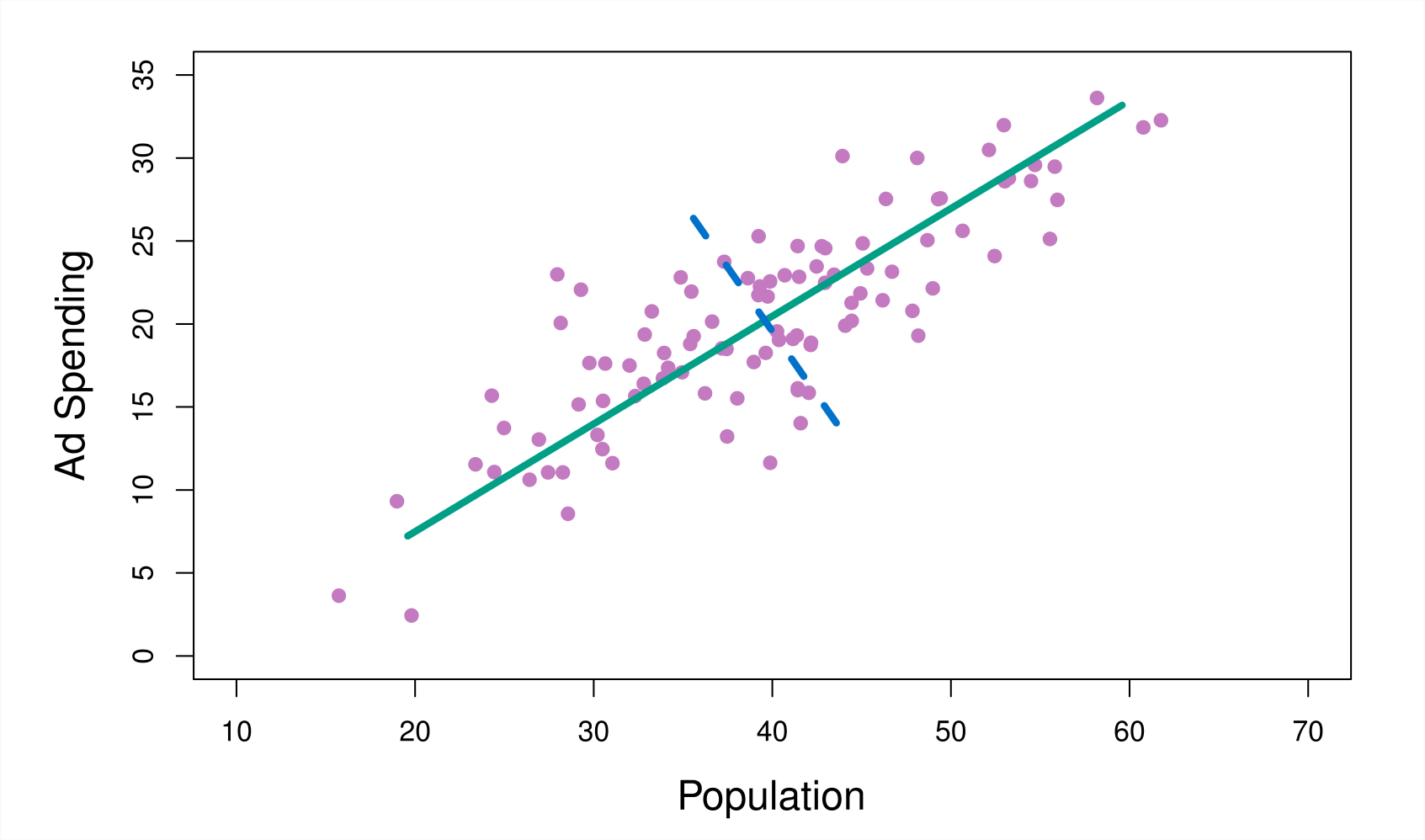
Principal Components Analysis (PCA)

- For a random vector $\boldsymbol{X} \in \mathbb{R}^p$, consider reducing its dimension.
- The **first principal component** is the linear combination

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p = \boldsymbol{\phi}_1^\top \boldsymbol{X},$$

  that has the largest variance, with normalization $\sum_{j=1}^p \phi_{j1}^2 = 1$.
- The elements $\phi_{11}, \ldots, \phi_{p1}$ are the *loadings* of the first principal component; together they form the *loading vector* $\boldsymbol{\phi}_1 = (\phi_{11}, \ldots, \phi_{p1})^\top$.

PCA Example



The green solid line indicates the first principal component direction; the blue dashed line indicates the second.

First Principal Component

- Suppose a random vector $\boldsymbol{X}$ with mean μ and variance matrix $\boldsymbol{\Sigma}$.
- The first loading vector solves

$$\phi_1 = \underset{\phi \in \mathbb{R}^p, \|\phi\|=1}{\operatorname{argmax}} \operatorname{Var}(\phi^\top \boldsymbol{X}) = \underset{\phi \in \mathbb{R}^p, \|\phi\|=1}{\operatorname{argmax}} \phi^\top \operatorname{Var}(\boldsymbol{X}) \phi.$$

- The resulting $Z_1 = \phi_1^\top \boldsymbol{X}$ is the *first principal component score*
- The loading vector ϕ_1 defines a direction in feature space along which the data vary the most.

k-th Principal Component

Given the first $k-1$ PC loading vectors $\phi_1, \cdots, \phi_{k-1}$, the kth PC loading vector is the unit vector $\phi_k \in \mathbb{R}^p$ that

- maximizes the variance of $\phi_k \boldsymbol{X}$;
- is orthogonal to the $k-1$ PC loading vectors $\phi_1, \cdots, \phi_{k-1}$.

That is,

$$\phi_k = \underset{\substack{\phi \in \mathbb{R}^p, \|\phi\|=1 \\ \phi^\top \phi_j = 0, j=1,\cdots,k-1}}{\operatorname{argmax}} \operatorname{Var}(\phi^\top \boldsymbol{X}).$$

- This can be solved via *eigen value decomposition* of $\operatorname{Var}(\boldsymbol{X}) = \boldsymbol{\Sigma}$.
- The resulting $Z_k = \phi_k^\top \boldsymbol{X}$ is the k-th *principal component score*.
- By construction, $\operatorname{Corr}(Z_i, Z_j) = 0$ for $\forall i \neq j$.

Relation to Eigen-decomposition of Σ (First PC)

The first PC loading vector ϕ_1 solves

$$\phi_1 = \underset{\|\phi\|=1}{\operatorname{argmax}} \operatorname{Var}(\phi^\top \boldsymbol{X}) = \underset{\|\phi\|=1}{\operatorname{argmax}} \phi^\top \boldsymbol{\Sigma} \phi.$$

- Using a *Lagrange multiplier* λ, we define

$$\Phi(\phi, \lambda) = \phi^\top \boldsymbol{\Sigma} \phi - \lambda(\phi^\top \phi - 1).$$

- Setting the derivative to zero gives

$$\boldsymbol{\Sigma} \phi = \lambda \phi.$$

- Thus, ϕ_1 is the *eigenvector* of $\boldsymbol{\Sigma}$ corresponding to the largest *eigenvalue* λ_1:

$$\operatorname{Var}(Z_1) = \lambda_1, \quad Z_1 = \phi_1^\top \boldsymbol{X}.$$

Relation to Eigen-decomposition of Σ (Remaining PCs)

Given the first $k-1$ PC directions $\phi_1, \ldots, \phi_{k-1}$, the k-th PC loading vector ϕ_k is the unit vector ϕ that

$$\phi_k = \underset{\substack{\|\phi\|=1, \\ \phi^\top \phi_j = 0,\ j<k}}{\operatorname{argmax}} \quad \phi^\top \Sigma \phi.$$

- Using the *Lagrangian function*

$$\Phi(\phi, \lambda, \gamma) = \phi^\top \Sigma \phi - \lambda(\phi^\top \phi - 1) - \sum_{j=1}^{k-1} \gamma_j \phi_j^\top \phi,$$

  the stationary condition yields

$$\Sigma \phi = \lambda \phi \quad \text{with} \quad \phi^\top \phi_j = 0 \ (j < k).$$

- Hence, each ϕ_k is an *eigenvector* of Σ, with variance $\mathrm{Var}(Z_k) = \lambda_k$.
- The eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$.

Sample PCA

Given multivariate data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the *sample PCA* finds orthogonal directions of maximal sample variance.

- The sample variance matrix is

$$\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top, \qquad \bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i.$$

- Eigen-decomposition:

$$\boldsymbol{S}\hat{\boldsymbol{u}}_k = \hat{\lambda}_k \hat{\boldsymbol{u}}_k, \qquad \hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_p.$$

- The kth *PC score* and its variance:

$$z_{ik} = \hat{\boldsymbol{u}}_k^\top(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \qquad \mathrm{Var}(z_{ik}) = \hat{\lambda}_k.$$

Illustration

- `USArrests` data: for each of 50 U.S. states, the number of arrests per 100,000 residents for `Assault`, `Murder`, `Rape`; plus `UrbanPop` (percent living in urban areas).
- **Principal component score vectors** z_k have length $n = 50$; **loading vectors** ϕ_k have length $p = 4$.
- PCA was performed after **standardizing** each variable to have mean zero and standard deviation one.

USArrests data: PCA plot

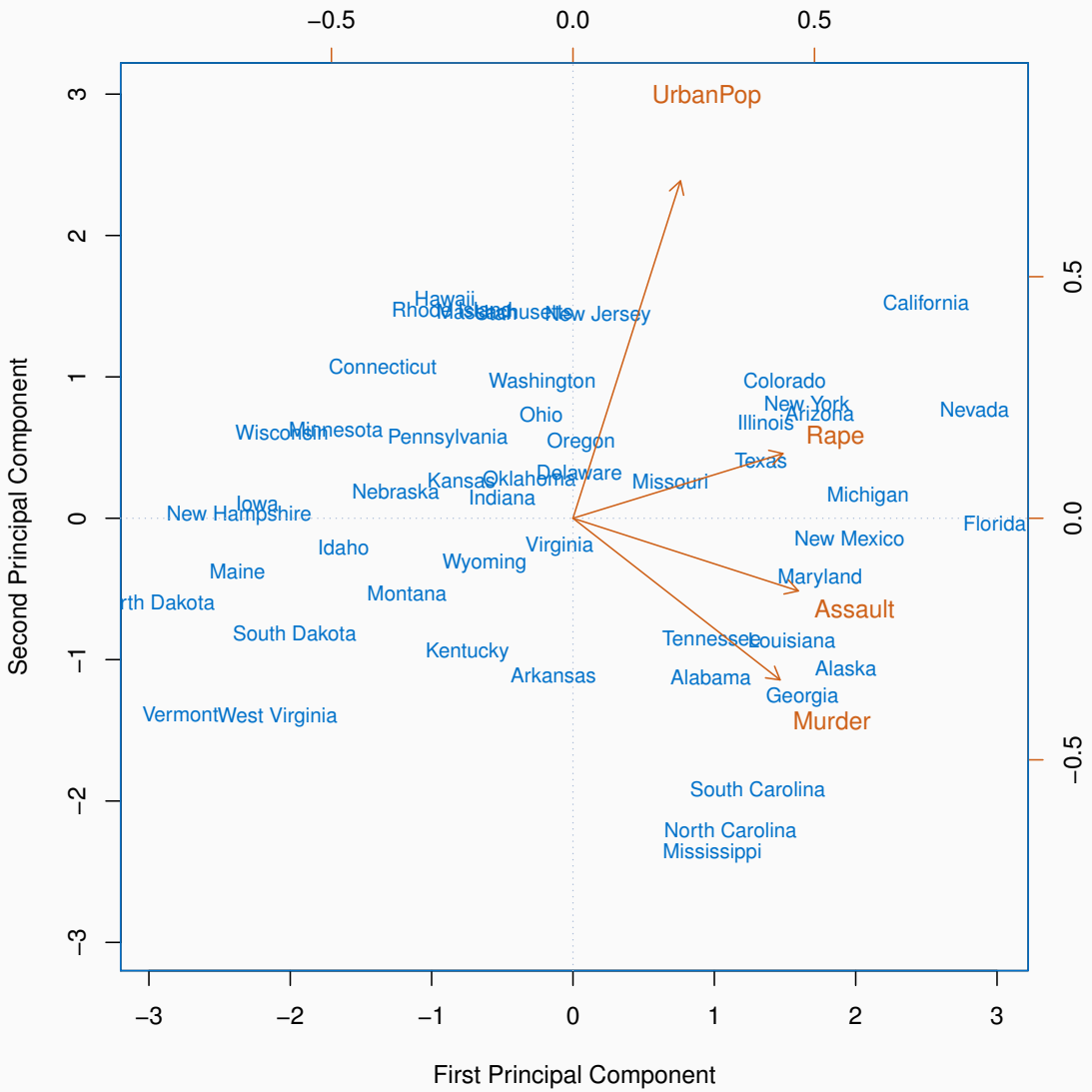


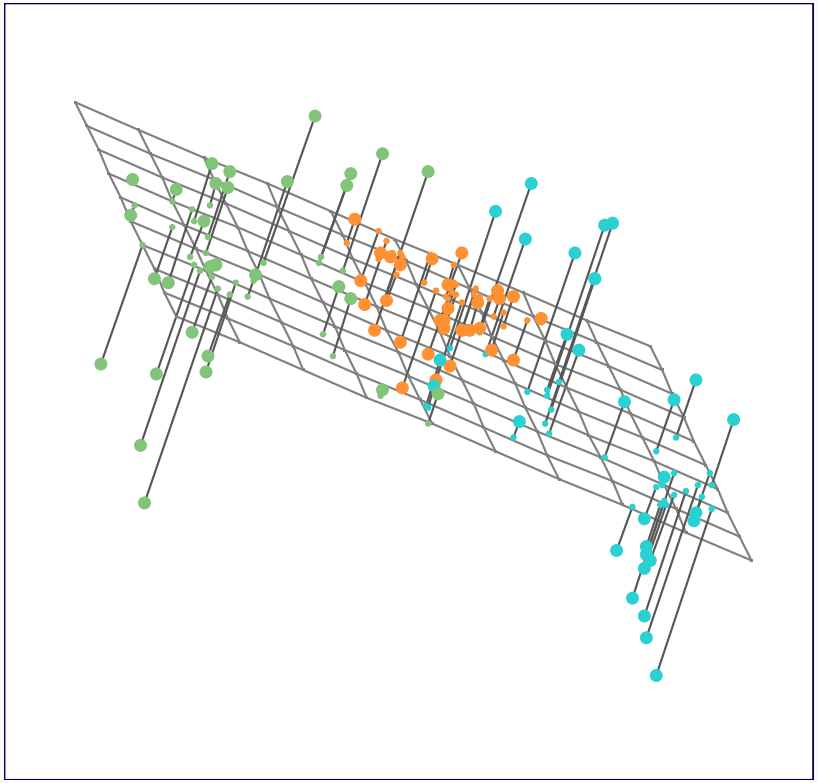
Figure details

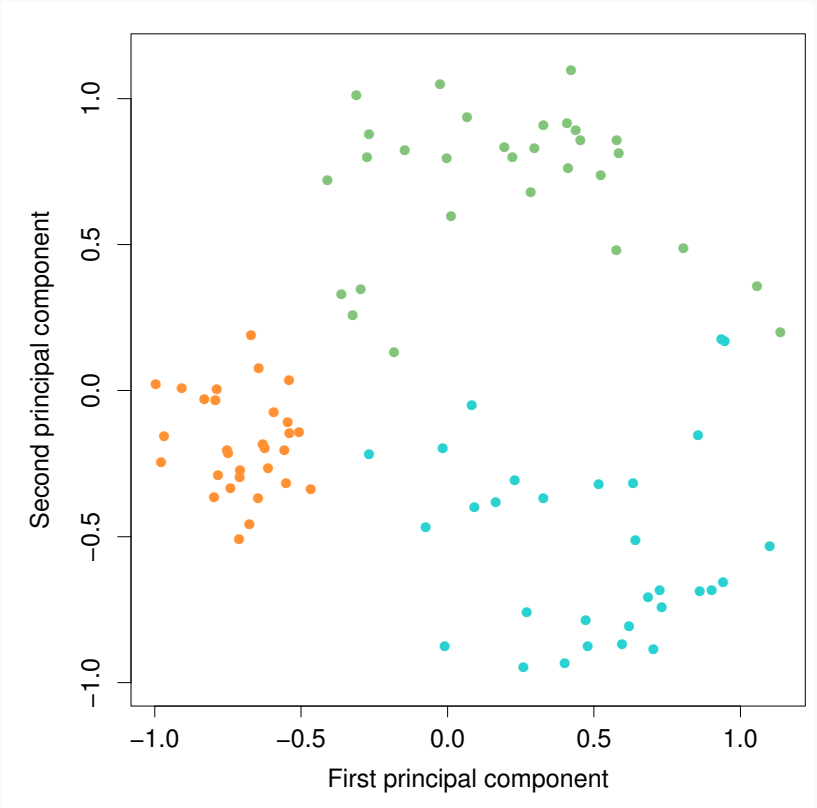
- Blue state names are the **scores** for PC1 and PC2.
- Orange arrows indicate the first two **loading vectors**. The word for a variable is centered at its pair of loadings.

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

- Example: the loading for *Rape* on PC1 is 0.54 and on PC2 is 0.17 (so the label is at (0.54, 0.17)).

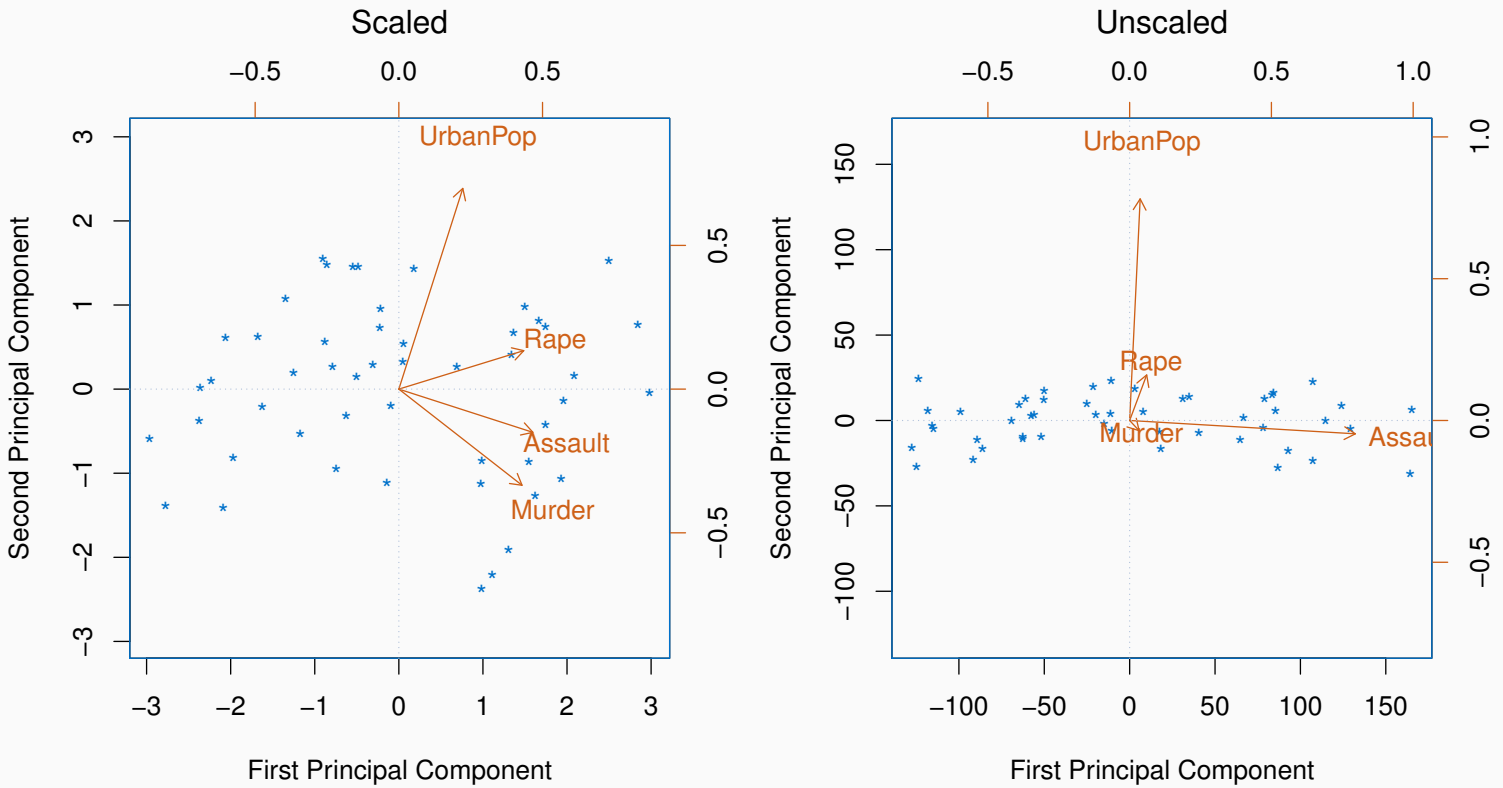
Another interpretation of principal components





Scaling of the variables matters

- If variables are in different units, **scale** each to have standard deviation one.
- If variables share the same units, scaling may or may not be appropriate.



Proportion Variance Explained (PVE)

- To understand the strength of each component, consider the *proportion of variance explained (PVE)* by each PC.
- With centered variables, total variance is

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2.$$

- Variance explained by the mth PC is

$$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2.$$

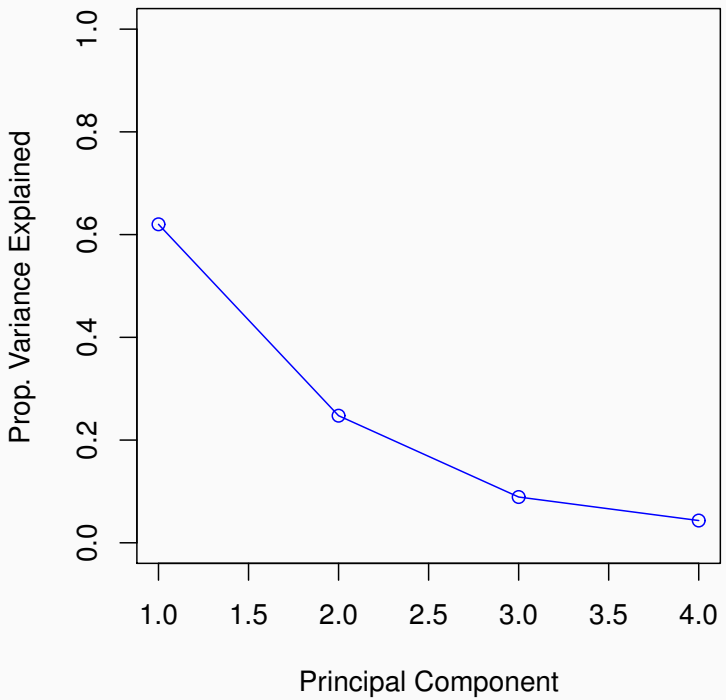
- It can be shown that

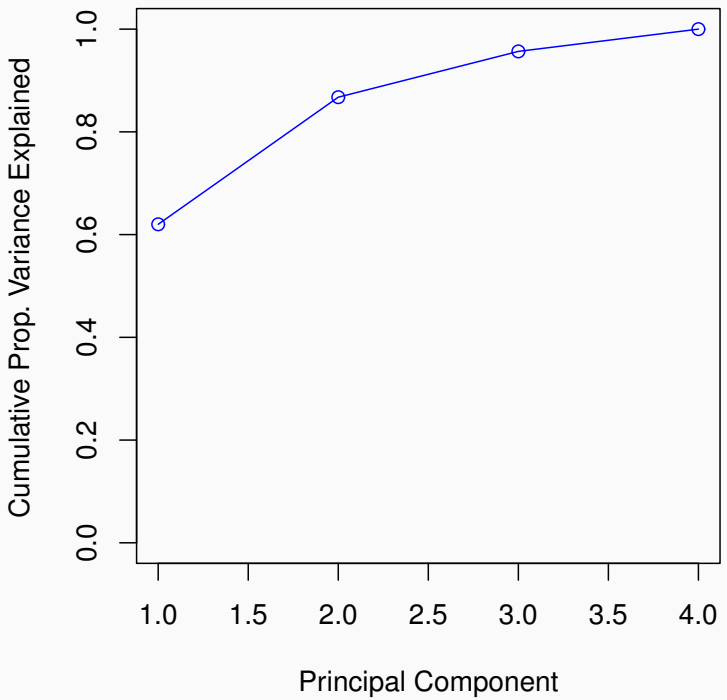
$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(Z_m), \quad M = \min(n-1, p).$$

Proportion Variance Explained: continued

Therefore, the *PVE* of the mth principal component is

$$\text{PVE}_m = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2}.$$





How many principal components should we use?

- If we use principal components as a summary of our data, how many components are sufficient?
- There is **no simple answer**: *cross-validation* is not directly applicable here.
- When could cross-validation be used to select the number of components?
    - When PCA is used as a pre-processing step within a supervised learning problem.
- The **scree plot** (previous slide) can guide us — we look for an "elbow" in the curve.

Missing Values and Matrix Completion

Matrix Completion and Missing Values

- Data matrices X often contain *missing entries*, represented as `NA`s.
- Many modeling procedures (e.g., regression, GLMs) require complete data.
- A simple approach: **mean imputation**, replacing missing values with the variable's mean.
- But this ignores correlations among variables; we can exploit them for better imputation.
- Assume missingness is random (not informative).
- We present an approach based on *principal components*.

Recommender Systems

	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	⋯
Customer 1	•	•	•	•	4	•	•	•	•	•	⋯
Customer 2	•	•	3	•	•	•	3	•	•	3	⋯
Customer 3	•	2	•	4	•	•	•	•	2	•	⋯
Customer 4	3	•	•	•	•	•	•	•	•	•	⋯
Customer 5	5	1	•	•	4	•	•	•	•	•	⋯
Customer 6	•	•	•	•	•	2	4	•	•	•	⋯
Customer 7	•	•	5	•	•	•	•	3	•	•	⋯
Customer 8	•	•	•	•	•	•	•	•	•	•	⋯
Customer 9	3	•	•	•	5	•	•	1	•	•	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

- Netflix example: users rate movies from 1–5. This forms a huge $n \times p$ matrix (n = users, p = movies) with mostly missing entries.
- Each user rates only a small subset of movies.

PCA as a Matrix Approximation Problem

- PCA can be viewed as finding a low-rank approximation to the data matrix $\boldsymbol{X} = (x_{ij})_{n\times p}$:

$$\min_{A\in\mathbb{R}^{n\times M},\, B\in\mathbb{R}^{p\times M}} \sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - \sum_{m=1}^{M} a_{im}b_{jm}\right)^2 .$$

- The solution is obtained by the first M principal components:

$$\hat{a}_{im} = z_{im} \quad \text{(scores)}, \qquad \hat{b}_{jm} = \phi_{jm} \quad \text{(loadings)}.$$

- The rank-M reconstruction is

$$\hat{x}_{ij} = \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm}.$$

- This least-squares formulation provides a natural starting point for *matrix completion*, where some x_{ij} are missing.

Matrix Completion via Principal Components

- When $\boldsymbol{X}$ has missing entries, modify the PCA objective to use only observed elements:

$$\min_{A,B} \sum_{(i,j)\in\mathcal{O}} \left(x_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2,$$

  where $\mathcal{O}$ is the set of observed index pairs.

- The fitted value for a missing entry is

$$\hat{x}_{ij} = \sum_{m=1}^{M} \hat{a}_{im} \hat{b}_{jm}, \quad (i,j) \notin \mathcal{O}.$$

- The estimated $\hat{A}$ and $\hat{B}$ play the same role as *scores* and *loadings* in PCA.

Iterative Algorithm for Matrix Completion

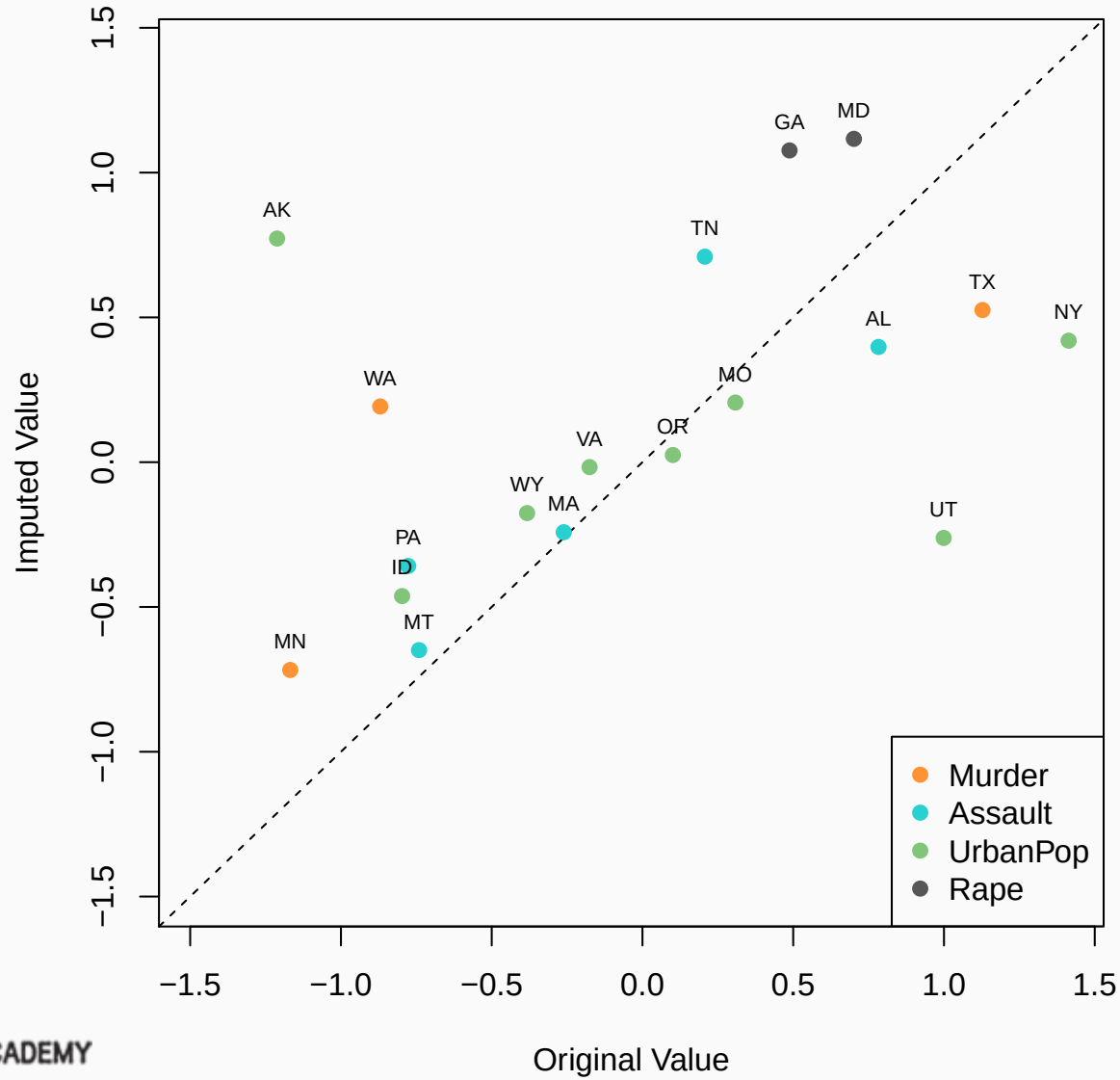
1. **Initialize:** Replace missing entries with column means to form a complete matrix $\tilde{\boldsymbol{X}}$.
2. **Iterate until convergence:**
   2.1 Perform PCA on $\tilde{\boldsymbol{X}}$ to minimize
   $$\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\tilde{x}_{ij}-\sum_{m=1}^{M}a_{im}b_{jm}\right)^2.$$
   2.2 Update missing values:
   $$\tilde{x}_{ij}\leftarrow\sum_{m=1}^{M}\hat{a}_{im}\hat{b}_{jm},\quad (i,j)\notin\mathcal{O}.$$
   2.3 Check if the objective on observed entries decreases.
3. Return $\tilde{x}_{ij}$ for missing (i,j).

Example: USArrests Data

- X: 50 states $\times$ 4 variables (Murder, Assault, UrbanPop, Rape).
- Randomly selected 20 states; for each, one variable was set to NA.
- Used $M = 1$ principal component for imputation.

Example: USArrests Data

Correlation between true and imputed values = 0.63.



Clustering Methods

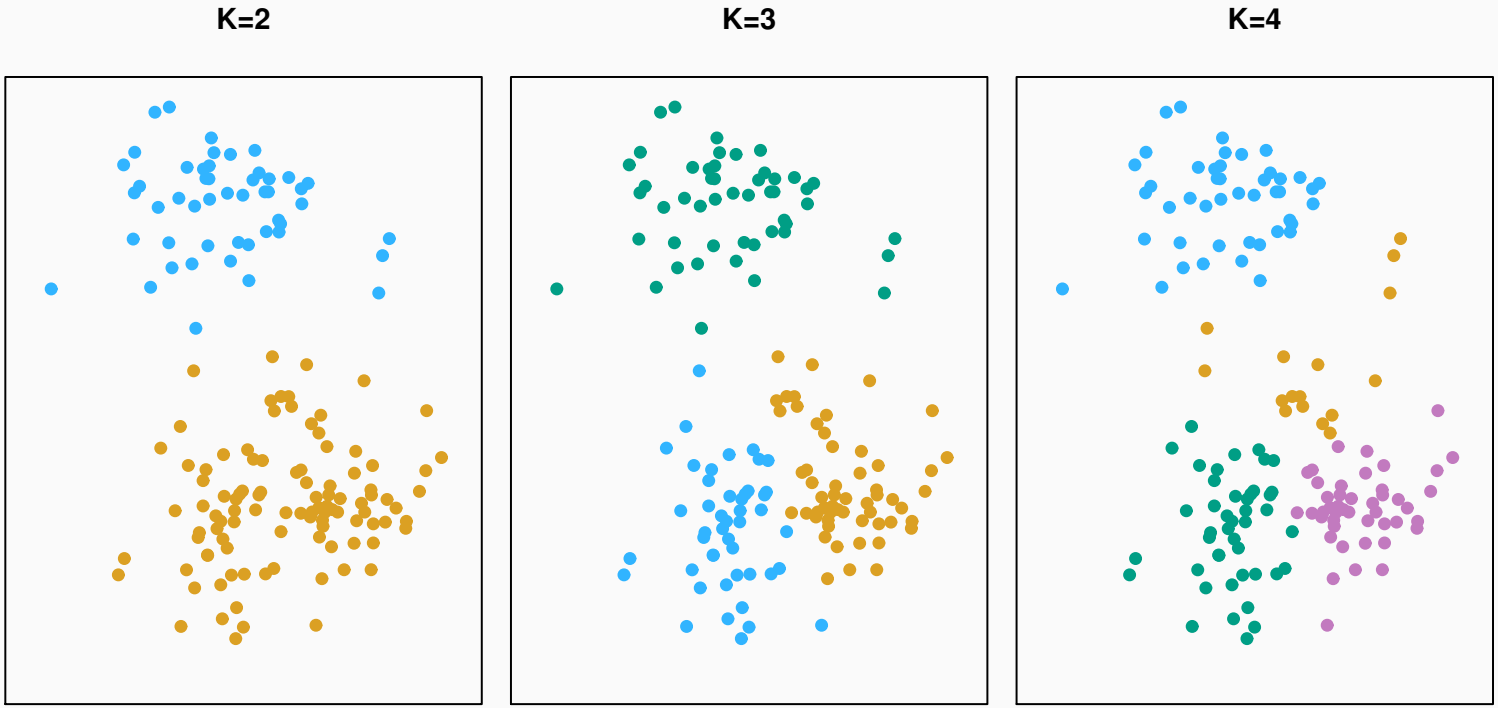
Clustering

- *Clustering* refers to a broad set of techniques for finding **subgroups** or clusters in a dataset.
- We seek a **partition** of the data into distinct groups so that observations within each group are **similar** to each other.
- To make this concrete, we must define what it means for two observations to be similar or different.
- This is often **domain-specific** and depends on the data being studied.

Two clustering methods

- *K-means clustering*: partition the observations into a pre-specified number K of clusters.
- *Hierarchical clustering*: number of clusters is not specified in advance; produces a **tree-like dendrogram** showing all possible clusterings from 1 to n.

K-means clustering



A simulated dataset with 150 observations in 2D space. Panels show results of applying K-means clustering with different K values. Each color shows a cluster assignment. The cluster colors are arbitrary — they were not used in clustering but are the **output** of the algorithm.

Details of K-means clustering

Let $C_1, C_2, \ldots, C_K$ denote the index sets of observations in each cluster.

- These sets satisfy:

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

- Thus, each observation belongs to exactly one cluster.
- Example: if the ith observation belongs to cluster k, then $i \in C_k$.

K-means Clustering Objective

- The goal of K-means is to partition observations into K clusters $\{C_1, \ldots, C_K\}$ that have **small within-cluster variation**.
- Define the *within-cluster variation* for cluster C_k as

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

  using the *Euclidean distance*.
- The **K-means objective function** minimizes the total within-cluster variation:

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

- This optimization defines the *K-means clustering solution*.

K-Means Clustering Algorithm

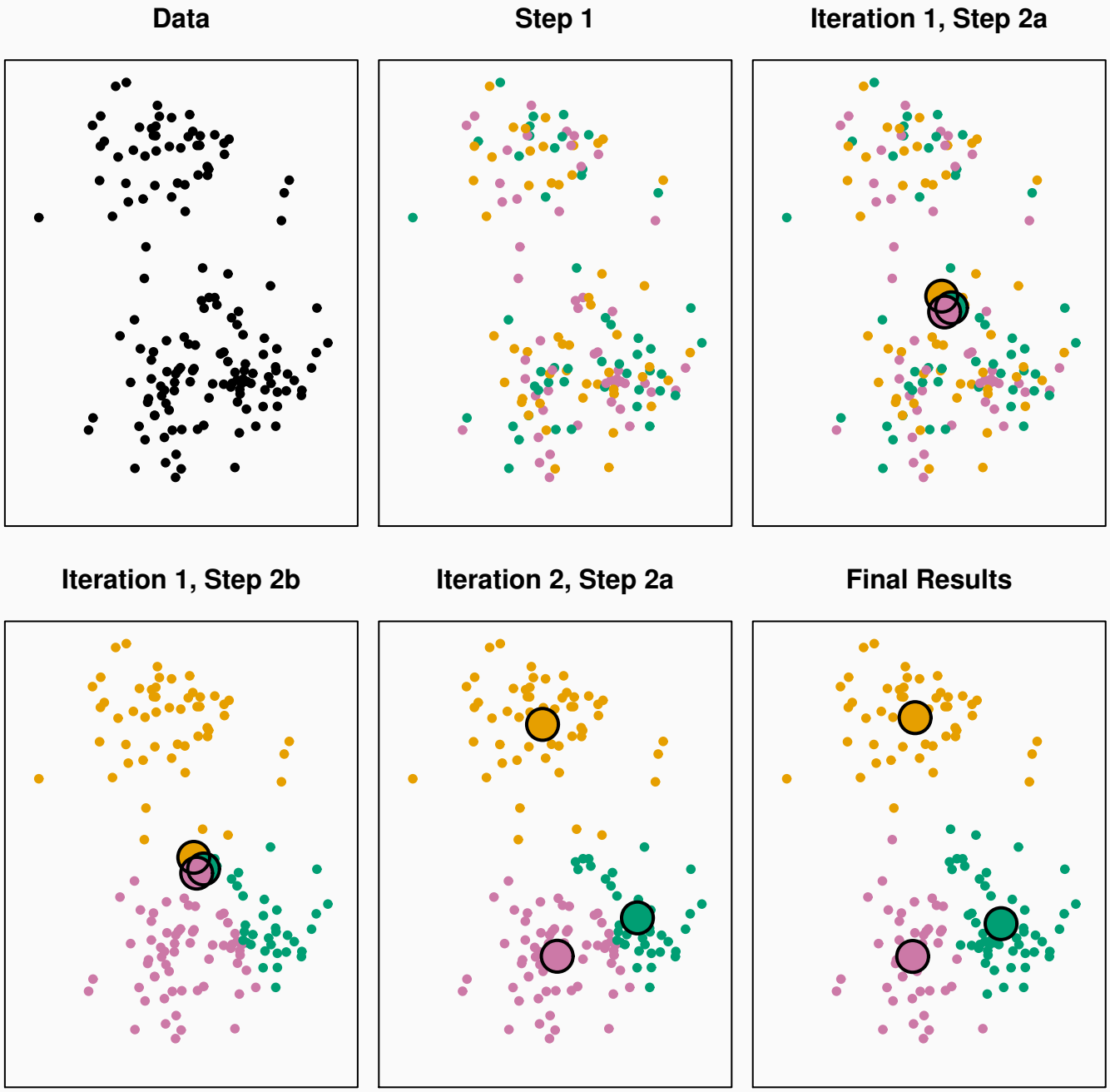
1. **Random initialization:** Randomly assign a number from 1 to K to each observation. These serve as initial cluster assignments.
2. **Iterate** until the cluster assignments stop changing:
   - 2.1 For each of the K clusters, compute the *cluster centroid*, the vector of feature means for observations in that cluster.
   - 2.2 Assign each observation to the cluster whose centroid is **closest** (based on Euclidean distance).

Properties of the Algorithm

- This algorithm is guaranteed to **decrease the objective function** at each step.
- Recall the K-means 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 $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$ is the mean of feature j in cluster C_k.
- However, it is **not guaranteed to reach the global minimum**, because the solution may depend on initial assignments.


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

Details of Previous Figure

- **Top left:** Original data.
- **Top center:** Step 1 — random cluster assignments.
- **Top right:** Step 2(a) — compute centroids (large colored disks).
- **Bottom left:** Step 2(b) — reassign each point to the nearest centroid.
- **Bottom center:** Step 2(a) repeated with new centroids.
- **Bottom right:** Results after 10 iterations — final cluster structure.

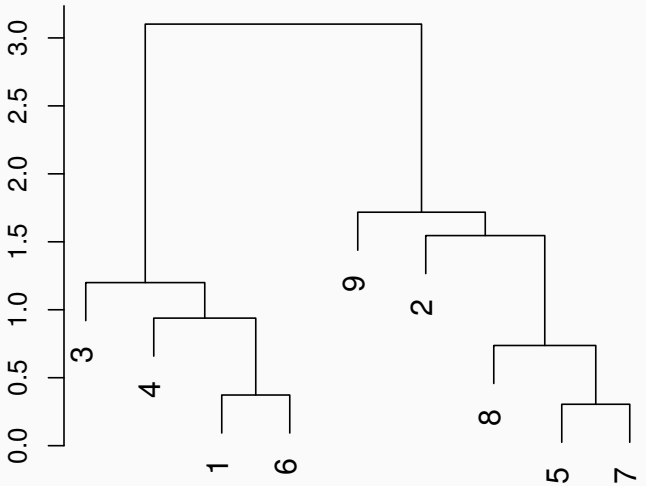
Six times K-means run with different random initial assignments.

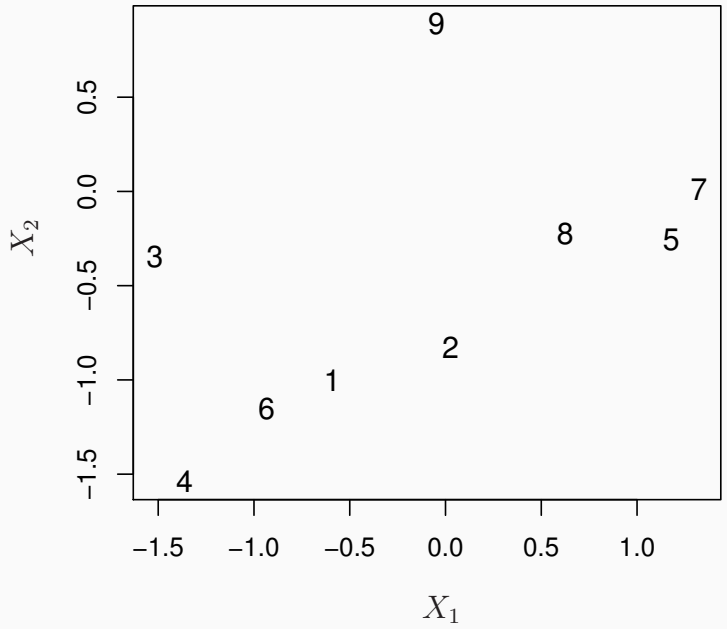
Hierarchical Clustering

- K-means requires specifying the number of clusters K in advance — a limitation.
- *Hierarchical clustering* does not require a pre-specified K.
- We describe the **agglomerative** or bottom-up approach, in which a **dendrogram** is built by successively merging observations or clusters.
- The process starts with each observation as its own cluster and merges upward to form larger clusters.

Hierarchical Clustering: the idea

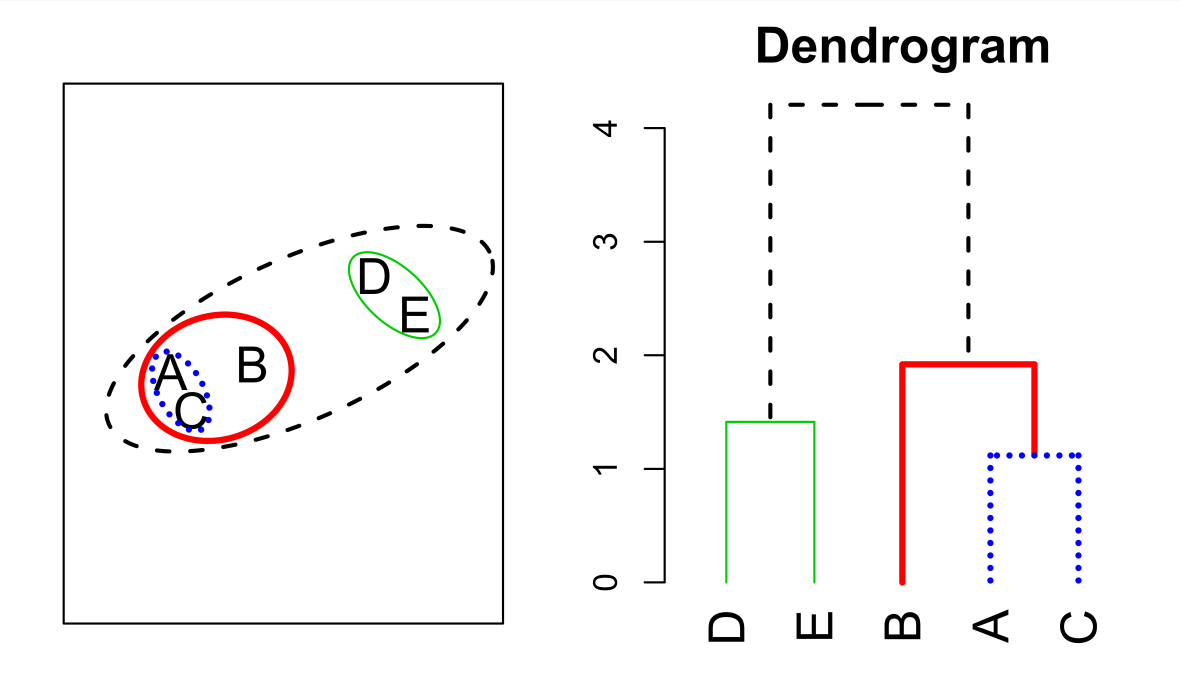
- Builds a hierarchy in a *bottom-up* fashion.
- Each observation starts as its own cluster.
- At each step, the two most similar clusters are merged.
- The process continues until all observations belong to one cluster.



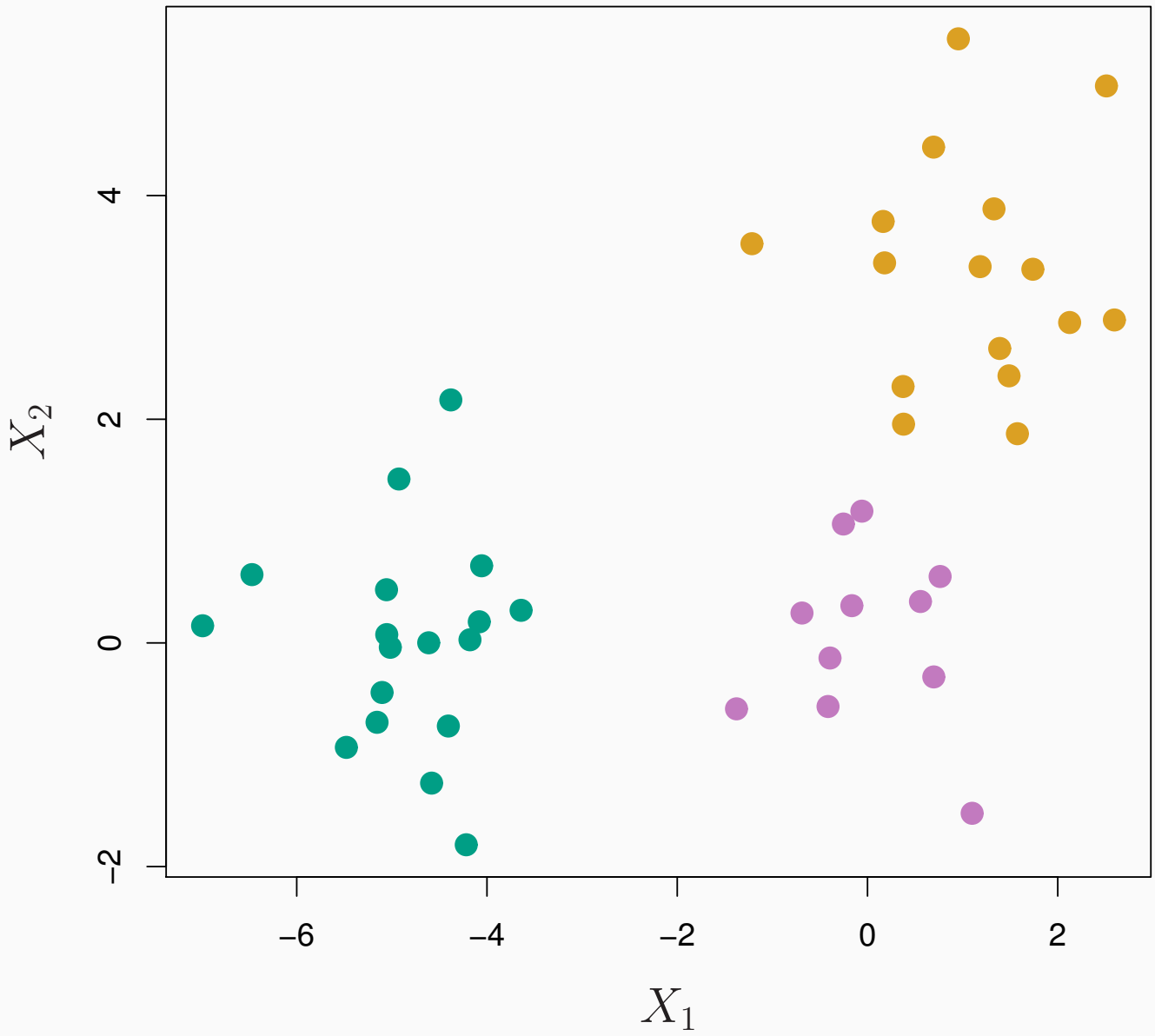


Hierarchical Clustering Algorithm

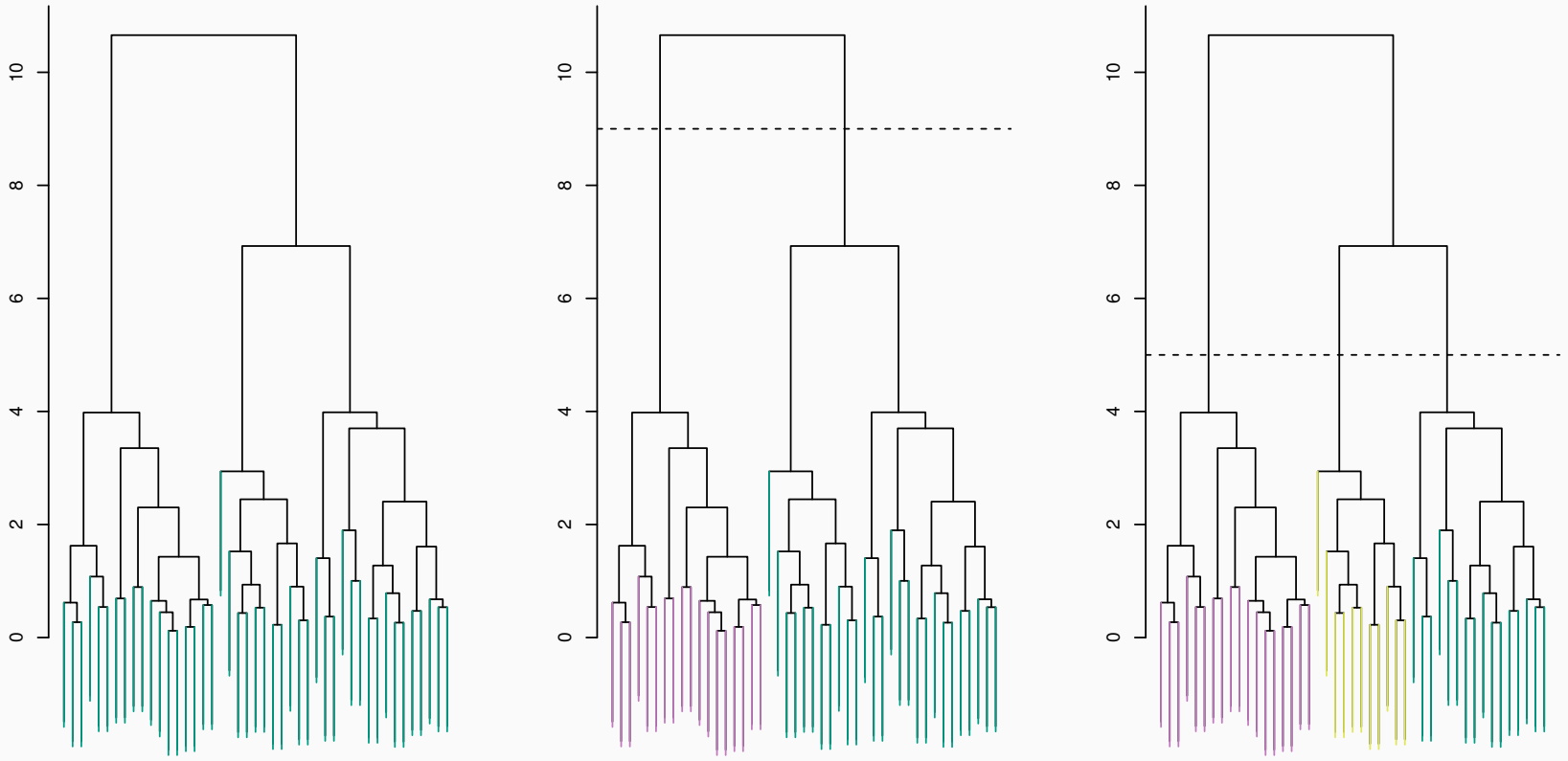
- Start with each point in its own cluster.
- Identify the two clusters that are closest and merge them.
- Repeat this process until all points are merged into a single cluster.
- The result is represented by a *dendrogram*.



An Example



Application of Hierarchical Clustering



Hierarchical clustering applied to the previous dataset (complete linkage, Euclidean distance). Different cut heights yield different numbers of clusters (2 or 3).

Details of Previous Figure

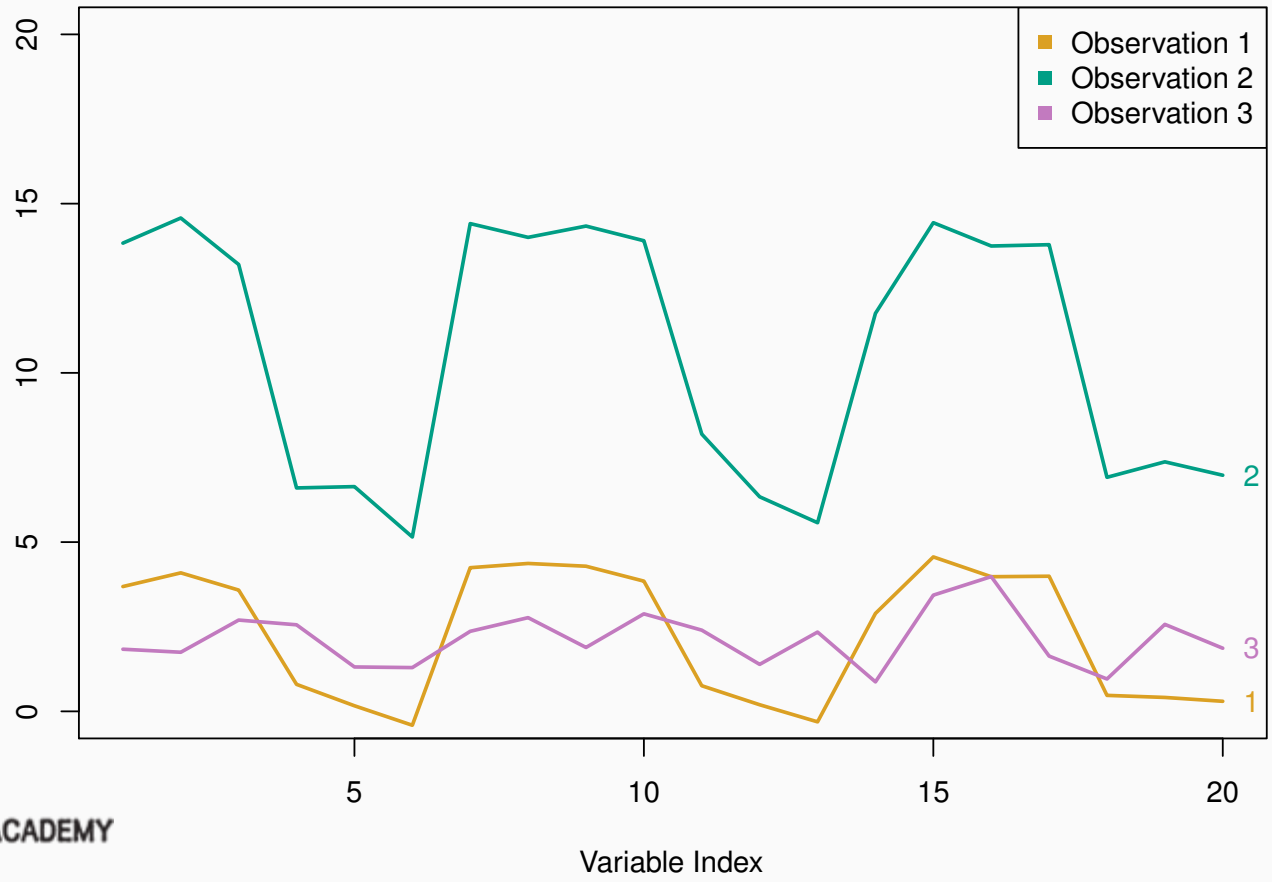
- **Left:** Complete-linkage dendrogram of the data.
- **Center:** Cut at height 9 $\rightarrow$ two clusters (different colors).
- **Right:** Cut at height 5 $\rightarrow$ three clusters.
- Colors are for display only — not used in the clustering itself.

Types of Linkage

Linkage	Description
Complete	Maximal inter-cluster dissimilarity (largest pairwise distance).
Single	Minimal inter-cluster dissimilarity (smallest pairwise distance).
Average	Mean inter-cluster dissimilarity (average pairwise distance).
Centroid	Distance between cluster centroids (can cause inversions).

Choice of Dissimilarity Measure

- The default measure is usually *Euclidean distance*.
- Alternatively, use a *correlation-based distance*, treating two observations as similar if their features are highly correlated.
- Here, correlation is computed between **observations**, not variables.

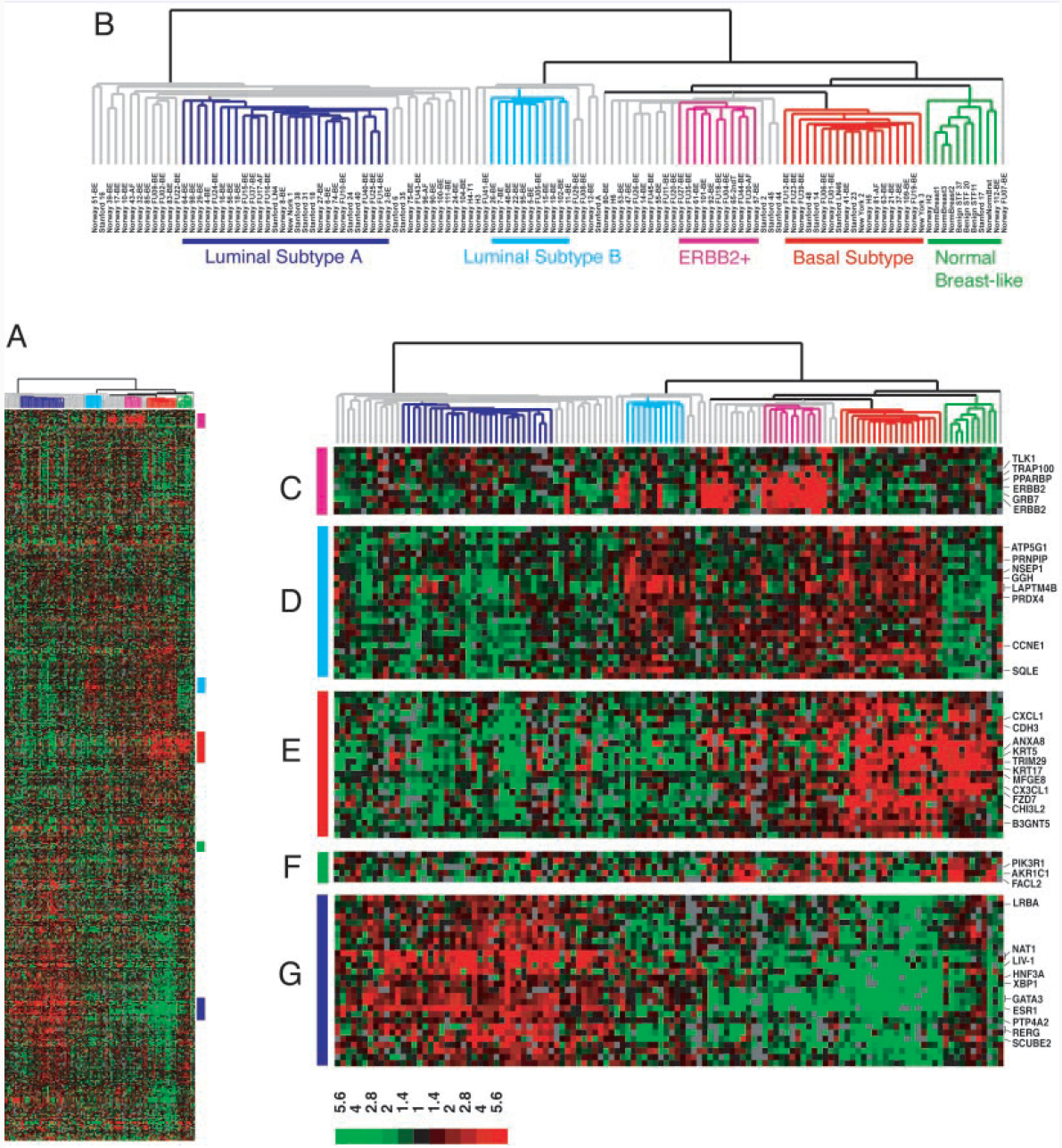


Practical Issues

- **Scaling matters!** Should features be standardized (mean zero, SD one)?
- In hierarchical clustering:
  - What dissimilarity measure to use?
  - What linkage method to choose?
  - How many clusters to retain?
- In both K-means and hierarchical clustering, the number of clusters is often subjective.
- See Elements of Statistical Learning, Chapter 13, for further discussion.
- Which features should drive the clustering?

Example: Breast Cancer Microarray Study

- Gene expression for ∼8,000 genes from 88 patients.
- Used **average linkage** and a **correlation-based distance**.
- Clustered samples using 500 *intrinsic genes*—those with smallest within/between variation.


B
Luminal Subtype A
Luminal Subtype B
ERBB2+
Basal Subtype
Normal Breast-like
A
C
TLK1
TRAP100
PPARBP
ERBB2
GRB7
ERBB2
D
ATP5G1
PRNPIP
NSEP1
GGH
LAPTM4B
PRDX4
CCNE1
SQLE
E
CXCL1
CDH3
ANXA8
KRT5
TRIM29
KRT17
MFGE8
CX3CL1
FZD7
CHI3L2
B3GNT5
F
PIK3R1
AKR1C1
FACL2
G
LRBA
NAT1
LIV-1
HNF3A
XBP1
GATA3
ESR1
PTP4A2
RERG
SCUBE2
5.6
4
2.8
2.4
1
1.4
2.8
4
5.6

Conclusions

- *Unsupervised learning* helps uncover structure in unlabeled data and often serves as a pre-processor for supervised learning.
- It is more challenging than supervised learning—no response variable or clear objective function.
- An active field with many recent tools:
    - *Self-organizing maps*, *Independent Component Analysis*, and *Spectral clustering*.
- See The Elements of Statistical Learning, Chapter 14.