# MY474: Applied Machine Learning for Social Science

## Lecture 9: Unsupervised Learning

Blake Miller

04 December 2019

# Agenda

1. Unsupervised Learning
2. Dimension Reduction
3. Clustering

# Unsupervised Learning

# Unsupervised Learning

**Unsupervised vs Supervised Learning:**

- Most of this course focused 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 or outcome variable $Y$. The goal is then to predict $Y$ using $X_1, X_2, \ldots, X_p$.
- The labels $Y$ supervise the process of learning
- Here we instead focus on **unsupervised learning**, we where observe only the features $X_1, X_2, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable $Y$.
- Find patterns and similarities between observations using features in $X$

# Why Unsupervised Learning?

- It's cheaper - labeling data can be costly
- Unsupervised learning can guide supervision (e.g. groups found through clustering can inform labeling)
- Feature learning: use unsupervised learning to find features that will then be useful for categorization (e.g. Word2Vec)
- Exploration: exploring structure or patterns in the data in the early stages of an investigation can be help guide research.

# The Goals of Unsupervised Learning

- Find pattern and structure in the data:
  - Can we discover subgroups among the variables or among the observations?
  - Can we represent the data in a more concise or useful way?
- We discuss two methods:
  - **principal components analysis**, a tool used for data visualization or data pre-processing before supervised techniques are applied, and
  - **clustering**, a broad class of methods for discovering unknown subgroups in data.
- We have discussed other examples before in class:
  - **word2vec**, a model that is used for pre-processing text data, and
  - **topic models**, for discovering underlying topics in a corpus

# The Challenge of Unsupervised Learning

- Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response.
- But techniques for unsupervised learning are of growing importance in a number of fields:
  - subgroups of breast cancer patients grouped by their gene expression measurements,
  - groups of shoppers characterized by their browsing and purchase histories,
  - movies grouped by the ratings assigned by movie viewers.

# Dimension Reduction

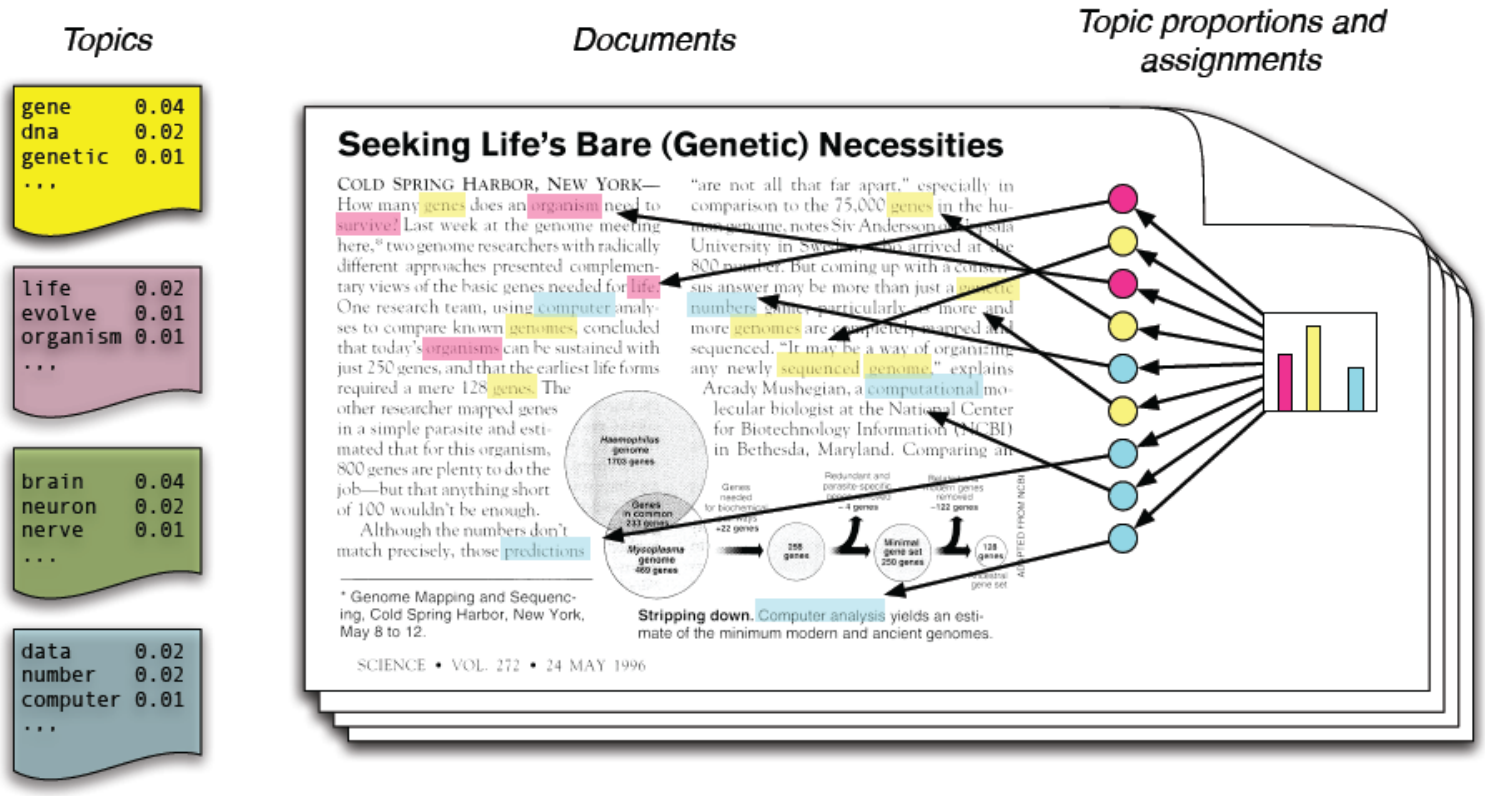
# Problems with High Dimensional Data

- Correlated features
  - Ideally we would like our features to each convey independent information
  - In reality we often have redundant or superfluous features.
  - We sometimes run into a scenario where adding more features reduces classifier performance due to added noise.
- Computational complexity
  - We are often limited by the computational complexity of classifiers.
  - With large $p$ data, we may have very long training times.

# Dimension Reduction

- To deal with the problem of excessive dimensionality we can combining features.
- One way to do this is through linear combinations: project the high-dimensional data onto a lower dimensional space.
- PCA finds a projection that best represents the data in a least-squares sense
- There are many other approaches to dimention reduction (FYI):
  - Factor analysis
  - Multi-dimensional scaling (MDS)
  - Singular value decomposition (SVD)
  - Non-negative matrix factorization (NMF)
  - T-distributed Stochastic Neighbor Embedding (t-SNE)

# Example: Latent Dirichlet Allocation Topic Models

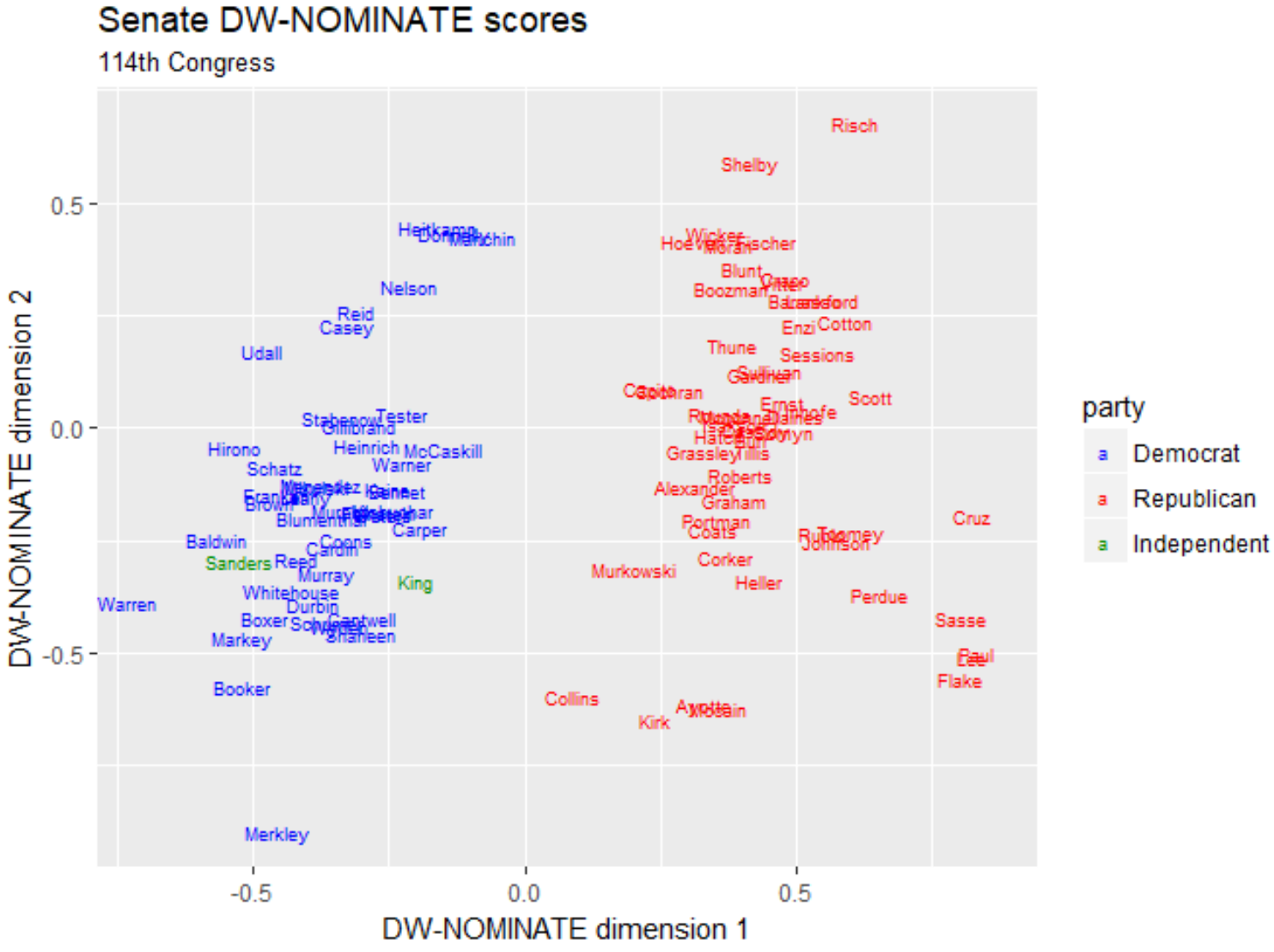



LDA models documents as a mixture topics and assigns each word to one of the document's topics.

# Example: NOMINATE

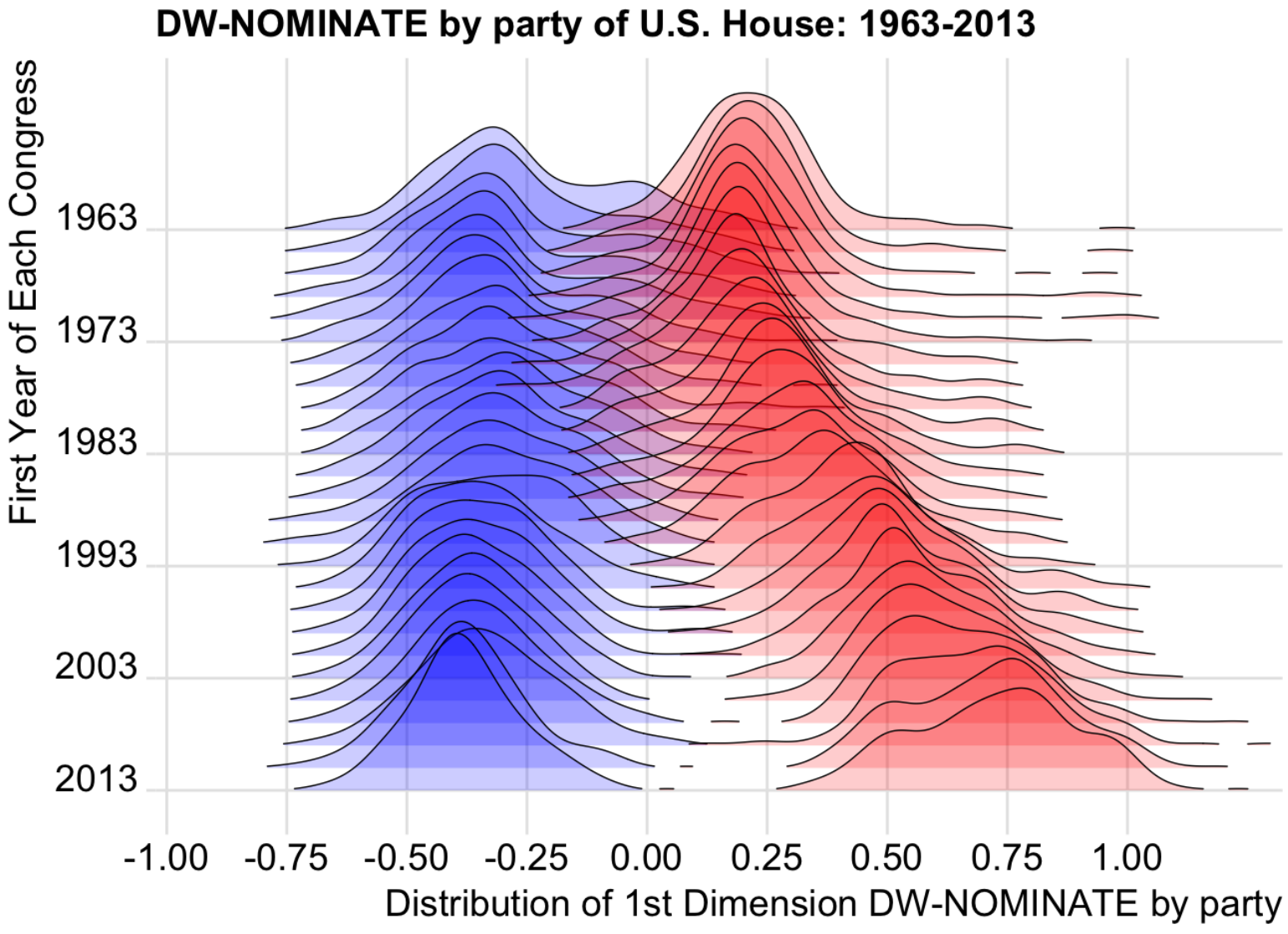
- Stands for "Nominal Three-Step Estimation" and is an application of multidimensional scaling by Poole & Rosenthal (1983)
- Goal: Analyzing legislative roll call votes through **dimension reduction**
- Representing alternative vote choices by projecting role call vote data onto two-dimensional Euclidean space.
- The first dimension ($x$-axis) is the liberal-conservative spectrum on economic affairs
- The second dimension ($y$-axis) represents other cross-cutting cleavages (slavery, civil rights, immigration reform, gay rights, etc.)

# Example: NOMINATE




DW-NOMINATE scores for the 114th Congress computed by Poole and Rosenthal (2017).

# Example: NOMINATE




Source: https://rpubs.com/ianrmcdonald/293304

# Principal Components Analysis

- The main objective: **reduce dimensionality** of the data set.
- This is done by finding a sequence of **linear combinations** of the variables that have **maximal variance**, and are **mutually uncorrelated**.
- Replaces the original $p$ variables with $k < p$ linear combinations of the original variables that are a "good representation" of the data (a linear dimension reduction method)
- Belongs to the class of projection methods Useful for
  - **visualization** (project to 2-d or 3-d)
  - as a **pre-processing** step for other methods that do not deal well with an excessive number of variables (**principle component regression (PCR)**, classification based on principle components)

# Principal Components Analysis: details

The **first principal component** of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

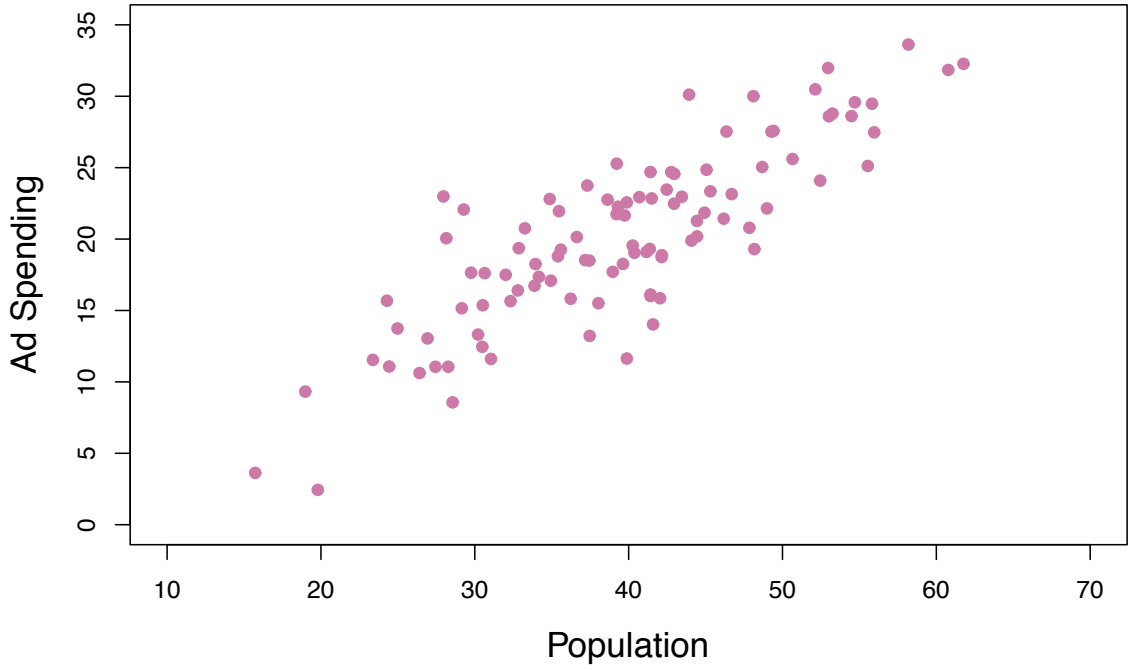
$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

that has the largest variance. By normalized, we mean that

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the **loadings** of the first principal component; together, the loadings make up the principal component loading vector,
$$\phi_1 = (\phi_{11} \phi_{21} \ldots \phi_{p1})^T.$$
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

# PCA: example




The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles.

**Question:** What is a good 1-dim projection of the data?

# PCA: example

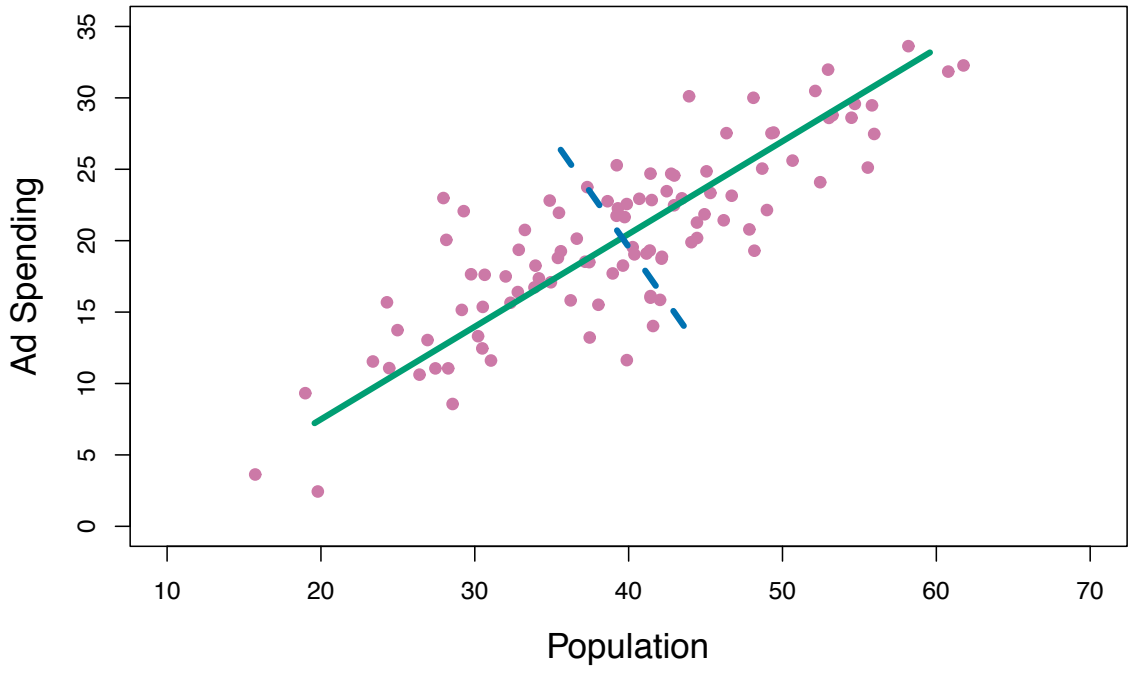
- Could use one of the variables (e.g. ad in the example). But what if there are many thousands of variables?
- Better idea: use a linear combination of the variables; i.e. a weighted average of the variables. In the example,

$$Z_1 = \phi_1 X_1 + \phi_2 X_2$$

.

- **Main issue:** what is a good choice for the weights $\phi_1$ and $\phi_2$? Need a criterion for choosing the weights.

# PCA: example




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

# Computation of Principal Components

- Suppose we have a $n \times p$ data set $\mathbf{X}$. Since we are only interested in variance, we assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero (that is, the column means of $\mathbf{X}$ are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \cdots + \phi_{p1} x_{ip} \text{ for } i = 1, \ldots, n$$

  that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- Since each of the $x_{ij}$ has mean zero, then so does $z_{i1}$ (for any values of $\phi_{j1}$). Hence the sample variance of the $z_{i1}$ can be written as $\frac{1}{n} \sum_{i=1}^{n} z_{i1}^2$ .

# Computation: continued

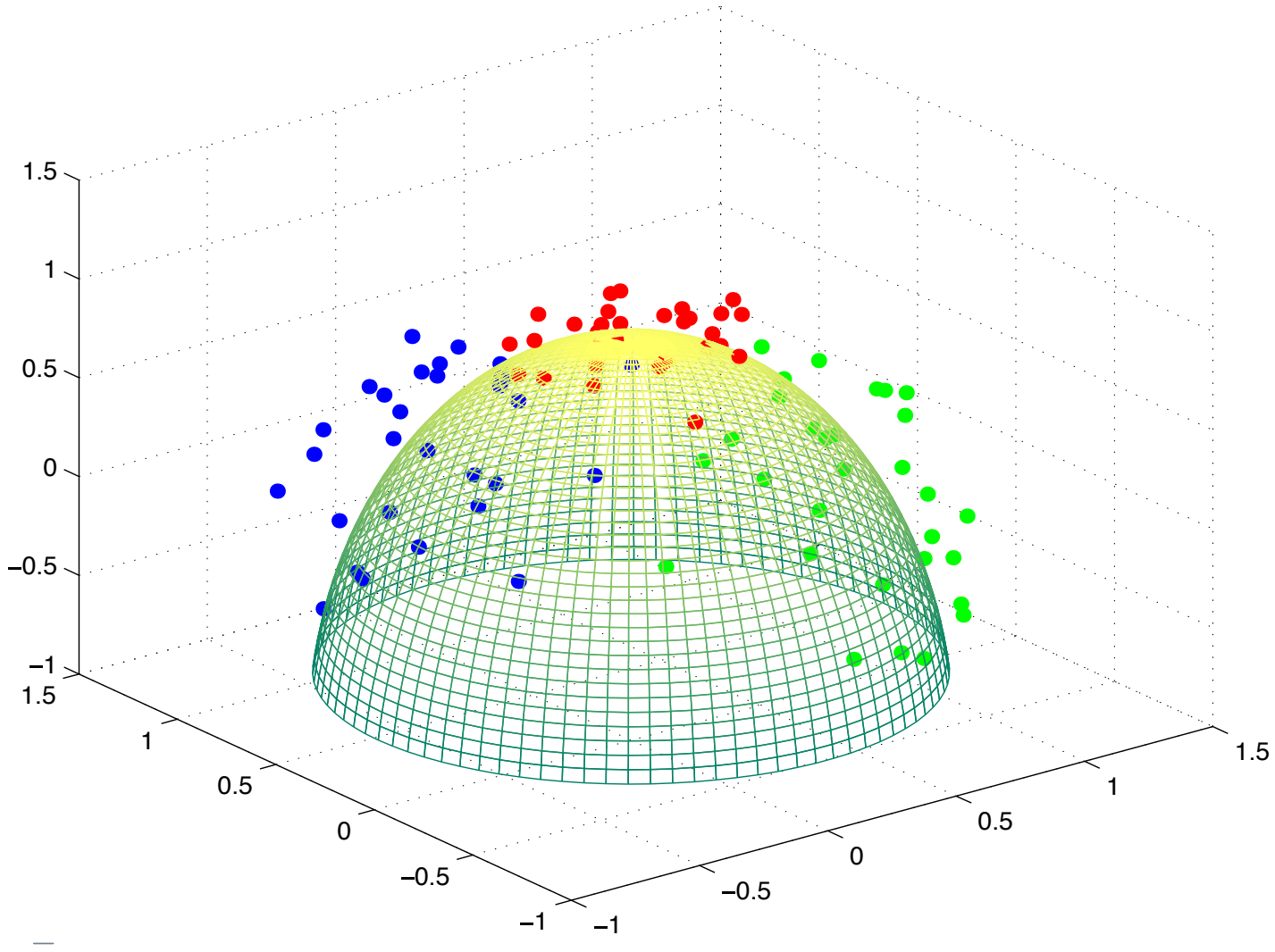
- Plugging in $z_{i1}$ from the previous slide, the first principal component direction/loading vector solves the optimization problem

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

# Geometry of PCA

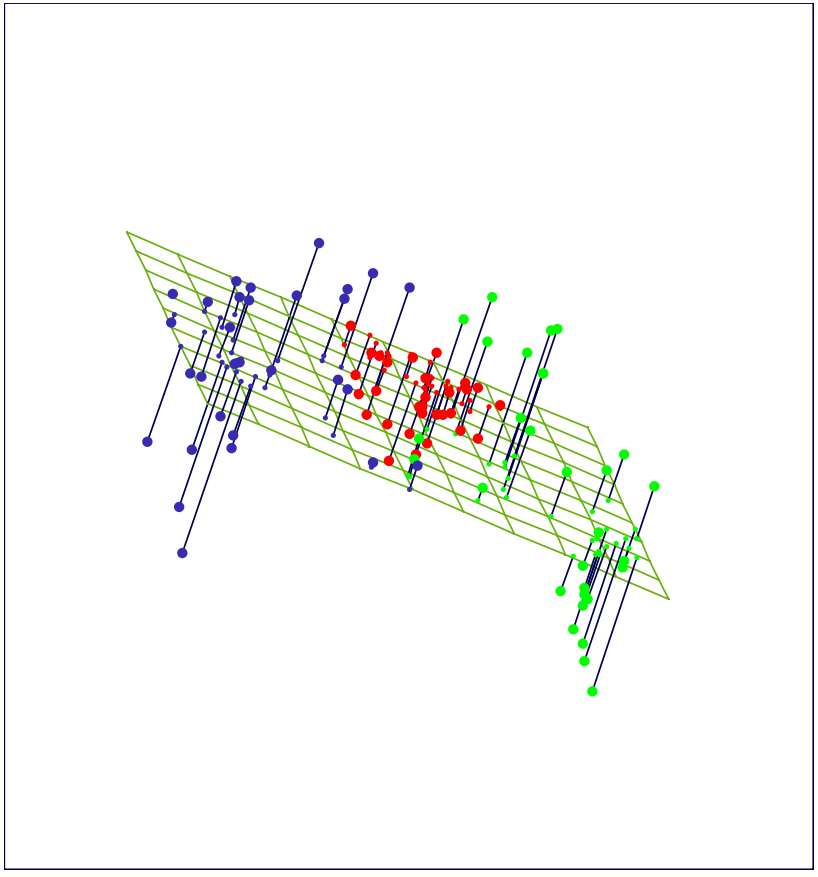
- The loading vectors $\phi_i$ capture "interesting" directions in the data
- According to the PCA, the criterion for "interesting" is the one that captures the most variance in the data.
- The loading vector $\phi_1$ defines a direction in feature space along which the data vary the most. Subsequent loading vectors are orthogonal to this vector.
- If we project the $n$ data points $x_1, \ldots, x_n$ onto this direction, the projected values are the principal component scores $z_1 1, \ldots, z_n 1$ themselves.
- This problem can be solved via a **singular-value decomposition** of the matrix $\mathbf{X}$, a standard technique in linear algebra.
- We refer to $Z_1$ as the first principal component, with realized values $z_{11}, \ldots, z_{n1}$

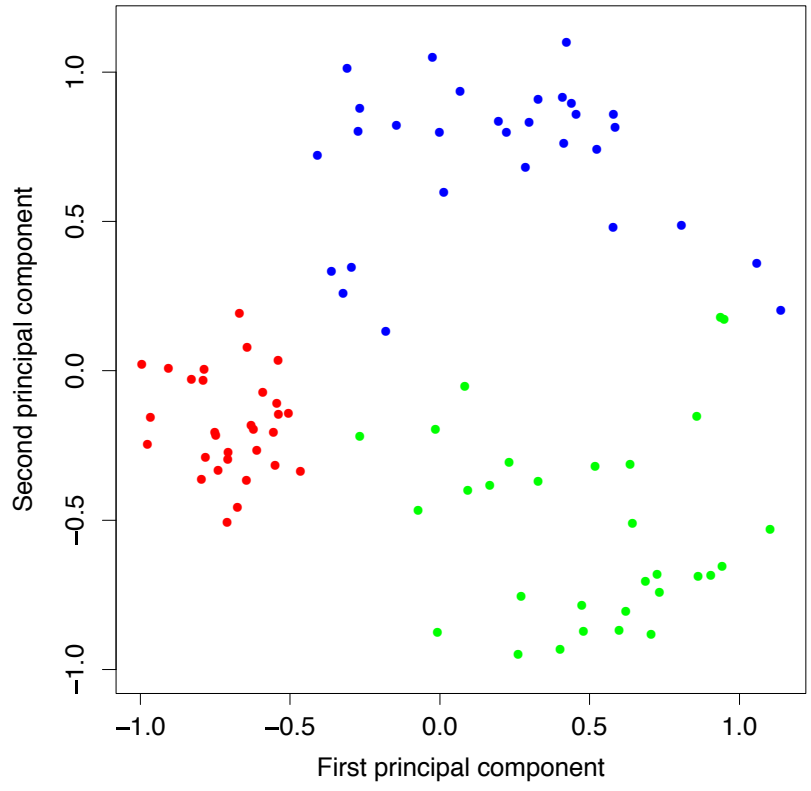
# Geometry of PCA




Simulated data in three classes, near the surface of a half–sphere.

# Geometry of PCA






The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by the first two principal components of the data (source: ESL p. 536)

# Geometry of PCA

- The first principal component loading vector has a very special property: it defines the line in p-dimensional space that is closest to the $n$ observations (using average squared Euclidean distance as a measure of closeness)
- The notion of principal components as the dimensions that are closest to the $n$ observations extends beyond just the first principal component.
- For instance, the first two principal components of a data set span the plane that is closest to the $n$ observations, in terms of average squared Euclidean distance.

# Further principal components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are uncorrelated with $Z_1$.
- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{p2} x_{ip},$$

where $\phi_2$ is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$.

# Further principal components: continued

- It turns out that constraining $Z_2$ to be uncorrelated with $Z_1$ is equivalent to constraining the direction $\phi_2$ to be orthogonal (perpendicular) to the direction $\phi_1$. And so on.
- The principal component directions $\phi_1, \phi_2, \phi_3, \ldots$ are the ordered sequence of right singular vectors of the matrix $\mathbf{X}$, and the variances of the components are $\frac{1}{n}$ times the squares of the singular values. There are at most $min(n-1, p)$ principal components.

# Properties of PCs

- PCs are centered: $E(Z_i) = 0$
- PCs are uncorrelated: $Cov(Z) = \Lambda$, $Var(Z_i) = \lambda_i$.
- If $X$ is multivariate normal, so is $Z$, and PCs are independent.

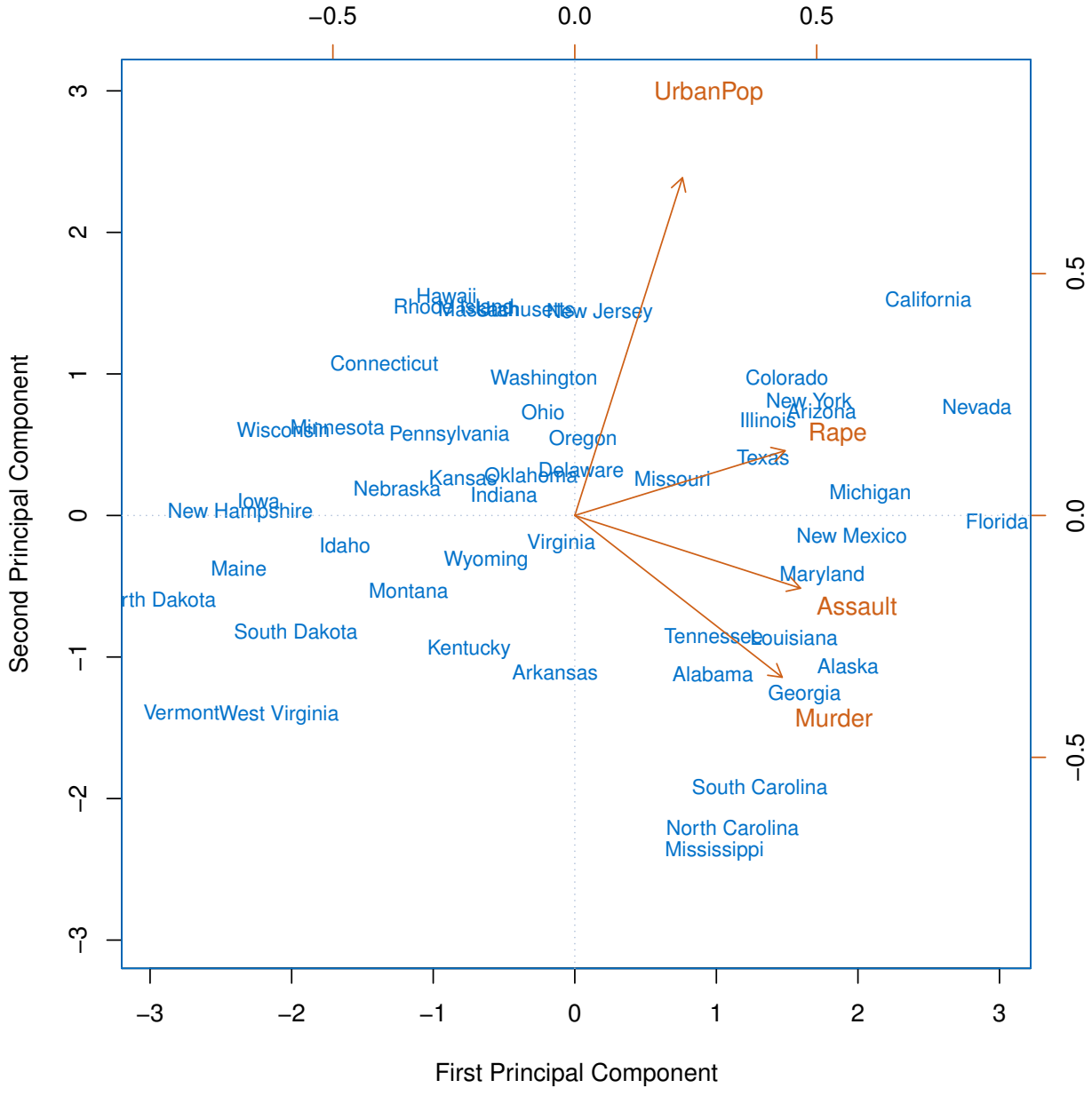
# PCA Terminology

- Let $X$ now be the $n \times p$ data matrix (centered)
- The vectors $\phi_i$ are called **PC directions** or **loading vectors**.
- Vectors $Z_i = X\phi_i$ are called the principal components of $X$ and are projections of the data onto the PC directions
- Components of $X\phi_i$ are also called **scores**
- The coordinates $\phi_{ij}$ are called **(factor) loadings**

# Illustration

- `USAarrests` data: For each of the fifty states in the United States, the data set contains the number of arrests per 100,000 residents for each of three crimes: `Assault`, `Murder`, and `Rape`. We also record `UrbanPop` (the percent of the population in each state living in urban areas).
- The principal component score vectors have length $n = 50$, and the principal component loading vectors have length $p = 4$.
- PCA was performed after standardizing each variable to have mean zero and standard deviation one.

# USAarrests data: PCA plot



# Figure details

The first two principal components for the USArrests data.

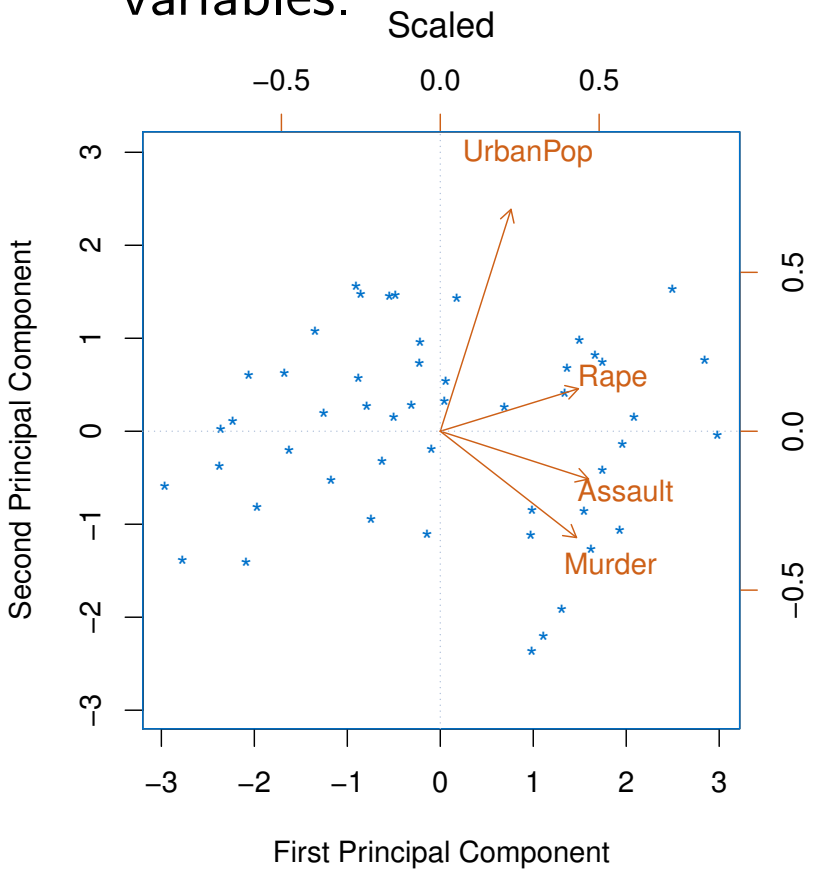
- The blue state names represent the scores for the first two principal components.
- The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for `Rape` on the first component is 0.54, and its loading on the second principal component 0.17 [the word `Rape` is centered at the point (0.54, 0.17)].
- This figure is known as a **biplot**, because it displays both the principal component scores and the principal component loadings.

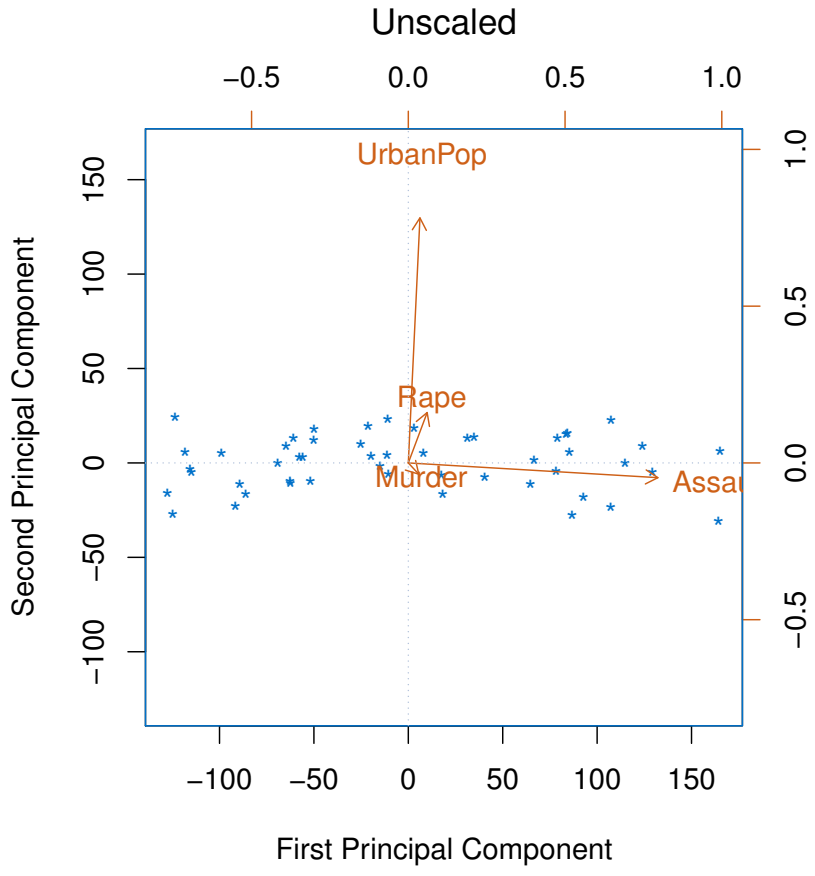
# PCA loadings

| | PC_1 | PC_2 |
|---|---|---|
| Murder | 0.5358995 | -0.4181809 |
| Assault | 0.5831836 | -0.1879856 |
| UrbanPop | 0.2781909 | 0.8728062 |
| Rape | 0.5434321 | 0.1673186 |

# Scaling of the variables matters

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.



# Proportion Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.
- The total variance present in a data set (assuming that the variables have been centered to have mean zero) is defined as

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

and the variance explained by the $m$th principal component is

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

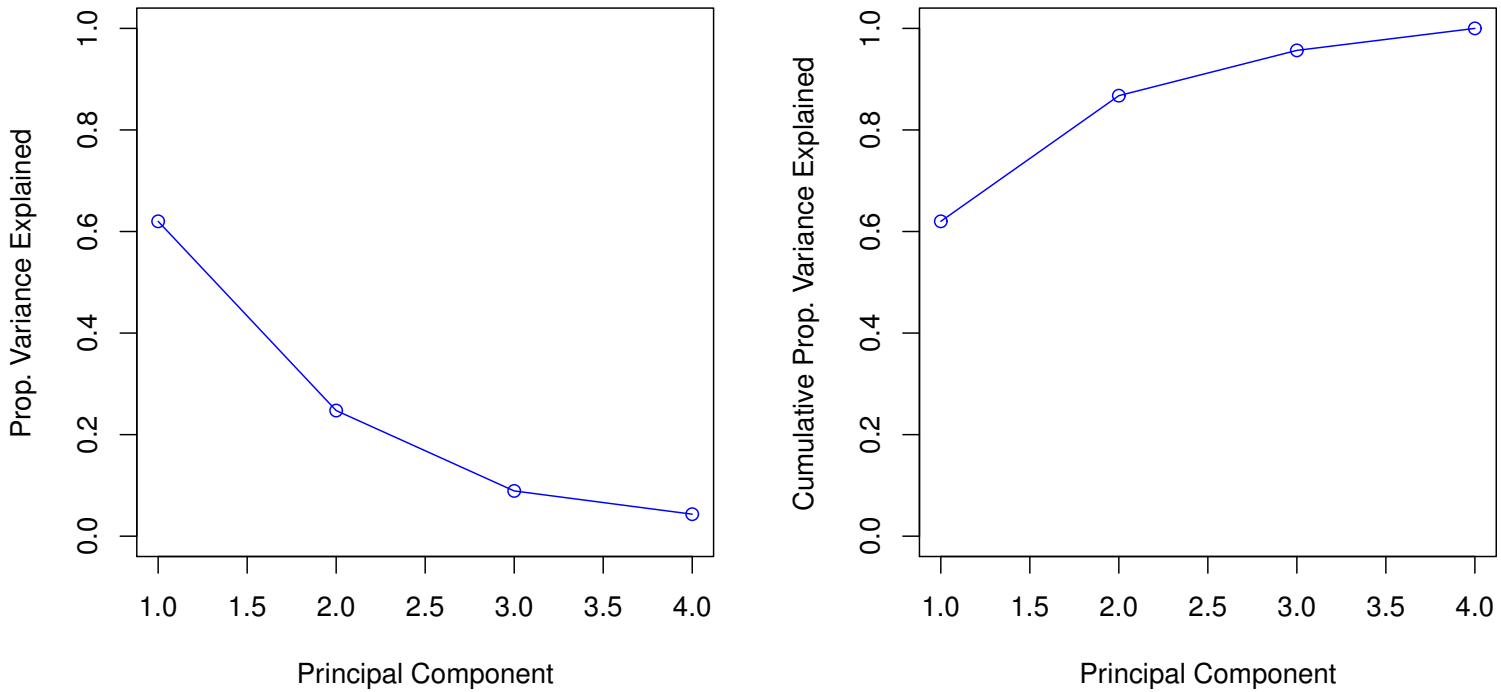
- It can be shown that $\sum_{j=1}^{p} Var(X_j) = \sum_{m=1}^{M} Var(Z_m)$, with $M = min(n-1, p)$.

# Proportion Variance Explained: continued

- Therefore, the PVE of the $m$th principal component is given by the positive quantity between 0 and 1

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

- The PVEs sum to one. We sometimes display the cumulative PVEs.



# How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.

# How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
- **Why not?**

# How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
- **Why not?**
- When could we use cross-validation to select the number of components?

# How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
- **Why not?**
- When could we use cross-validation to select the number of components?
- the **"scree plot"** on the previous slide can be used as a guide: we look for an **"elbow"**.

# Clustering

# Clustering

- **Task:** Automatically identify groups of similar observations within data.
- **Objective:** find groups that are:
  - **Homogeneous**: observations within a group are very similar to each other
  - **Distinct**: observations within a group are very different from those outside the group
- When we don't have a clear idea of what to look for in data, clustering can aid in **discovery** and **description**
- Clustering is not ideal for **inference** because the clusters discovered must be interpreted *a posteriori*
- The task of clustering is **unsupervised** meaning that these groups are **discovered** and not specified *a priori*

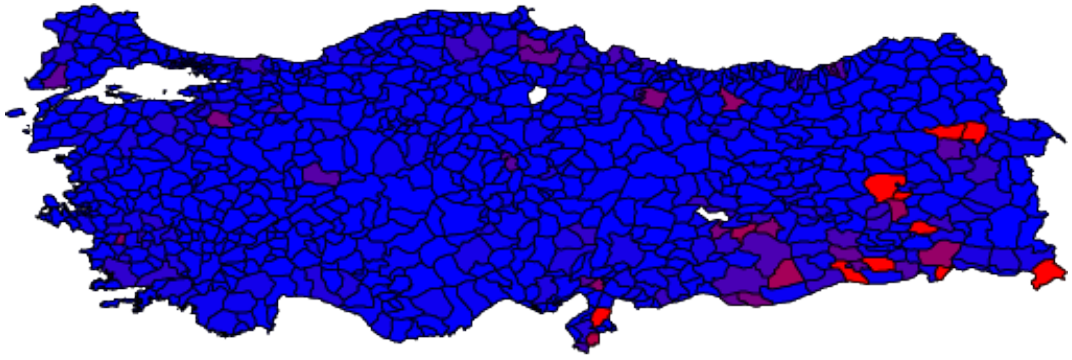
# Applications of Clustering

- **Market segmentation:** identify demographic or behavioral groups for targeted advertising or marketing campaigns.
- **Fraud detection:** identify election fraud, anomolous network activity, etc.
- **Discovering heterogeneity in treatment effects:** are certain groups more likely to respond to a drug?
- **"Distance reading" for digital humanities:** which $n$ documents will give the most comprehensive overview of a corpus?

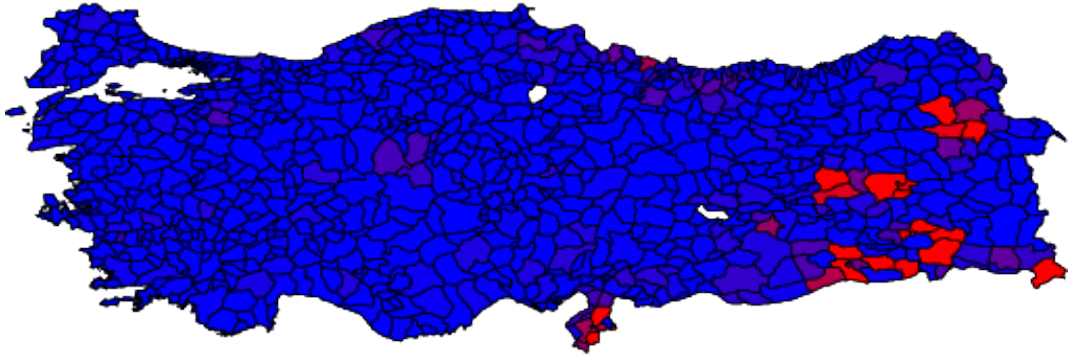
# Example: Election Fraud in Turkey

- Using clustering, political scientists at Michigan have developed a tool to identify election fraud (see more here)
- Election fraud models such as Klimek et al. (2012) or Mebane (2017) specify two mechanisms of fraud benefitting a winning party:
  1. Inventing votes from genuine nonvoters
  2. Stealing votes from the nonwinning set of parties
- If frauds occur, turnout and vote proportion distributions form two or three clusters instead of a single cluster. Clusters are defined as follows:
  - "Extreme fraud" cluster: "election data have turnout near 100 percent with nearly all votes going to the winner."
  - "Incremental fraud" cluster: "a substantial number but not almost all votes are reallocated to the winner."
  - "No fraud" cluster: No anomalies in turnout and proportion of votes to the winner

# Example: Election Fraud in Turkey



(a) June



(b) November

Fraudulent Vote Proportions by Town, Turkey 2015. Turkey's Justice and Development Party (AKP) lost control of the legislature after the June election but regained control in November. Towns where signs of frauds occur significantly more often than average are red and significantly less often than average are blue.

# Discussion

How is the task of clustering **similar**/**different** from the task of classification?

# Discussion

- Can clusters have meaningful **class labels**?

# Discussion

- Can clusters have meaningful **class labels**?
- Clustering will give you $k$ clusters, but it is up to the researcher to interpret what these clusters mean. This can be problematic. **Why?**

# PCA vs Clustering

- PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
- Clustering looks for homogeneous subgroups among the observations.
- We can interpret the meaning of components *a posteriori*; recall that the first PC in the crime data had high loadings for different measures of crime and the second PC had high loadings for population.
- PCA results in **loadings** which give an idea of how much a certain observation "loads" onto a component
- In other words: there are not distinct observations belonging to each component as there are with clusters.

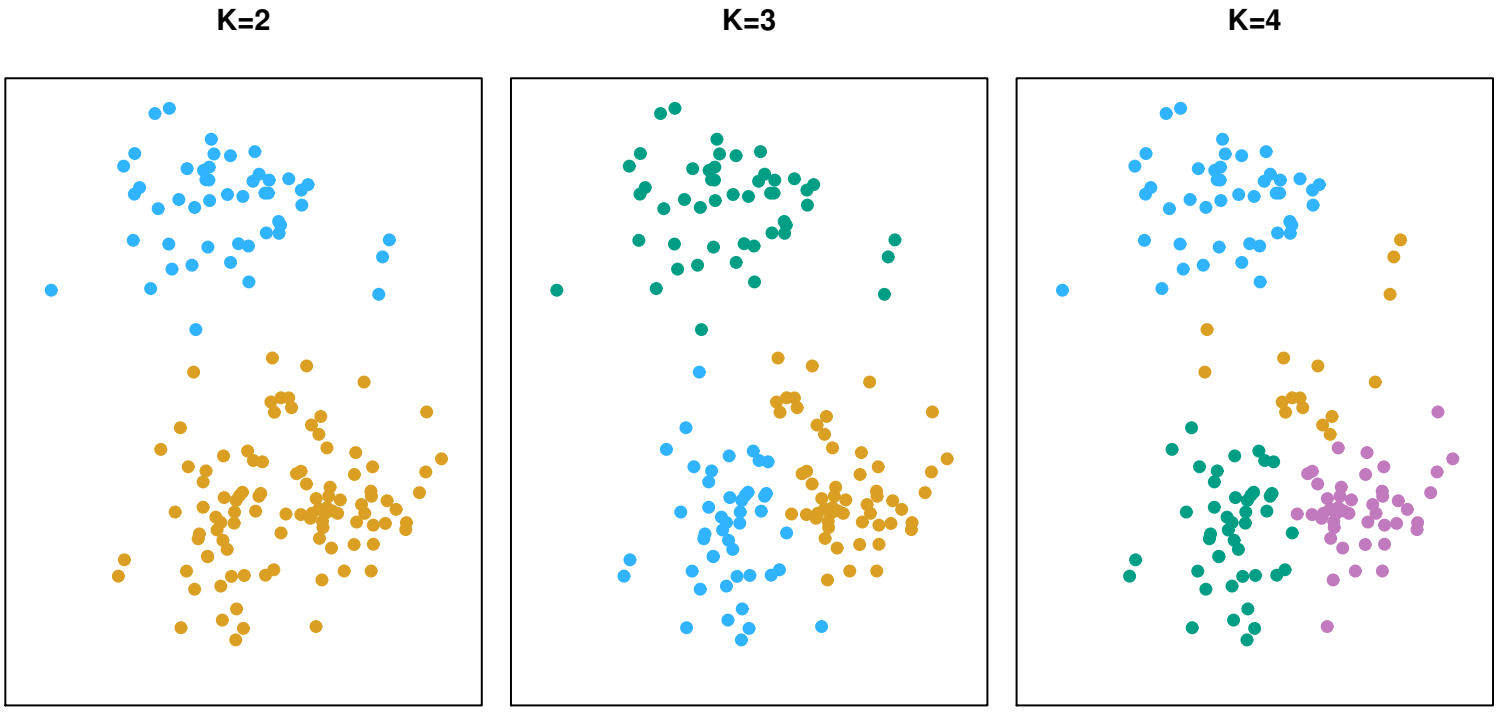
# Clustering for Market Segmentation

- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform **market segmentation** by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

# Two clustering methods

- In **K-means clustering**, we seek to partition the observations into a pre-specified number of clusters.
- 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




A simulated data set with 150 observations in 2-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.

# Details of K-means clustering

Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$. Inotherwords,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.

For instance, if the $i$th observation is in the $k$th cluster, then $i \in C_k$.

# Details of K-means clustering: continued

- 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 $C_k$ is a measure $WCV(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Hence we want to solve the problem

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

- In words, this formula says that we want to partition the observations into $K$ clusters such that the total within-cluster variation, summed over all $K$ clusters, is as small as possible.

# How to define within-cluster variation?

- Typically we use the average Euclidean distance between all pairwise combination of observations in a cluster:

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

where $|C_k|$ denotes the number of observations in the $k$th cluster. - The optimization problem that defines K-means clustering is

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

# K-Means Clustering Algorithm

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:
   - For each of the $K$ clusters, compute the cluster **centroid**. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.
   - Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

# Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective function at each step. **Why?**

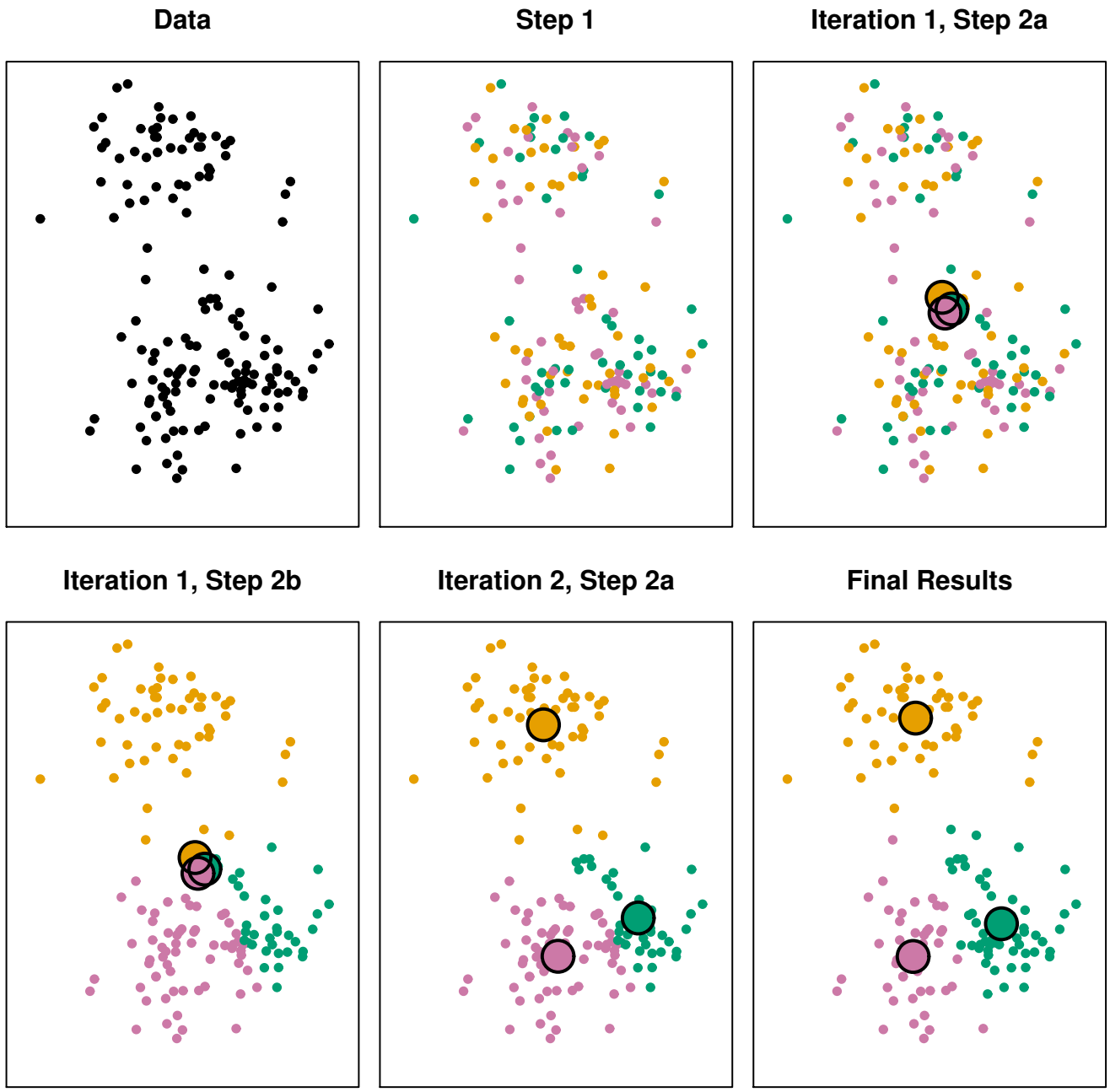
# Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective function at each step. **Why?** Note that

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

where $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ is the mean for feature $j$ in $C_k$. >- however it is not guaranteed to give the global minimum. **Why not?**

# Example


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

# Details of Previous Figure

The progress of the K-means algorithm with K=3.

- Top left: The observations are shown.
- Top center: In Step 1 of the algorithm, each observation is randomly assigned to a cluster.
- Top right: In Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.
- Bottom left: In Step 2(b), each observation is assigned to the nearest centroid.
- Bottom center: Step 2(a) is once again performed, leading to new cluster centroids.
- Bottom right: The results obtained after 10 iterations. 34/52

# Example: different starting values

# Details of Previous Figure

- K-means clustering performed six times on the data from previous figure 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 (4).
- 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

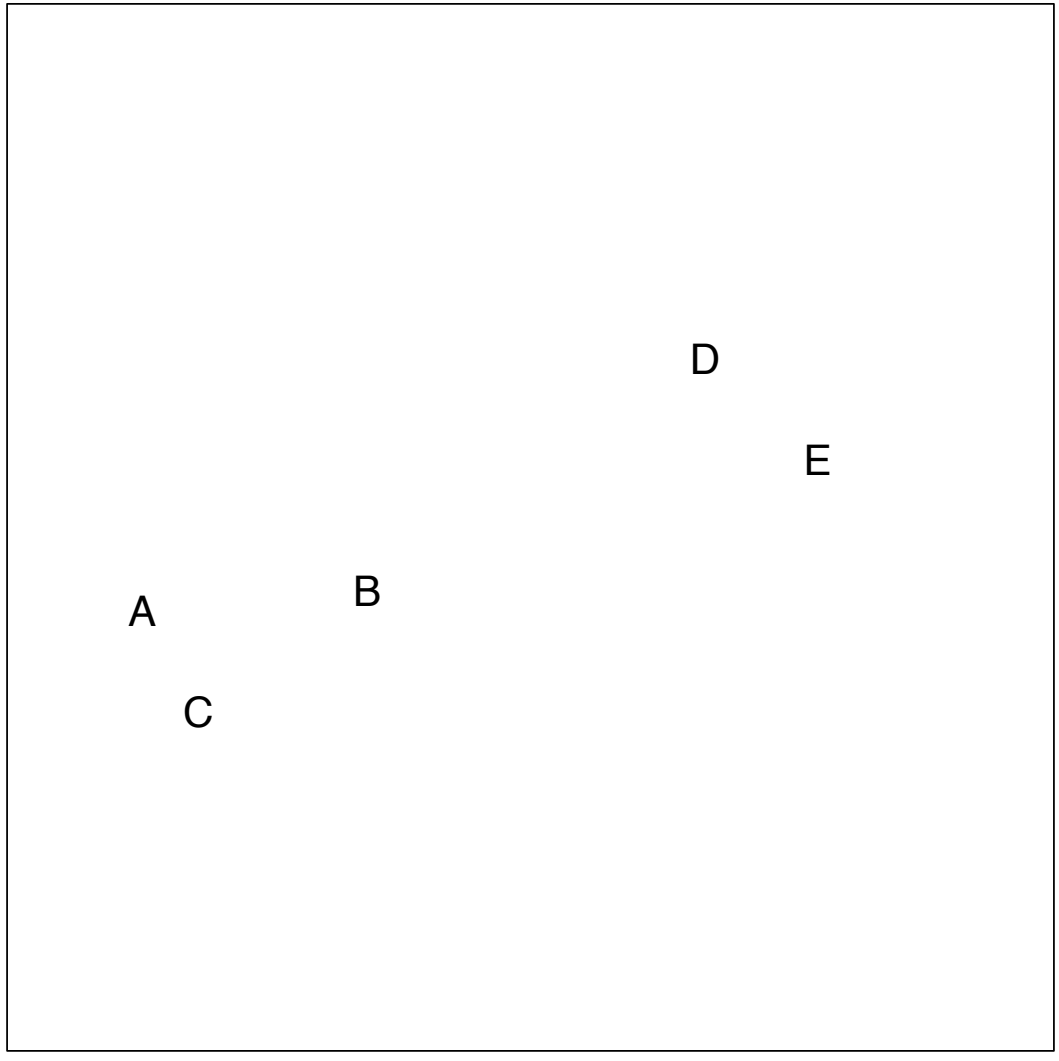
# K-Means Applied to Image Compression

Sir Ronald A. Fisher (1890 - 1962) was one of the founders of modern day statistics. Left: 1024×1024 grayscale image at 8 bits per pixel, Center: 2 × 2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. Right: 2 block VQ, using only four code vectors, with a compression rate of 0.50 bits/pixel (source: ESL p. 514)

# Hierarchical Clustering

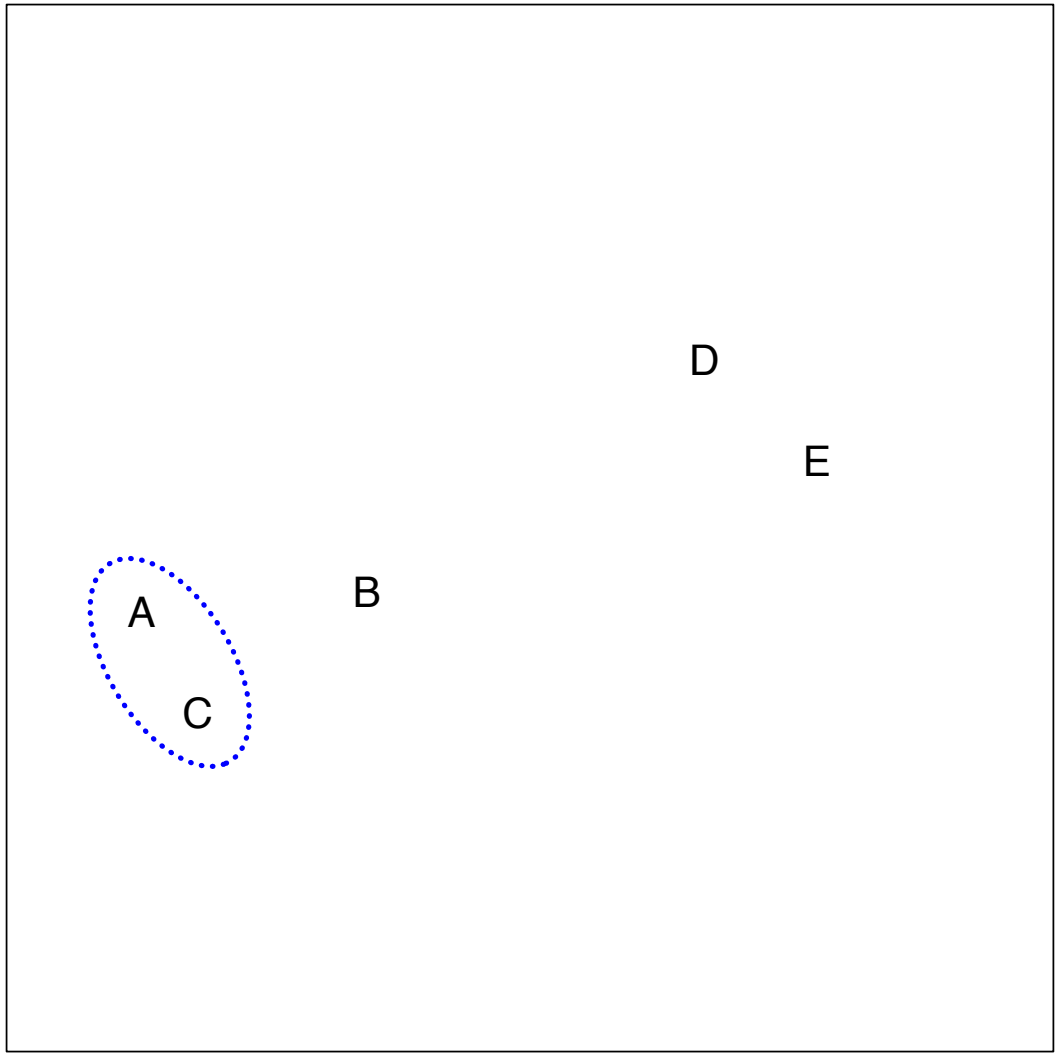
- K-means clustering requires us to pre-specify the number of clusters K. This can be a disadvantage (later we discuss strategies for choosing K)
- Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.
- In this section, we describe bottom-up or agglomerative clustering. This is the most common type of hierarchical clustering, and refers to the fact that a **dendrogram** is built starting from the leaves and combining clusters up to the trunk.

# Hierarchical Clustering: the idea



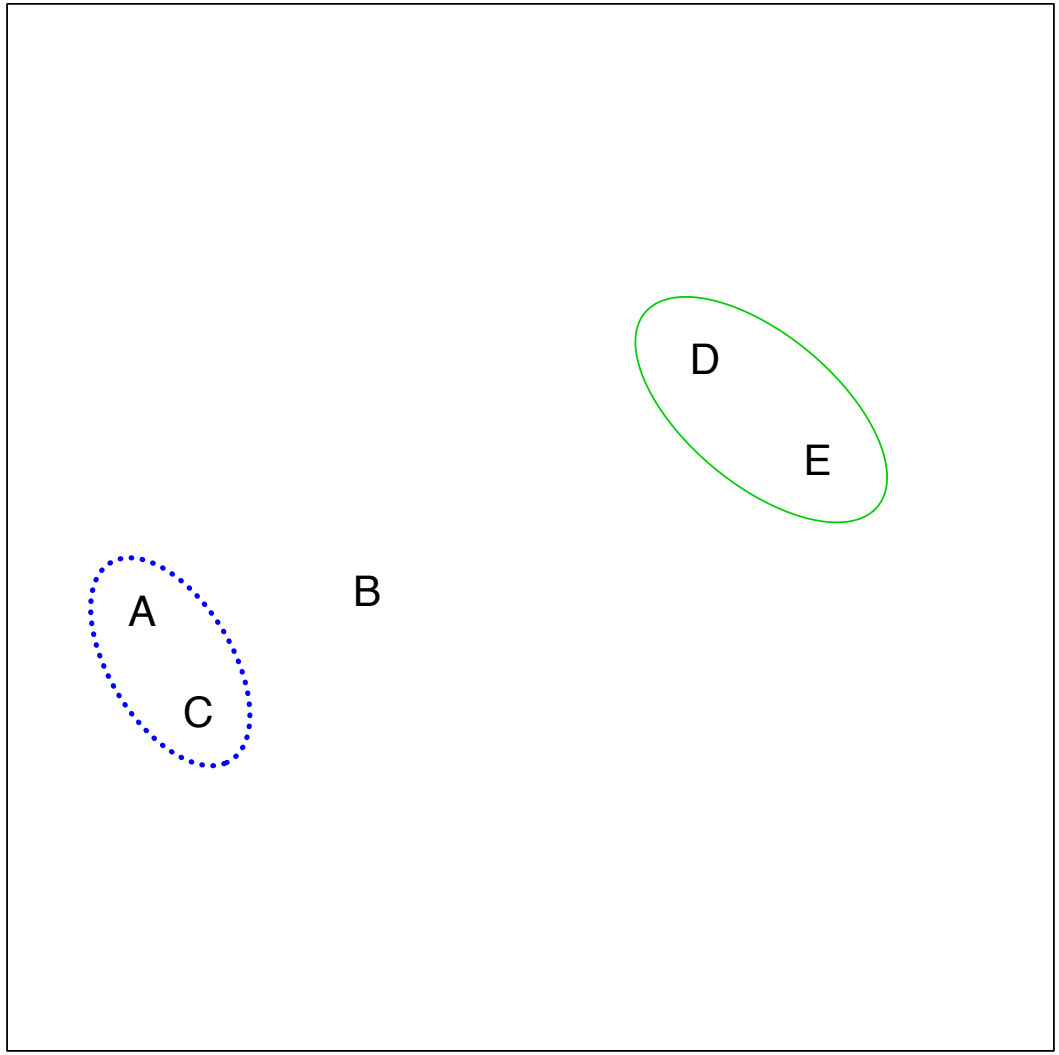

Builds a hierarchy in a "bottom-up" fashion

# Hierarchical Clustering: the idea



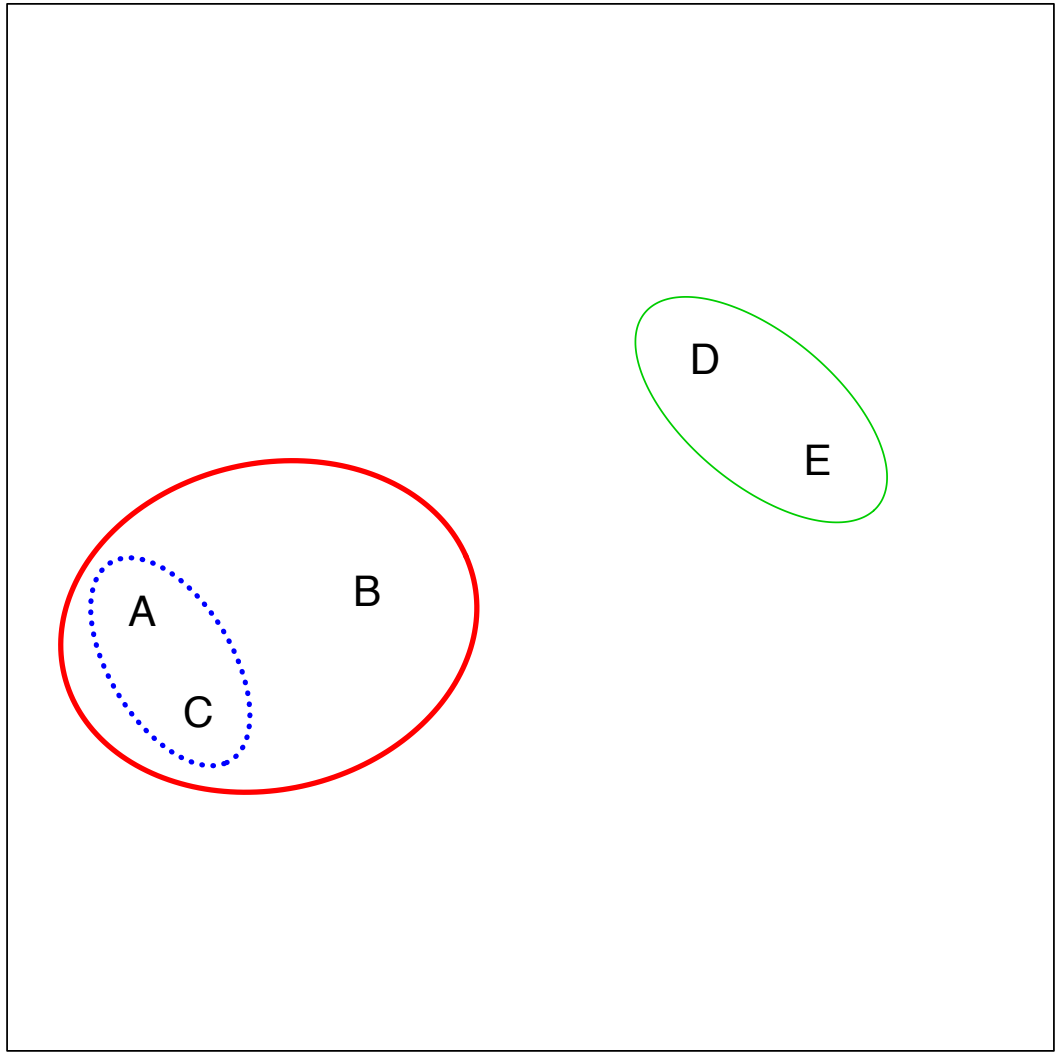

Builds a hierarchy in a "bottom-up" fashion

# Hierarchical Clustering: the idea



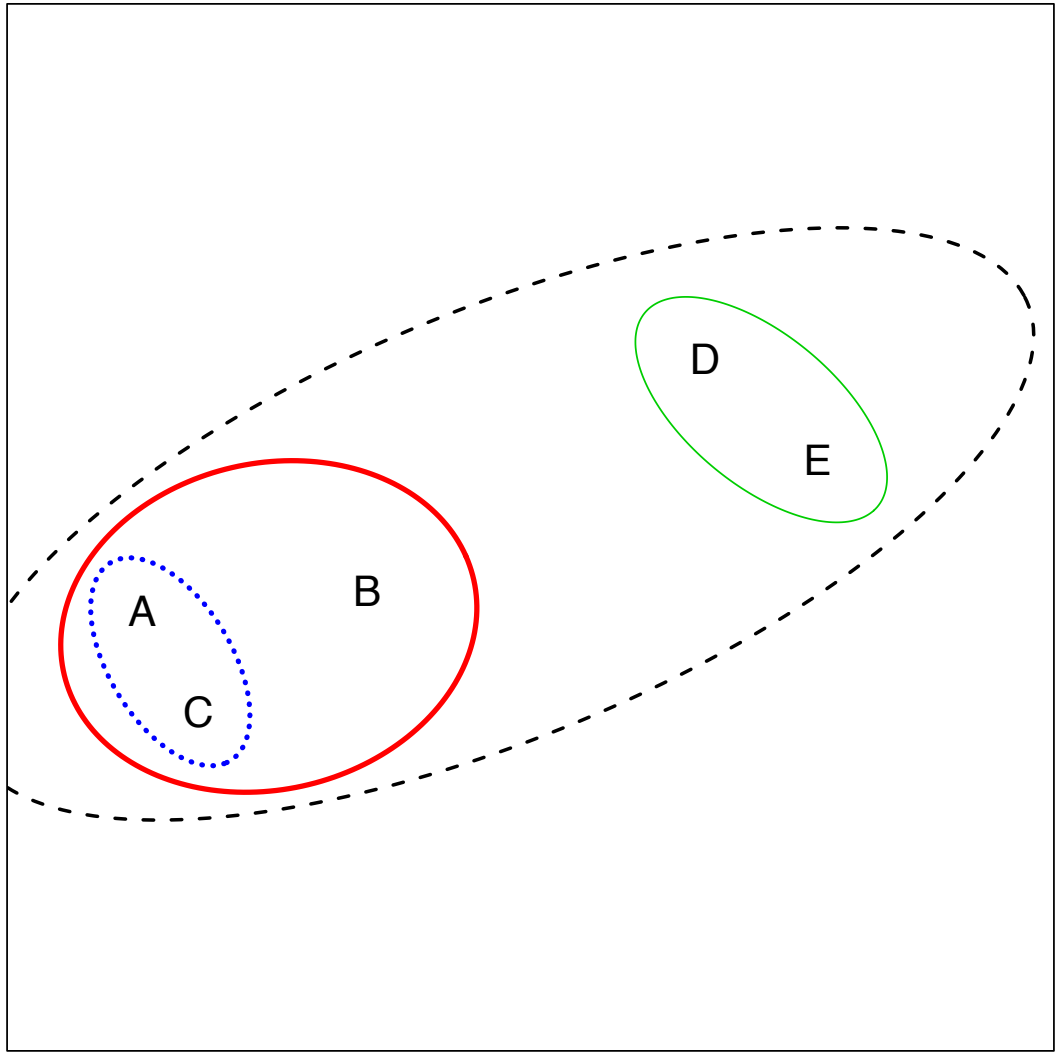

Builds a hierarchy in a "bottom-up" fashion

# Hierarchical Clustering: the idea




Builds a hierarchy in a "bottom-up" fashion

# Hierarchical Clustering: the idea

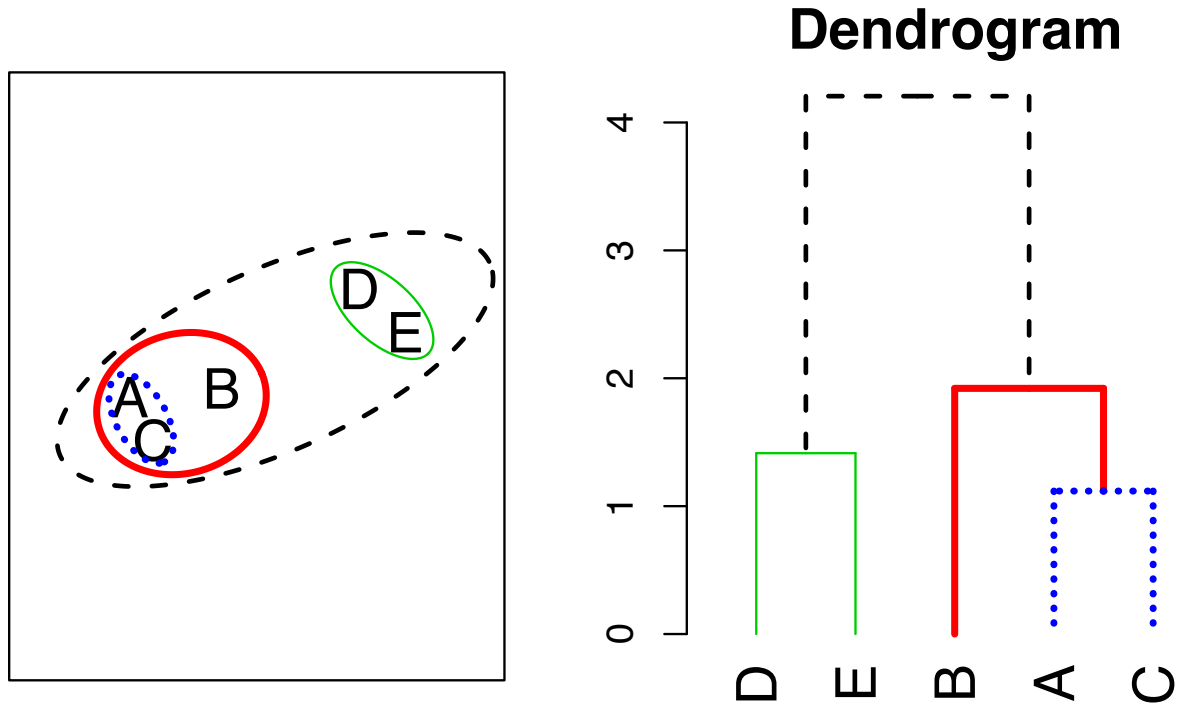



Builds a hierarchy in a "bottom-up" fashion

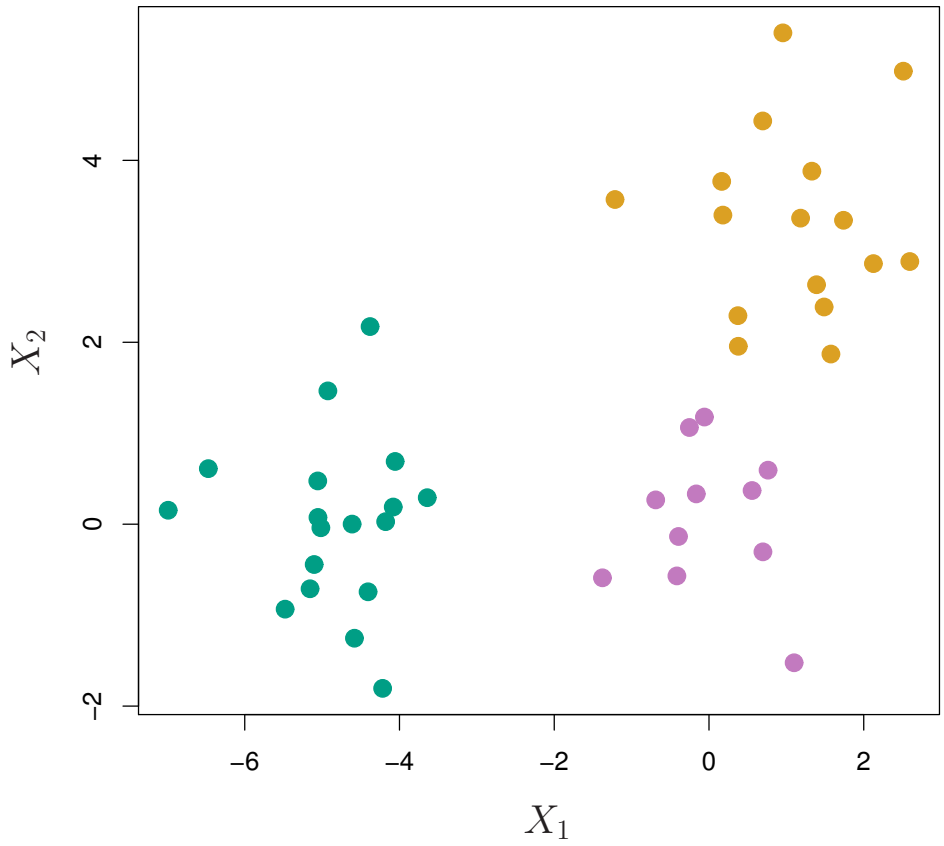
# Hierarchical Clustering Algorithm

**The approach in words:**

- Start with each point in its own cluster.
- Identify the closest two clusters and merge them. -Repeat.
- Ends when all points are in a single cluster.

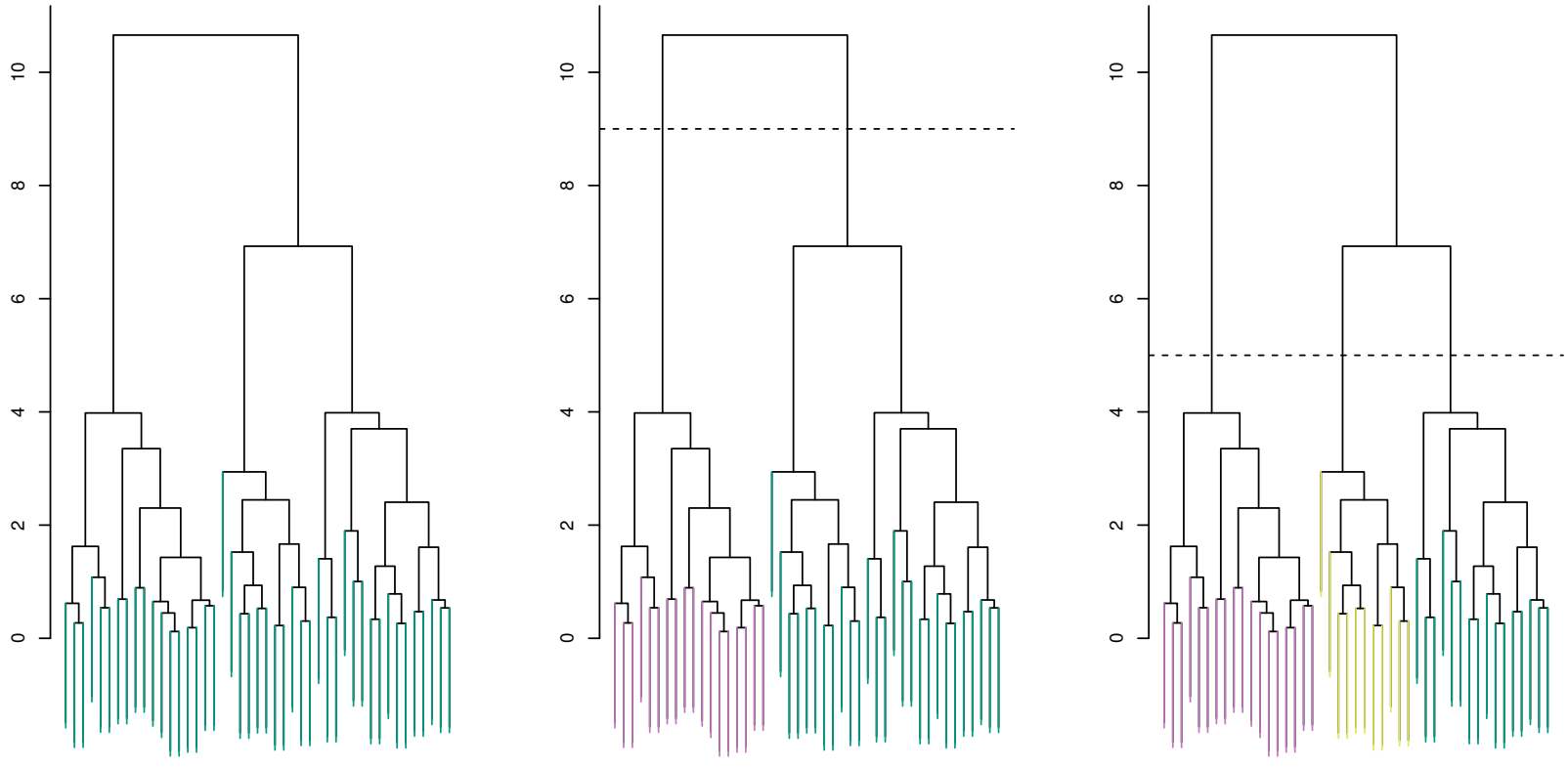


# An Example




45 observations generated in 2-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.

# Application of hierarchical clustering

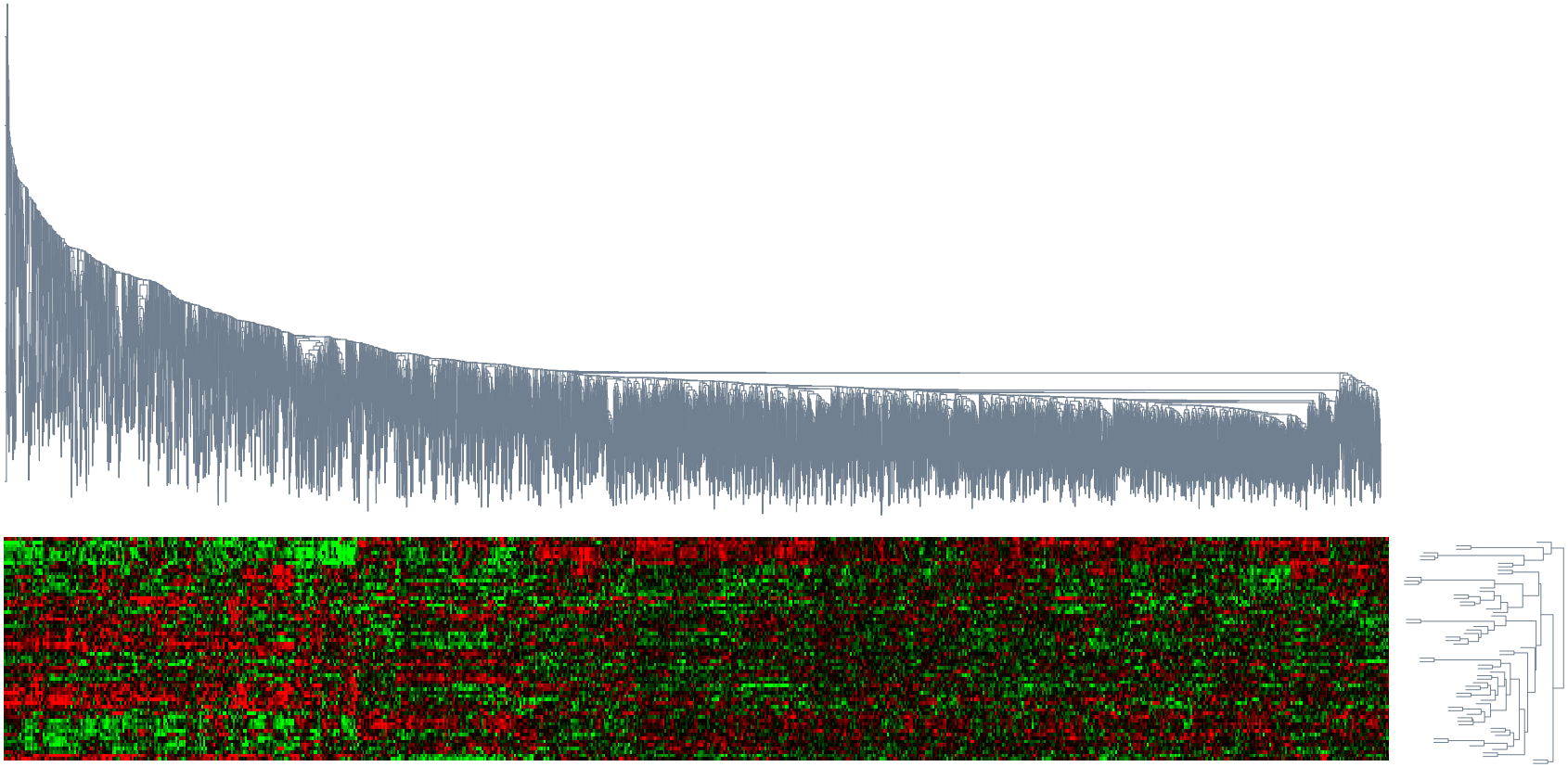



(see details in next slide)

# Details of previous figure

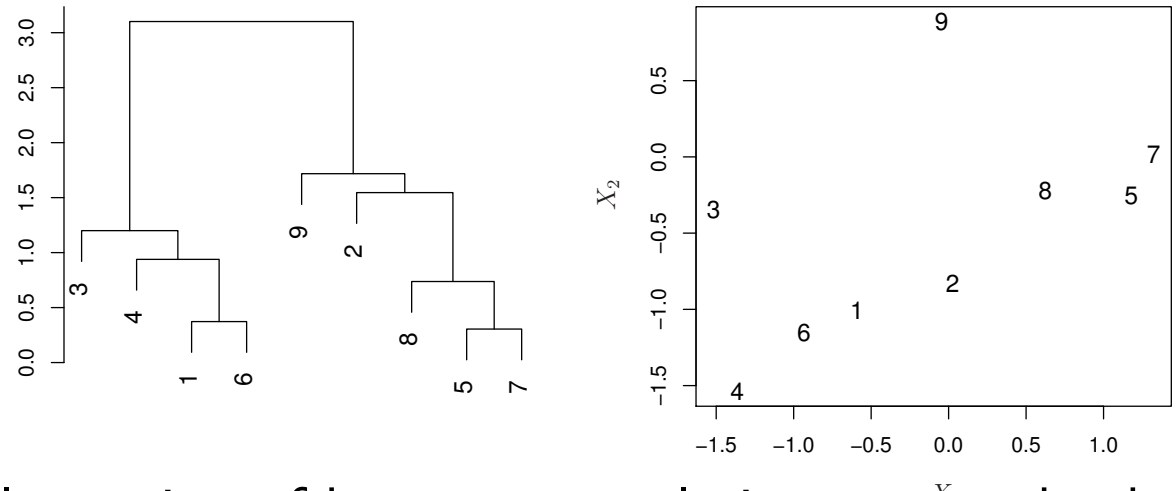
- Left: Dendrogram obtained from hierarchically clustering the data from previous slide, with complete linkage and Euclidean distance.
- Center: The dendrogram from the left-hand panel, cut at a height of 9 (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.
- Right: The dendrogram from the left-hand panel, now cut at a height of 5. 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
- Note: dendagram height is determined by the squared distance for that join.

# Application of hierarchical clustering



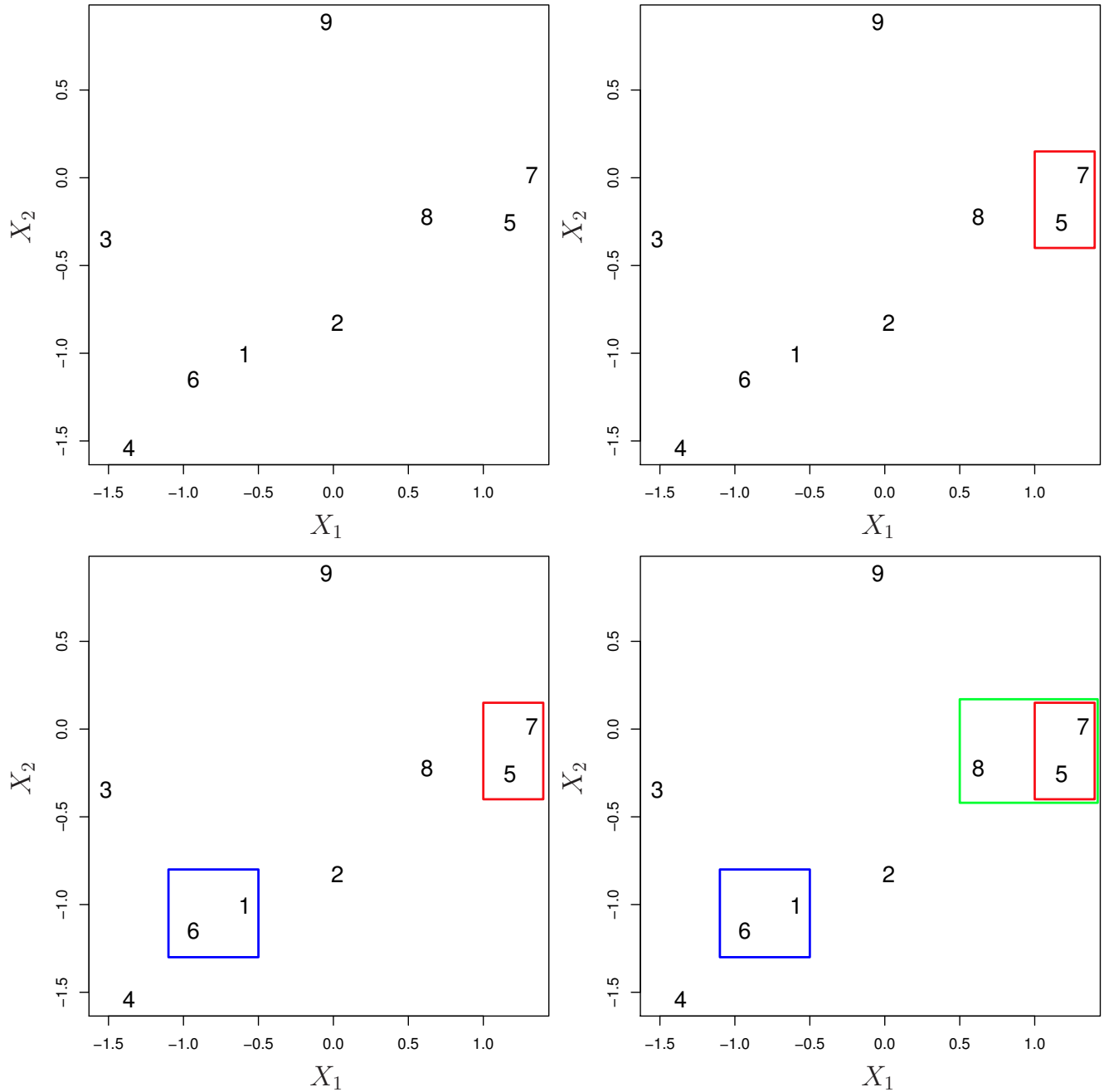
A gene expression matrix clusters genes with similar expression patterns together. This aids in the discovery of genes that may be associated with certain kinds of cancer. This displays genes (rows) and samples (columns) of the expression matrix in orderings derived from hierarchical clustering. (Source: ESL p. 527)

# Another Example




- An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. The raw data on the right was used to generate the dendrogram on the left.
- Observations 5 and 7 are quite similar to each other, as are observations 1 and 6.
- However, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7, even though observations 9 and 2 are close together in terms of horizontal distance.
- This is because observations 2, 8, 5, and 7 all fuse with observation 9 at the same height, approximately 1.8.

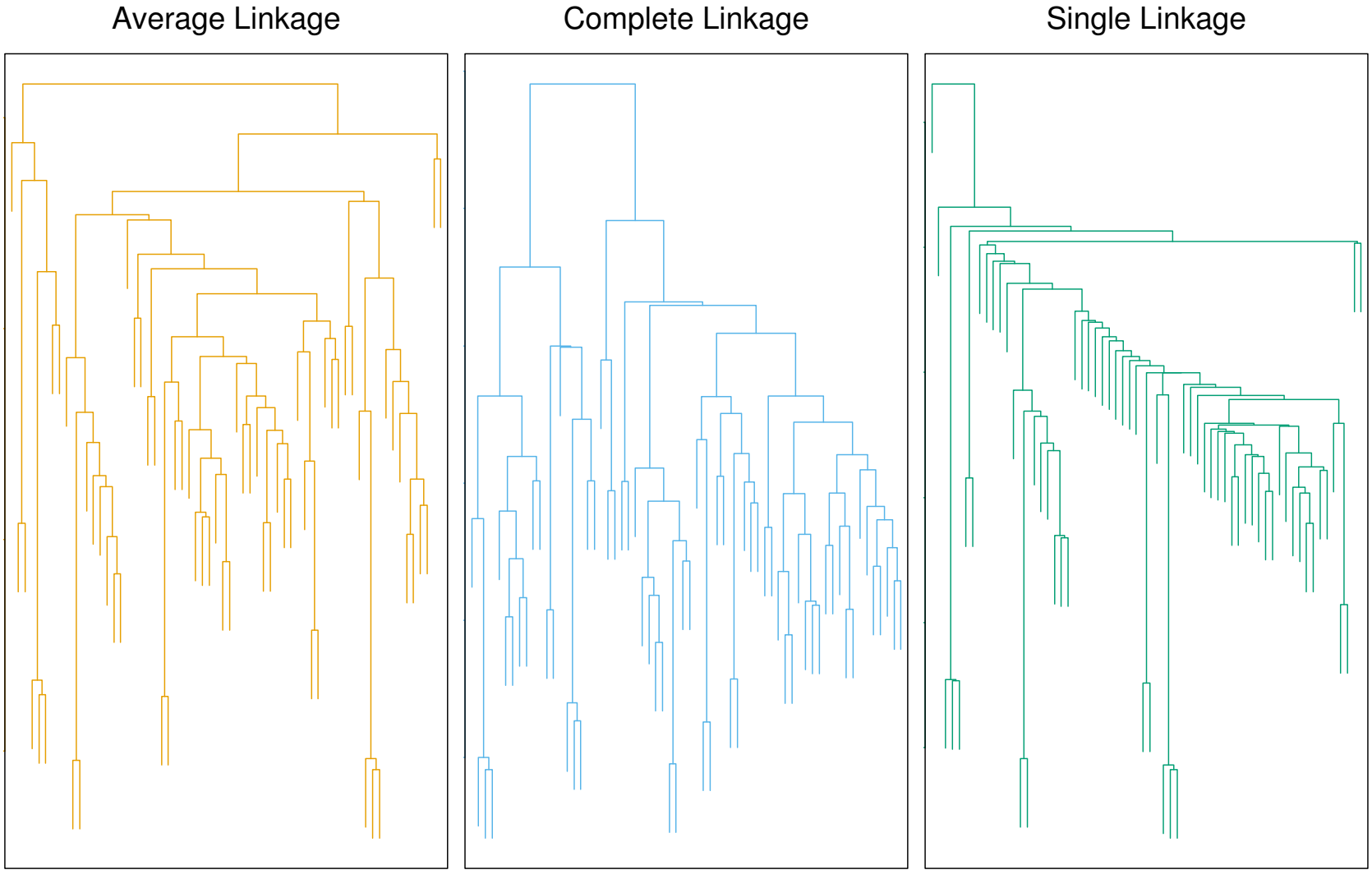
# Merges in previous example



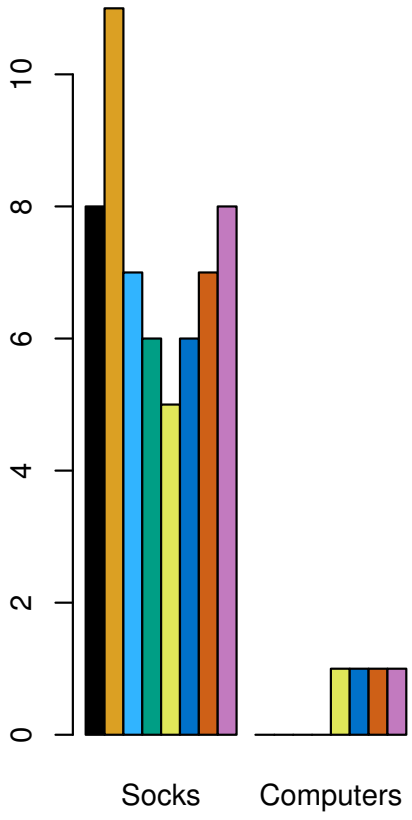
# Types of Linkage

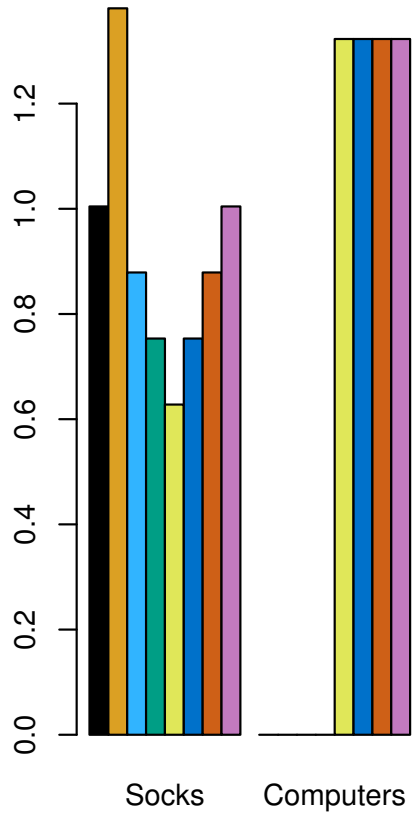
| **Linkage** | **description** |
|---|---|
| Complete | Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities. |
| Single | Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities. |
| Average | Mean inter-cluster 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. Cen- troid linkage can result in undesirable inversions. |

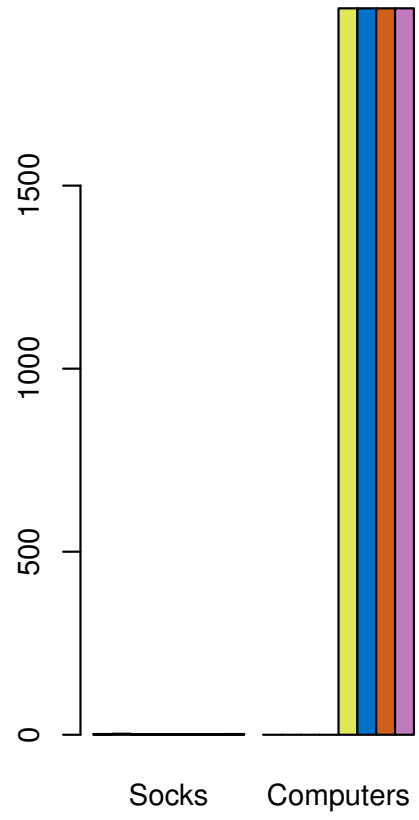
# Choice of Dissimilarity Measure



# Scaling of the variables matters



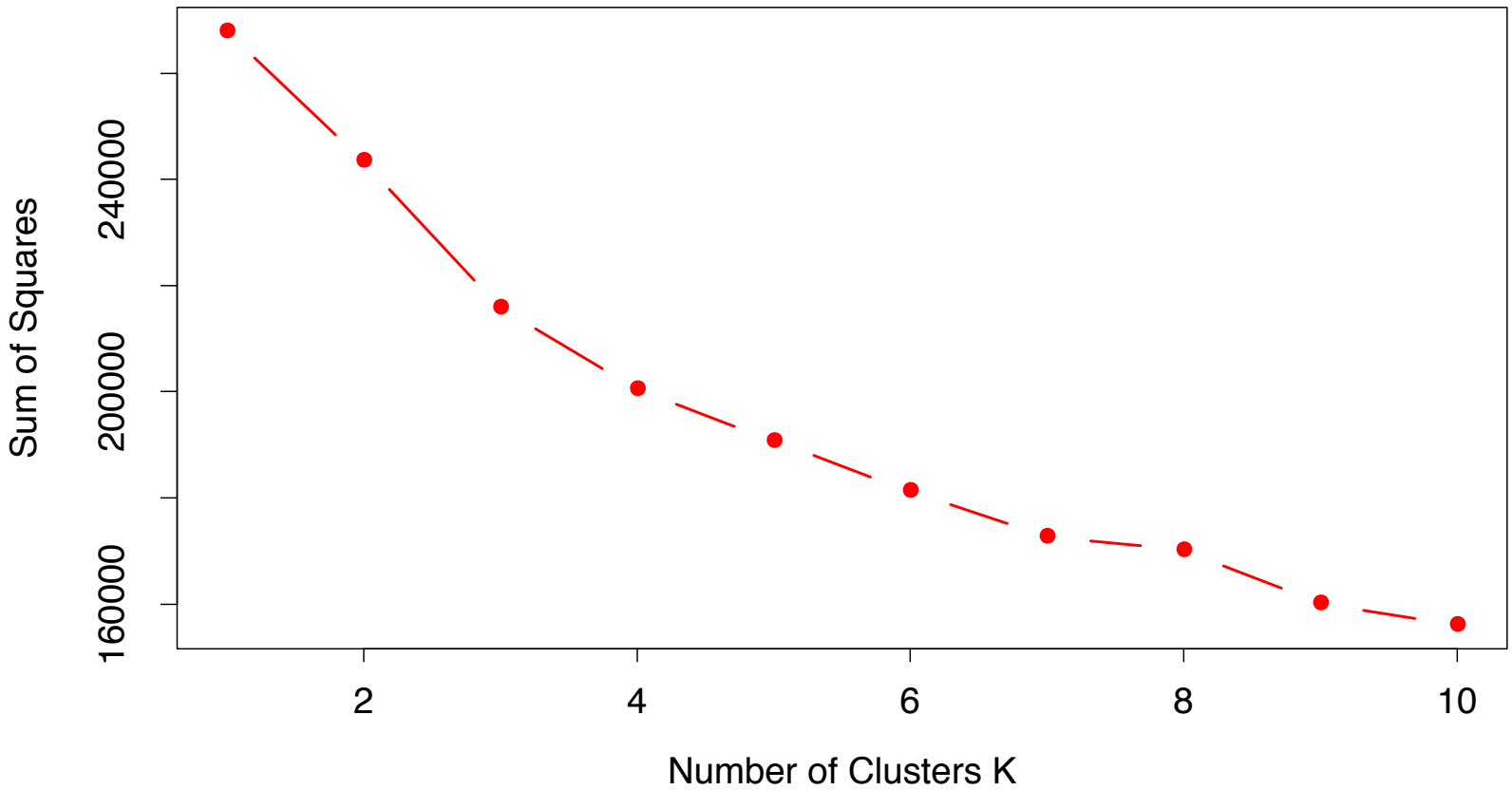
# Practical issues

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
- What dissimilarity measure should be used? -What type of linkage should be used?

# Issues with validity

- How many clusters to choose?
  - In all these cases we have assumed that the number of clusters $k$ is known.
  - This may not be justified if exploring a data set whose properties are unknown.
  - Deciding the number of clusters is a non-trivial problem.
  - There is no agreed-upon method. See Elements of Statistical Learning, chapter 13 for more details.
- Can we label clusters after discovering them?
- How do we measure error? How do we decide which model is best?

# Choosing $k$




Total within-cluster sum of squares for K-means clustering applied to the human tumor microarray data. Typically we look for an elbow in the sum of squares curve, but there isn't a clear one here. (Source ESL p. 513)

# Conclusions

- **Unsupervised learning** is important for understanding the variation and grouping structure of a set of unlabeled data, and can be a useful pre-processor for supervised learning
- It is intrinsically more difficult than **supervised learning** because there is no gold standard (like an outcome variable) and no single objective (like test set accuracy)
- It is an active field of research, with many recently developed tools such as **self-organizing maps**, **independent components analysis** and **spectral clustering**.