

PCA and clustering



Overview

Principal components analysis (PCA)

PCA in R

Clustering

Clustering in R

Principal Component Analysis

Supervised learning and unsupervised learning

In **supervised learning** we have a response variable y , along with explanatory variables x_1, x_2, \dots, x_k

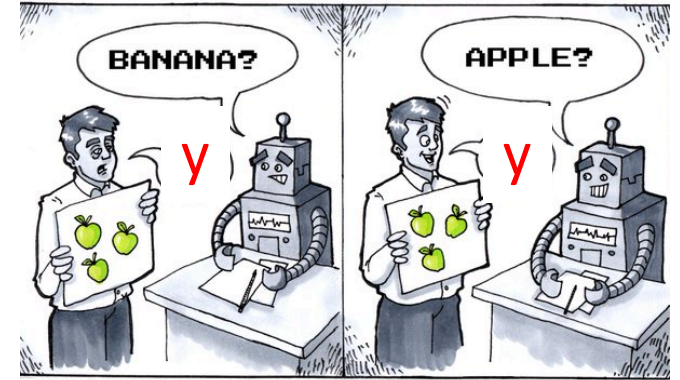
- For example, linear and logistic regression are supervised learning problems because we model a response variable y as a function of several explanatory variables, x_1, x_2, \dots, x_k

In **unsupervised learning**, we have explanatory variables x_1, x_2, \dots, x_k but **no** response variable y

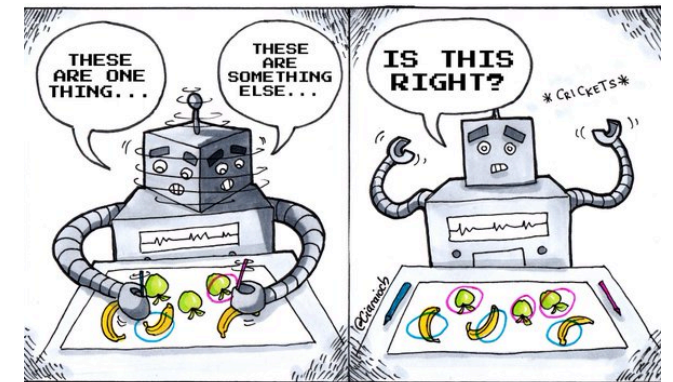
Unsupervised learning can be useful in order to find structure in the data and to visualize patterns

A key challenge in unsupervised learning is that there is no real ground truth response variable y

- So we don't have measures like RSS to see how well our model is fitting the data



Supervised Learning

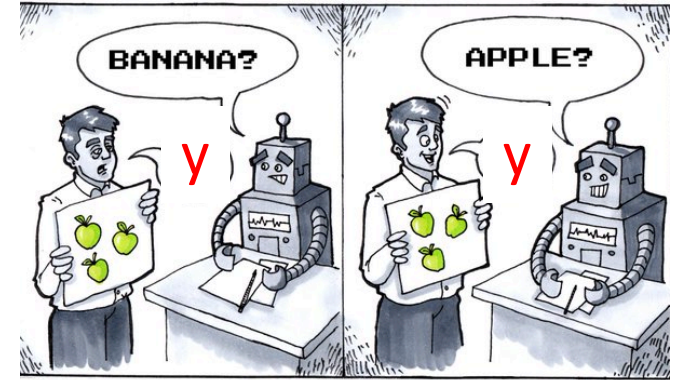


Unsupervised Learning

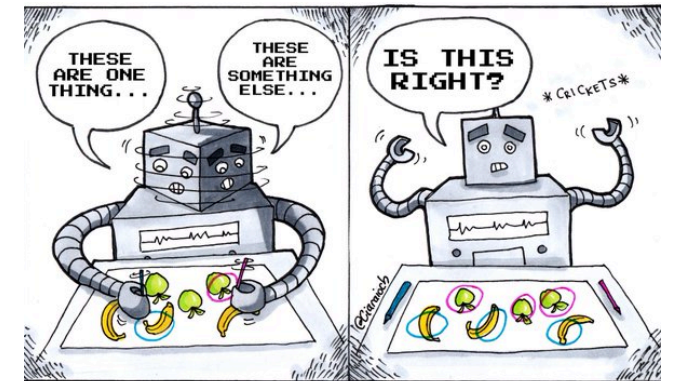
Unsupervised learning

We will discuss two types of unsupervised learning:

1. **Dimensionality reduction** where we try to find a smaller set of features that captures most of the variability in the data
 - Principal component analysis (PCA)
2. **Clustering** where we try to group similar data points together



Supervised Learning



Unsupervised Learning

Principal Component Analysis

Dimensionality reduction methods find a new smaller set of explanatory variables that capture key properties in the data:

- $f(x_1, x_2, \dots, x_k) \longrightarrow t_1, t_2, \dots, t_d$ where $d \ll k$
- This can be useful for visualization if d is 2 or 3

The diagram illustrates the transformation of a data matrix. On the left, a matrix of size $n \times k$ is shown, with rows representing observations and columns representing original features. The matrix is enclosed in large red brackets labeled n (vertical) and k (horizontal). The matrix elements are $x_{11}, x_{12}, \dots, x_{1k}$ in the first row, $x_{21}, x_{22}, \dots, x_{2k}$ in the second row, \vdots in the third row, and $x_{n1}, x_{n2}, \dots, x_{nk}$ in the last row. A horizontal arrow points to the right, where a matrix of size $n \times d$ is shown. This matrix is also enclosed in large red brackets labeled n (vertical) and d (horizontal). The matrix elements are t_{11}, t_{12} in the first row, t_{21}, t_{22} in the second row, \vdots in the third row, and t_{n1}, t_{n2} in the last row.

$$\begin{matrix} & \overbrace{\hspace{10em}}^k \\ \underbrace{\hspace{1em}}_n \left[\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{array} \right] & \longrightarrow & \underbrace{\hspace{1em}}_n \left[\begin{array}{cc} \overbrace{\hspace{2em}}^d \begin{array}{cc} t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{array} \end{array} \right. \end{matrix}$$

Principal Component Analysis

Principal Component Analysis is a dimensionality method that can be thought of in two ways:

1. As a transformation that captures most of the variability in the original data
2. As a transformation that tries to preserve the distance between all points

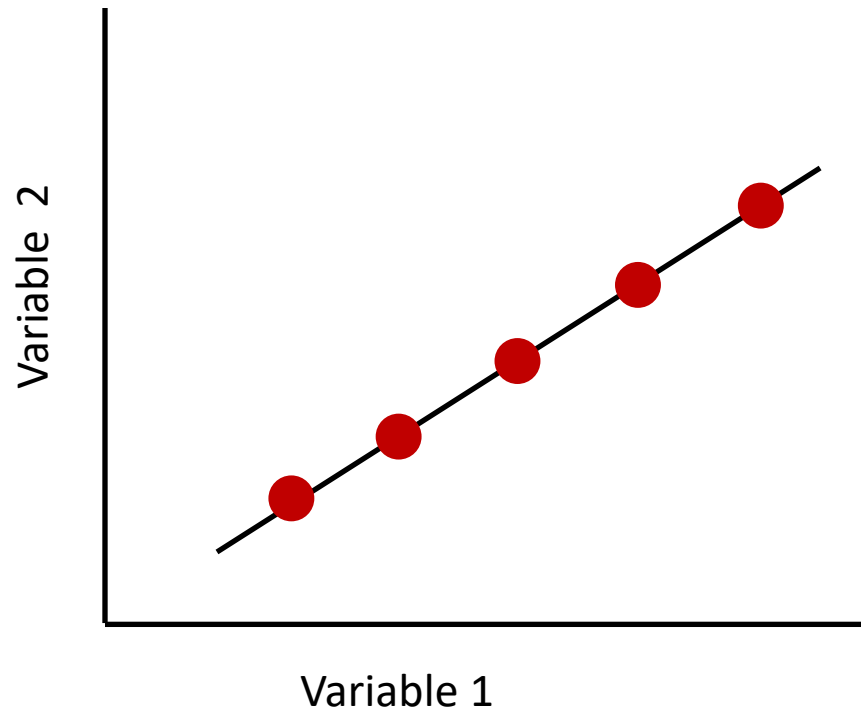
The diagram illustrates the transformation of a data matrix. On the left, a matrix of size $n \times k$ is shown, with rows representing data points and columns representing features. The matrix is enclosed in large red brackets labeled n (vertical) and k (horizontal). The matrix elements are $x_{11}, x_{12}, \dots, x_{1k}$ in the first row, $x_{21}, x_{22}, \dots, x_{2k}$ in the second row, \vdots in the third row, and $x_{n1}, x_{n2}, \dots, x_{nk}$ in the n -th row. An arrow points to the right, where a matrix of size $n \times d$ is shown. This matrix is also enclosed in large red brackets labeled n (vertical) and d (horizontal). The matrix elements are t_{11}, t_{12} in the first row, t_{21}, t_{22} in the second row, \vdots in the third row, and t_{n1}, t_{n2} in the n -th row.

$$\begin{matrix} & \overbrace{\hspace{1.5cm}}^k \\ \underbrace{\hspace{1cm}}_n \left[\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{array} \right] & \longrightarrow & \underbrace{\hspace{1cm}}_n \left[\begin{array}{cc} \overbrace{\hspace{1cm}}^d \\ t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{array} \right] \end{matrix}$$

Principal Component Analysis

Suppose that two features are highly correlated.

We can summarize their joint values (x_1, x_2) using a single features t_1



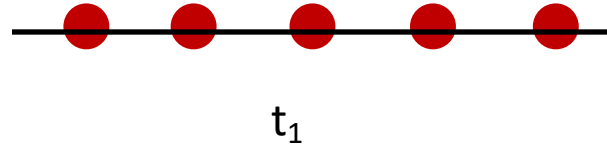
$$t_1 = \frac{1}{2} x_1 + \frac{1}{2} x_2$$

Principal Component Analysis

Suppose that two features are highly correlated.

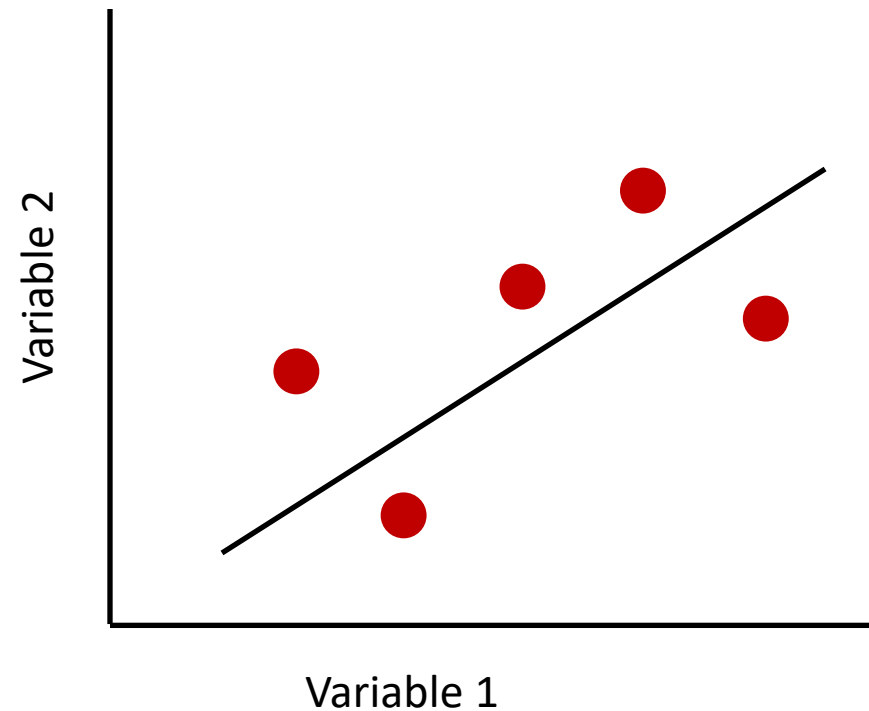
We can summarize their joint values (x_1, x_2) using a single features t_1

$$t_1 = \frac{1}{2} x_1 + \frac{1}{2} x_2$$



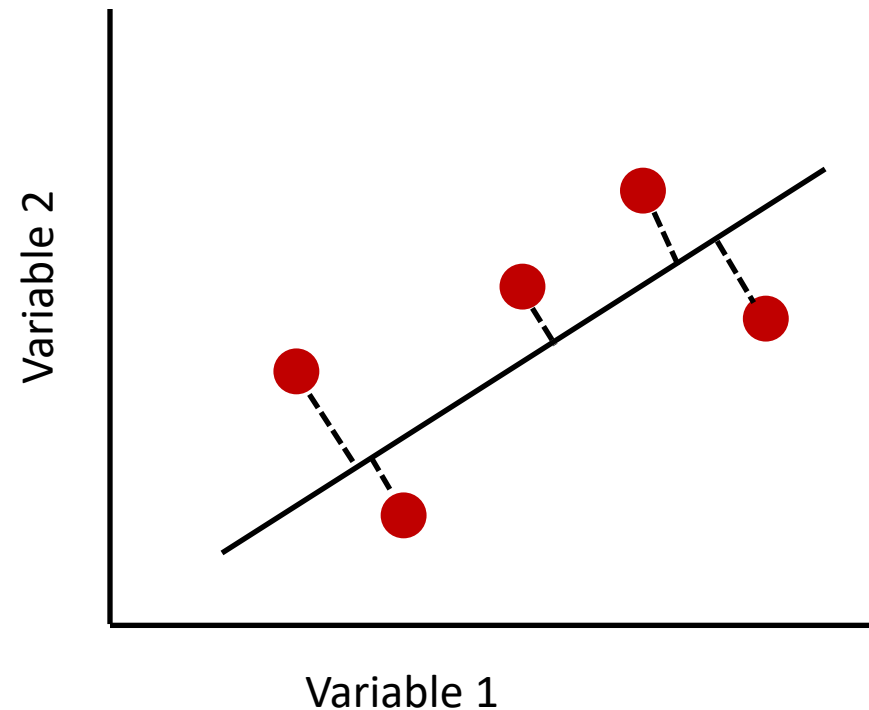
Principal Component Analysis

We can also do this even if the correlation is not perfect



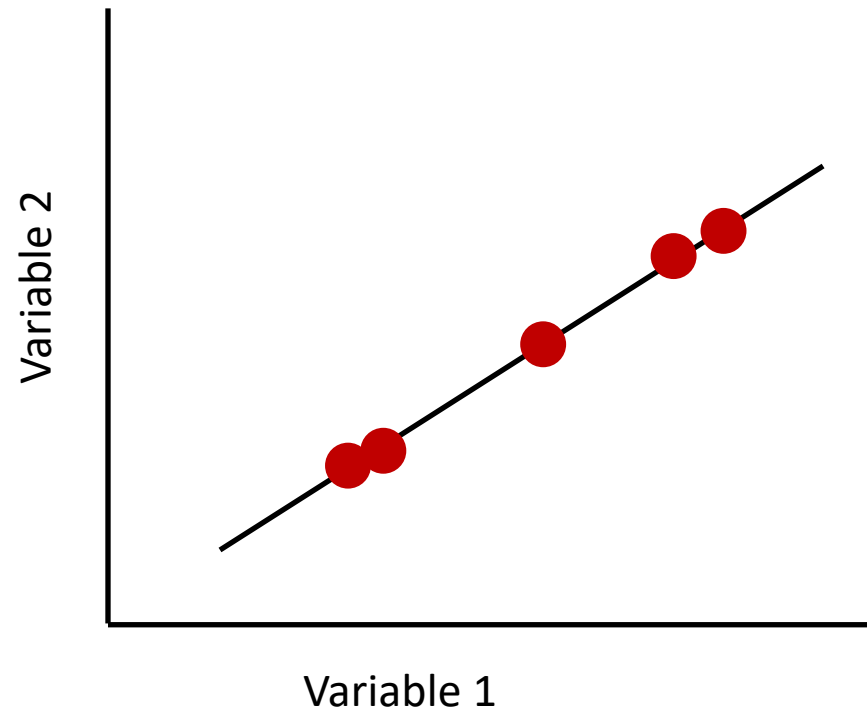
Principal Component Analysis

We can also do this even if the correlation is not perfect



Principal Component Analysis

We can also do this even if the correlation is not perfect



Principal Component Analysis

Principal component **scores** t_i 's are linear combinations of the original variables x_{ij} 's:

$$t_{i1} = \alpha_{11}x_{i1} + \alpha_{21}x_{i2} + \dots + \alpha_{k1}x_{ik}$$

α_{j1} are the **loadings** for the first principal component

- The "norm" of the loadings is 1

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

We can do this for each case in our data set we get values: t_{11}, \dots, t_{n1}

Principal Component Analysis

To run PCA, we start by centering each variable x_i so that it has a mean of 0

We also usually divide all variables by their standard deviation

- i.e., z-score transform the features before performing PCA
- We divide by the s_i 's so that variables with large variances don't dominate

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$
$$\begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_k \end{bmatrix}$$
$$\begin{bmatrix} s_1 & s_2 & \dots & s_k \end{bmatrix}$$

Principal Component Analysis

The loadings for the first principal component are found by finding the projection vector $A_1 = (\alpha_{11}, \alpha_{21}, \alpha_{k1})$ such that the variance of the t_i is maximized

Find the α 's that maximize:

$$\frac{1}{n-1} \sum_{i=1}^n t_i^2 = \frac{1}{n-1} \sum_{i=1}^n (\alpha_{11}z_{i1} + \alpha_{21}z_{i2} + \dots + \alpha_{k1}z_{ik})^2$$

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

Subject to the constraint:

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$

Principal Component Analysis

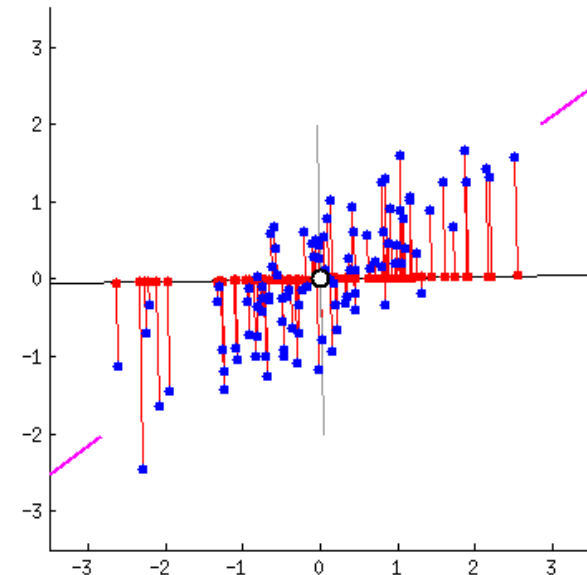
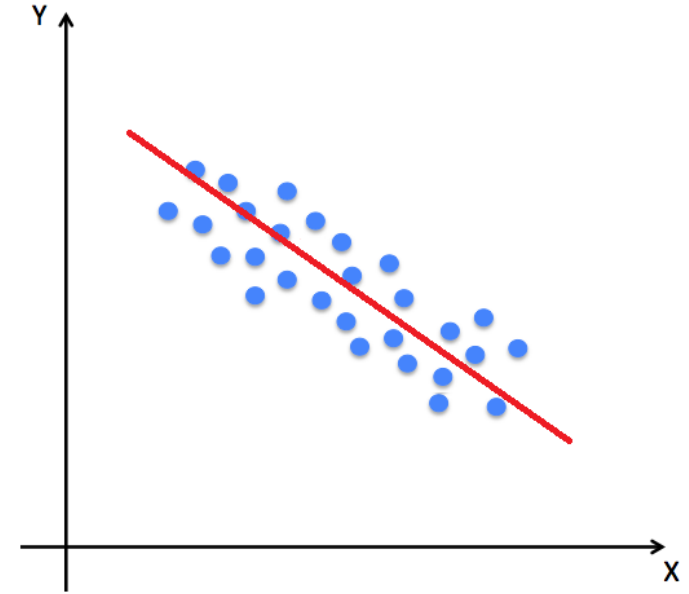
The loadings for the first principal component are found by finding the projection vector $A_1 = (\alpha_{11}, \alpha_{21}, \alpha_{k1})$ such that the variance of the t_i is maximized

Find the α 's that maximize:

$$\frac{1}{n-1} \sum_{i=1}^n t_i^2 = \frac{1}{n-1} \sum_{i=1}^n (\alpha_{11}z_{i1} + \alpha_{21}z_{i2} + \dots + \alpha_{k1}z_{ik})^2$$

Subject to the constraint:

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$



The Second Principal Component

The second principal component scores t_{i2} is the linear combination of the z_1, z_2, \dots, z_k that has maximal variance and is **uncorrelated** with the first principal component scores t_{i1}

- $t_{i2} = \alpha_{12}z_1 + \alpha_{22}z_2 + \dots + \alpha_{k2}z_k$
- $\text{cor}(T_1, T_2) = 0$

This is equivalent of having A_1 be orthogonal to A_2

- $A_1^T A_2 = 0$
$$\sum_{j=1}^k \alpha_{j1} \cdot \alpha_{j2} = 0$$

First principal component

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

Second principal component

$$\begin{bmatrix} t_{12} \\ t_{22} \\ \vdots \\ t_{n2} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{k2} \end{bmatrix}$$

The Second Principal Component

The second principal component scores t_{i2} is the linear combination of the z_1, z_2, \dots, z_k that has maximal variance and is **uncorrelated** with the first principal component scores t_{i1}

- $t_{i2} = \alpha_{12}z_1 + \alpha_{22}z_2 + \dots + \alpha_{k2}z_k$
- $\text{cor}(T_1, T_2) = 0$

This is equivalent of having A_1 be orthogonal to A_2

- $A_1^T A_2 = 0$
$$\sum_{j=1}^k \alpha_{j1} \cdot \alpha_{j2} = 0$$

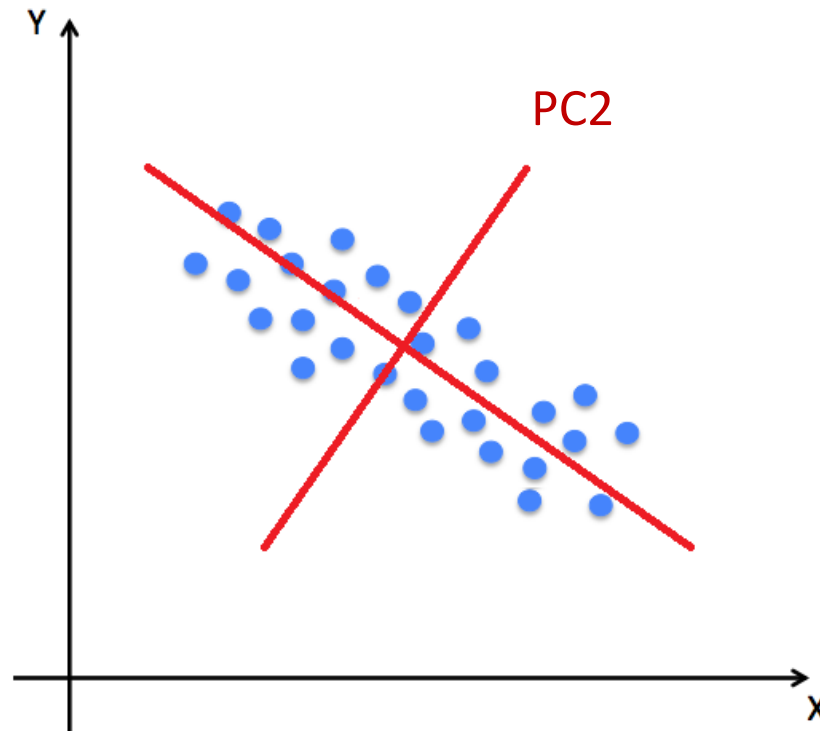
First and second principal components

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \\ \vdots & \vdots \\ \alpha_{k1} & \alpha_{k2} \end{bmatrix}$$

Geometric interpretation of the second PC

Find the direction that maximizes the variance of t_i 's

- Data projected on to the principal component is most spread out that is perpendicular (orthogonal) to the other PCs



Higher Principal Components

We continue this process until we find all the principal component scores, T_1, T_2, \dots, T_d

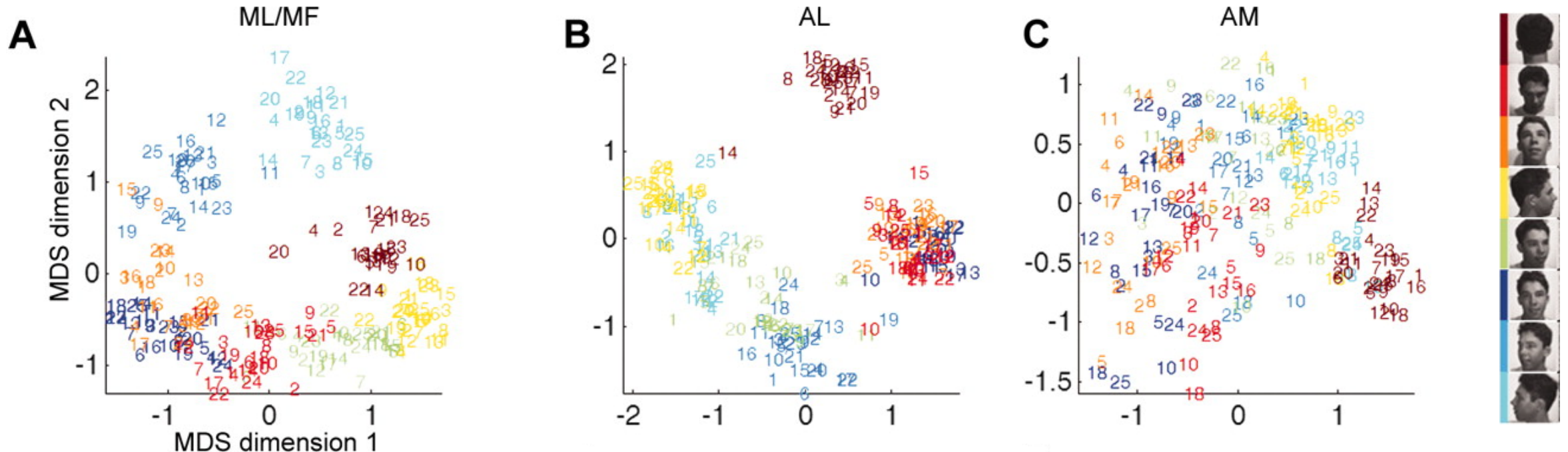
- The principal component scores are unique up to a sign flip $T_i = -T_i$
 - To find the principal components what is really done is an eigenvalue decomposition of the covariance matrix.

All principal components

$$\begin{bmatrix} t_{11} & t_{12} & \dots & t_{1d} \\ t_{21} & t_{22} & \dots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nd} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1d} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{k1} & \alpha_{k2} & \dots & \alpha_{kd} \end{bmatrix}$$

Neuroscience example

Freiwald and Tsao (Science 2010) used dimensionality reduction to reduce the activity of a large population of neurons to two dimensions so that they could visualize how different brain regions represent faces.

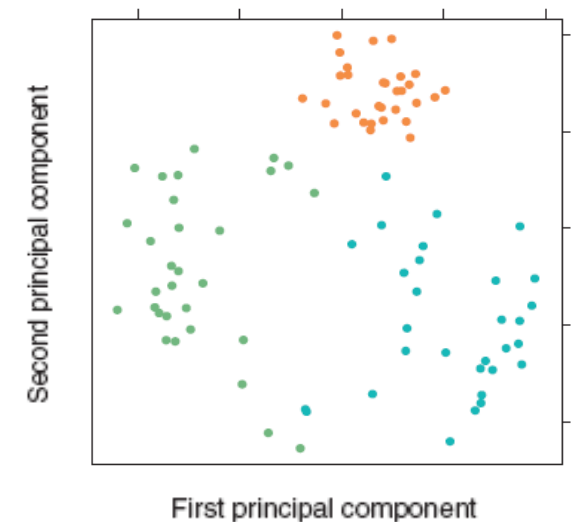
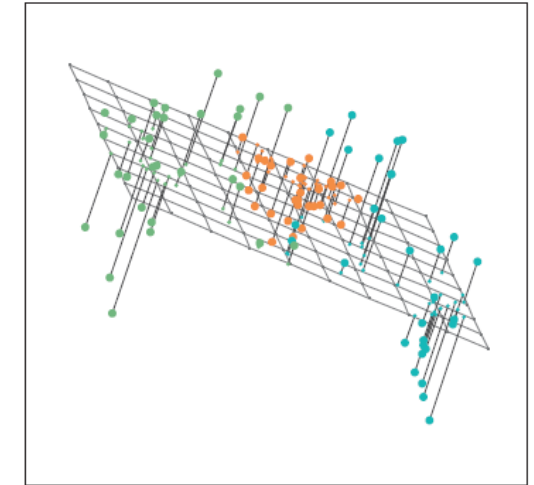


Another interpretation of PCA

PCA can also be viewed as reducing the dimension of the data from k dimensions to d dimensions that approximately preserves the distances between all pairs of points.

- Suppose x_j and x_l are two data points in our original k dimensional representation
- Suppose t_j and t_l are the same two data points in our PCA transformed d dimensional representation
- Then $\text{dist}(x_i, x_j) \approx \text{dist}(t_i, t_j)$

$$\text{dist}(x_j, x_l) = \sqrt{\sum_{i=1}^k (x_{ji} - x_{li})^2} \approx \sqrt{\sum_{i=1}^d (t_{ji} - t_{li})^2} = \text{dist}(t_j, t_l)$$



Proportion of Variance Explained

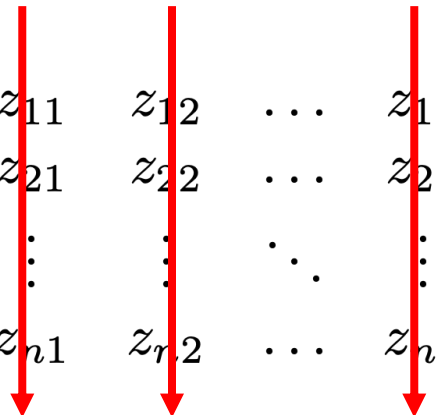
In order to know how many principal components to use, it is usual to assess the **proportion of variance explained** (PVE) by each PC

Total variance:
$$\sum_{j=1}^k Var(z_j) = \sum_{j=1}^k \frac{1}{n-1} \sum_{i=1}^n (z_{ij})^2$$

Variance explained by m^{th} principal component:


$$Var(t_m) = \frac{1}{n-1} \sum_{i=1}^n t_{im}^2 = \frac{1}{n-1} \sum_{i=1}^n \left(\sum_{j=1}^k \alpha_{jm} z_{ij} \right)^2$$

$$\begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix}$$



 $Var(z_1)$

$$\begin{bmatrix} t_{11} & t_{12} & \dots & t_{1d} \\ t_{21} & t_{22} & \dots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nd} \end{bmatrix}$$



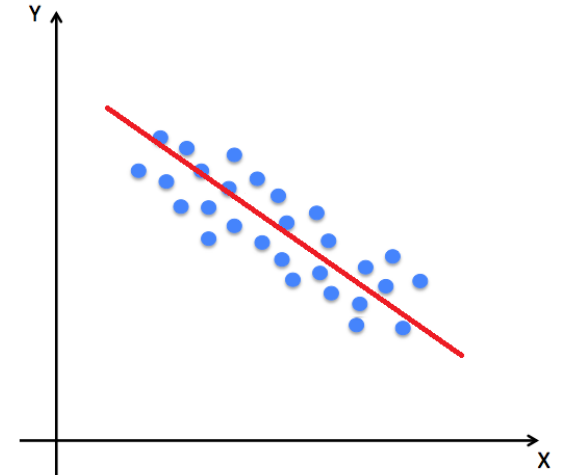
 $Var(t_m)$

Proportion of Variance Explained

Proportion *of variance that is explained* by each PC:

$$PVE_{mv} = \frac{\text{Variance explained by } m^{\text{th}} \text{ principal component}}{\text{Total variance}}$$

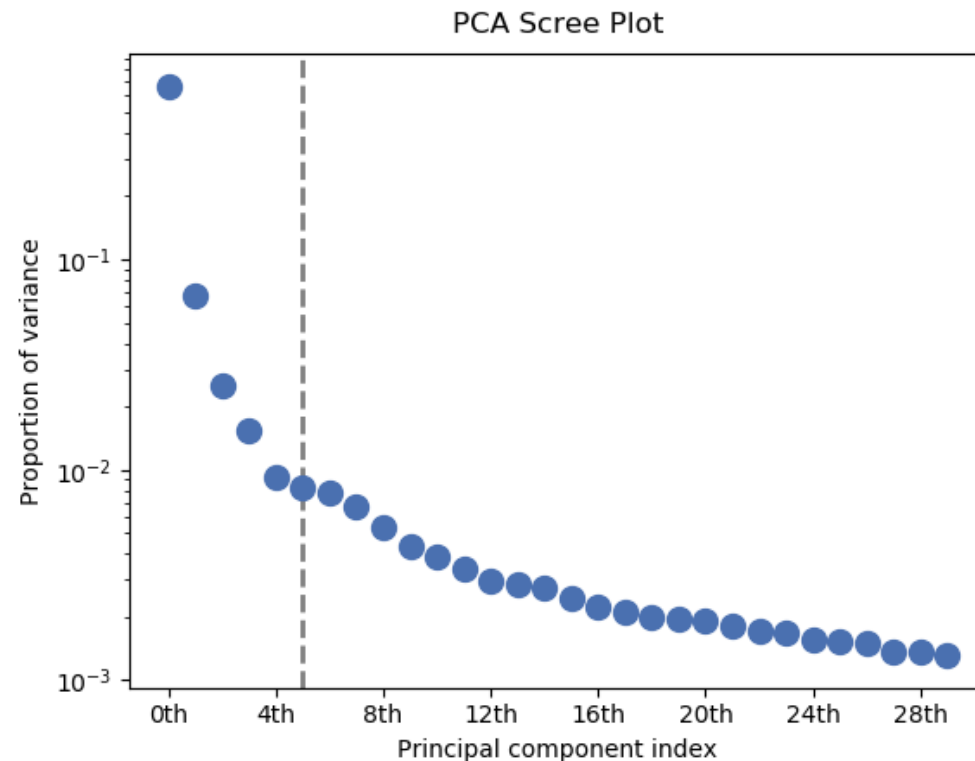
$$PVE_m = \frac{Var(t_m)}{Var(total)} = \frac{\sum_{i=1}^n (\sum_{j=1}^k \alpha_{jm} z_{ij})^2}{\sum_{j=1}^k \sum_{i=1}^n z_{ij}^2}$$



Deciding how many PCs to use

A **scree plot** shows the PVE as a function of PC number

- The number of PCs chosen is often selected by looking for the “elbow” in this plot
 - i.e., point where PVE stop dramatically dropping and levels off



PCA example: personality traits of fictional characters

The [Open-Source Psychometrics Project](#) conducted a survey where they got ratings of 235+ personality traits from 800 fictional characters.

Let's use PCA to assess:

- How to personality traits commonly covary
- Which fictional characters are most similar

If you want to find out which fictional character you are most similar you can take their [“Which Character” personality quiz](#)

Rate characters from Good Will Hunting:



Where does Will Hunting fall on this spectrum?

oppressed  privileged

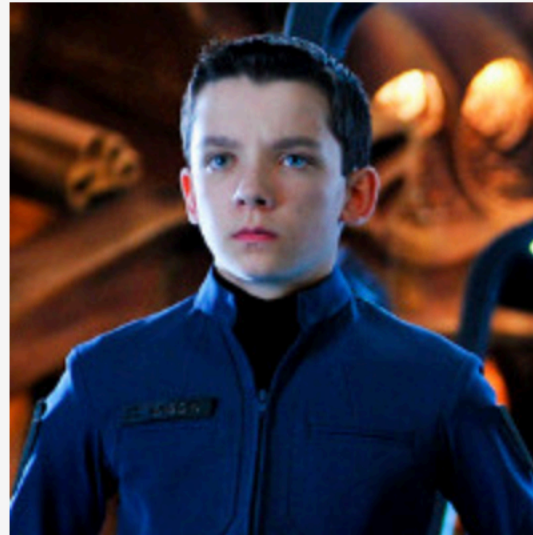
Answer

(don't know, skip)

19/25

Let's try the PCA in R...

The best match between the self assessment you provided and the profile of a fictional character as rated by other people who have taken this survey is the character Ender Wiggin (Ender's Game).



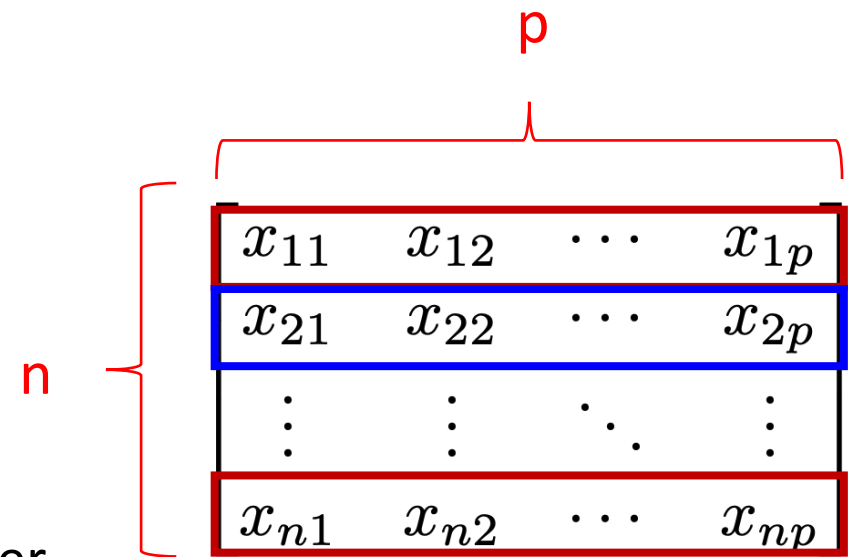
84% match

Your traits versus their traits are graphed below (click on points for labels).

Clustering

Clustering divides n data points x_i 's into subgroups

- Data points in the same group are similar/homogeneous
- Data points in different groups are different from each other



Examples:

- Examining gene expression levels to group cancer types together
- Examining consumer purchasing behavior to perform market segmentation

Clustering can be:

- **Flat:** no structure beyond dividing points into groups
- **Hierarchical:** Population is divided into smaller and smaller groups (tree like structure)

K-means clustering

K-means clustering partitions the data into **K** distinct, non-overlapping clusters

- i.e., each data point x_i belongs to exactly one cluster C_k

The number of clusters, **K** , needs to be specified prior to running the algorithm

The goal is to minimize the within-cluster variation for some measure $W(C_k)$

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

K-means clustering

A common within cluster similarity measure $W(C_k)$ is the sum of the **Euclidean distance** between all pairs of points in a cluster:

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

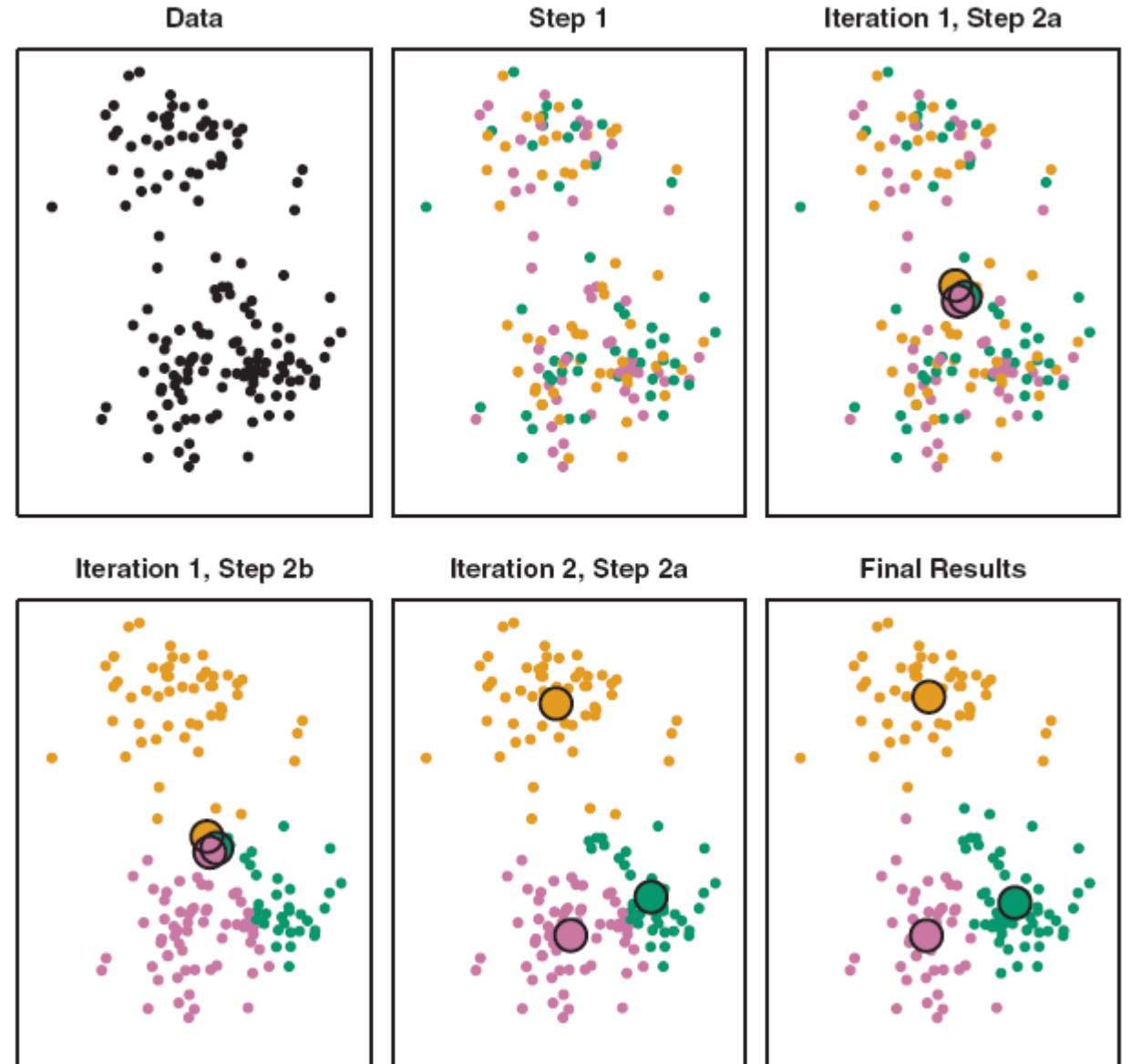
This is equivalent to minimizing: $\frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2$

Finding the exact optimal solution is computationally intractable (there are k^n possible partitions), but a simple algorithm exists to find a local optimum which often works well in practice.

x_{11}	x_{12}	\cdots	x_{1p}
x_{21}	x_{22}	\cdots	x_{2p}
\vdots	\vdots	\ddots	\vdots
x_{n1}	x_{n2}	\cdots	x_{np}

K-means clustering

1. Randomly assign points to clusters C_k
2. Calculate cluster centers as means of points in each cluster
3. Assign points to the closest cluster center
4. Recalculate cluster center as the mean of points in each cluster
5. Repeat steps 3 and 4 until convergence



K-means clustering

Because only a local minimum is found, different random initializations will lead to different solutions

- One should run the algorithm multiple times to get better solutions

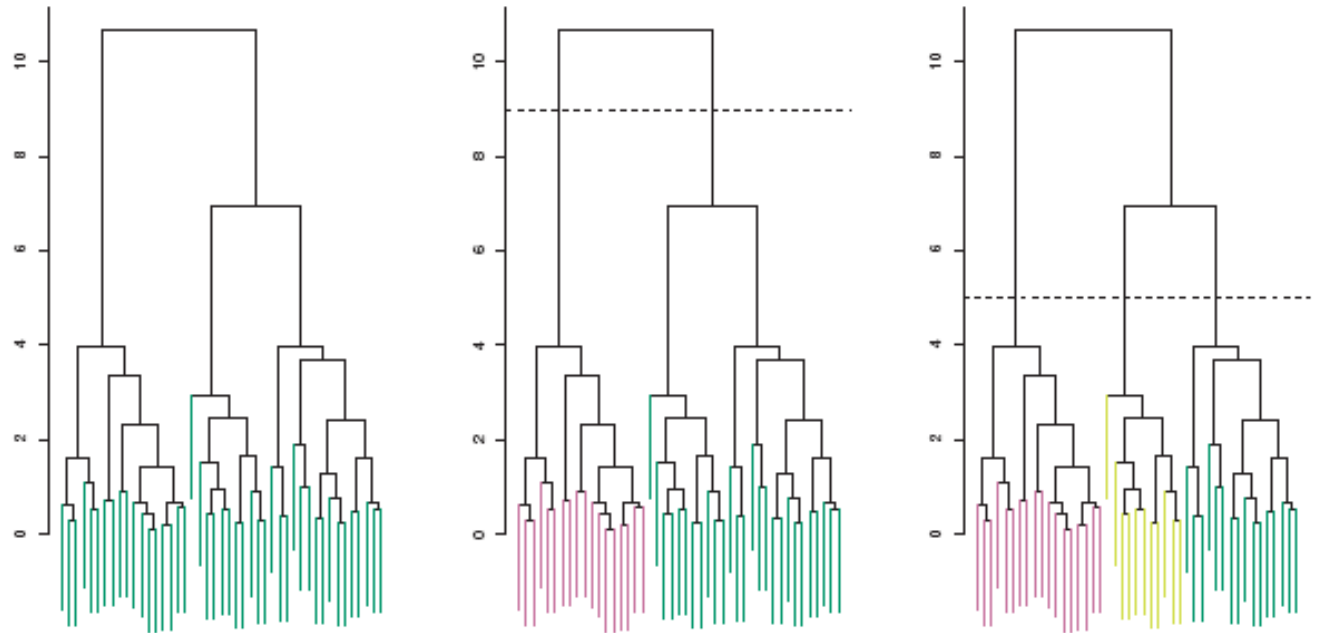


Hierarchical clustering

In **hierarchical clustering** we create a dendrogram which is a tree-based representation of successively larger clusters.

We can cut the dendrogram at any point to create as many clusters as desired

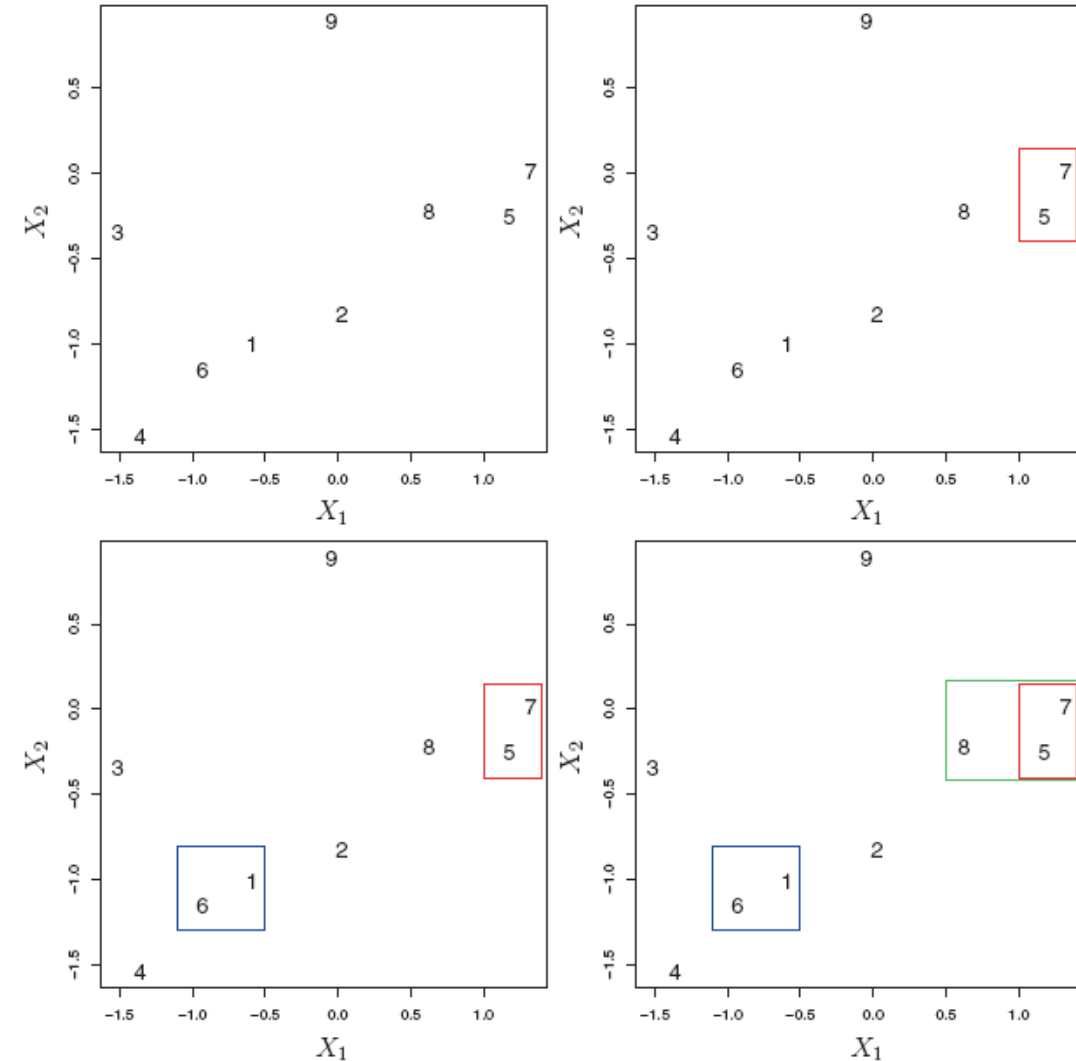
- i.e., don't need to specify the number of clusters, K , beforehand



Hierarchical clustering

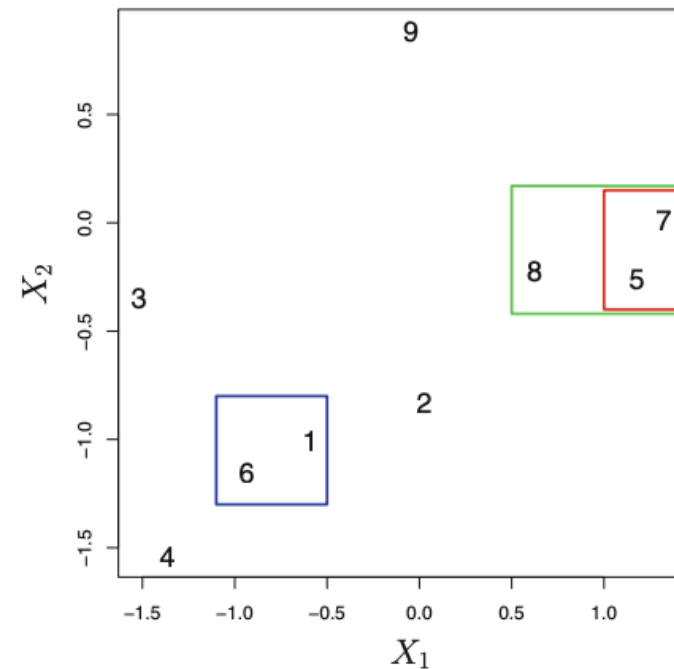
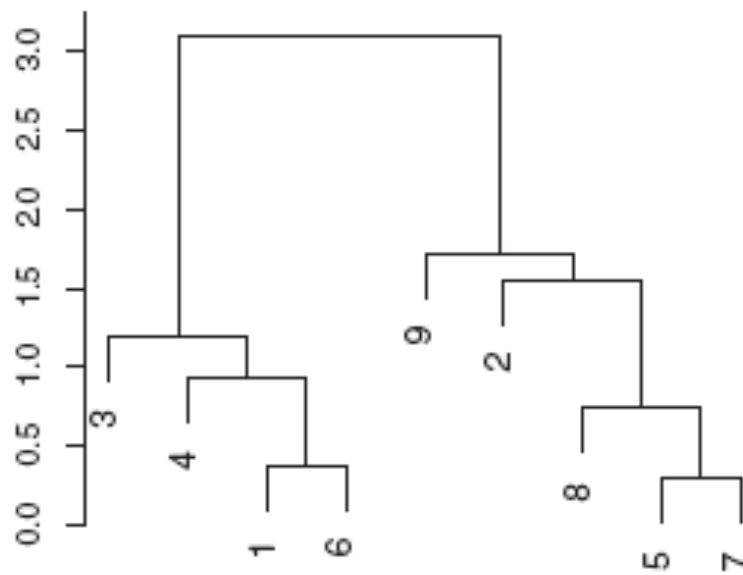
We can create a hierarchical clustering of the data using simple bottom-up agglomerative algorithm:

1. Choosing a (dis)similarity measure
 - E.g., The Euclidean distance
2. Initializing the clustering by treating each point as its own cluster
3. Successively merging the pair of clusters that are most similar
 - i.e., calculate the similarity between all pairs of clusters and merging the pair that is most similar
4. Stopping when all points have been merged into a single cluster



Hierarchical clustering

The vertical height that two clusters/points merge show how similar the two *clusters* are



Note: horizontal distance between *individual points* is not important:

- point 9 is considered as similar to point 2 as it is to point 7

Hierarchical clustering choices

We can define the similarity between two data points using the Euclidean distance or another measure, but how do we define similarity between groups of data points?

- A few choices for 'linkage' functions are:

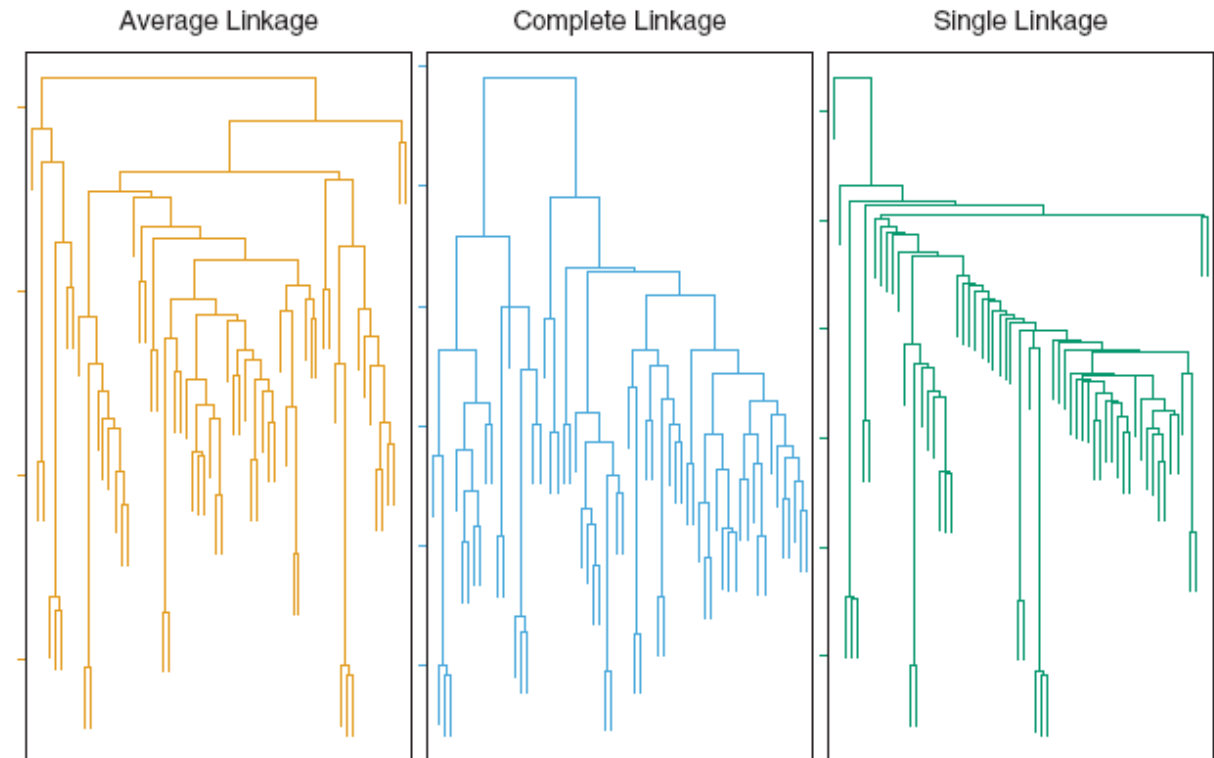
Complete	Compute the dissimilarity between all pairs of points in the two clusters. The cluster dissimilarity is defined as the <i>maximum dissimilarity</i> between all points.
Single	Compute the dissimilarity between all pairs of points in the two clusters. The cluster dissimilarity is defined as the <i>minimum dissimilarity</i> between all points.
Average	Compute the dissimilarity between all pairs of points in the two clusters. The cluster dissimilarity is defined as the <i>average dissimilarity</i> between all points.
Centroid	Compute the dissimilarity between centroids (i.e., the means) of the two clusters.

Hierarchical clustering choices

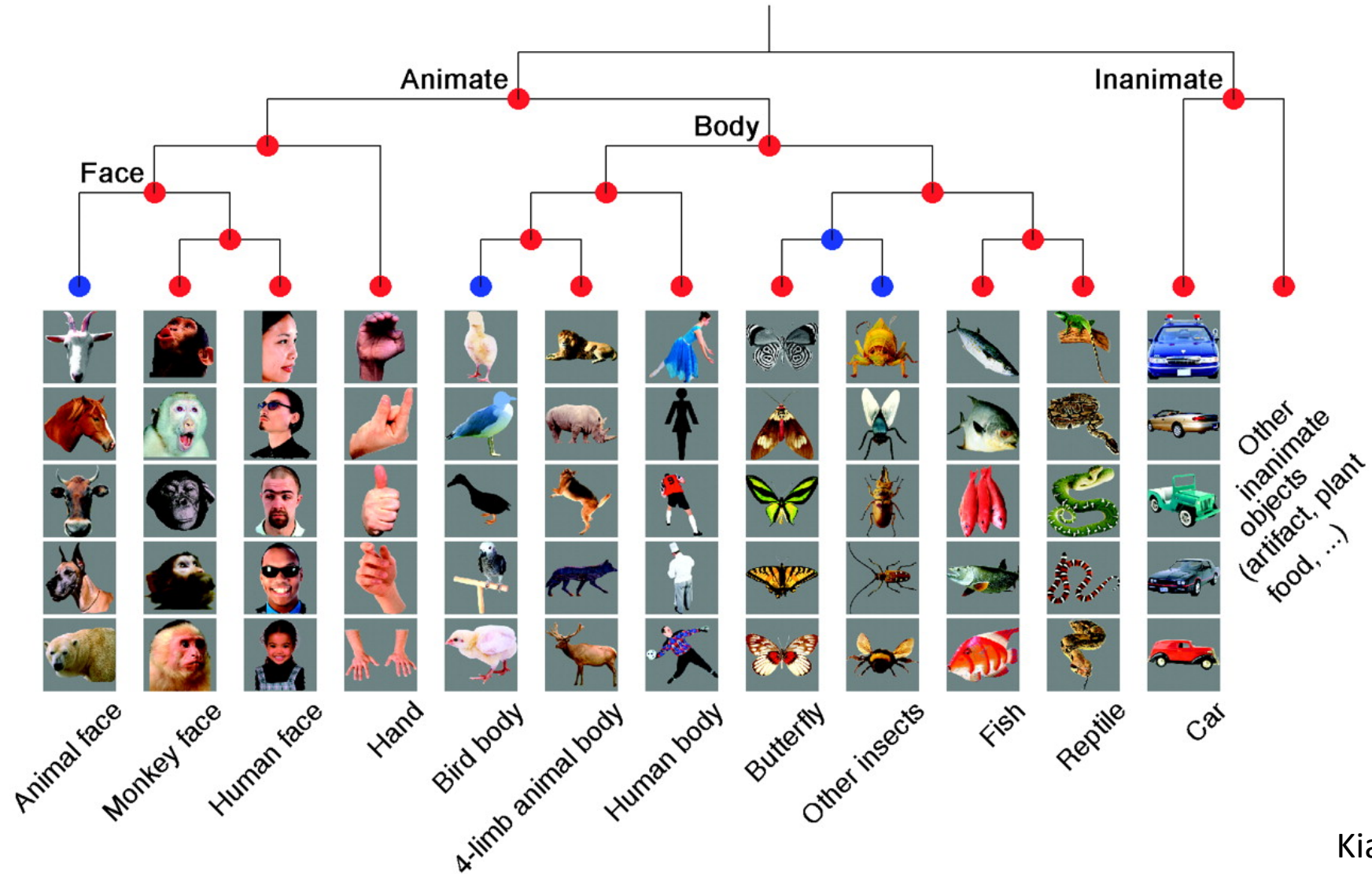
Generally average and complete linkage chosen over single linkage since they tend to yield more balanced trees

Centroid linkage can lead to inversions in which two clusters can be merged below the height of the individual clusters

- This makes it impossible to visualize the clustering as a tree



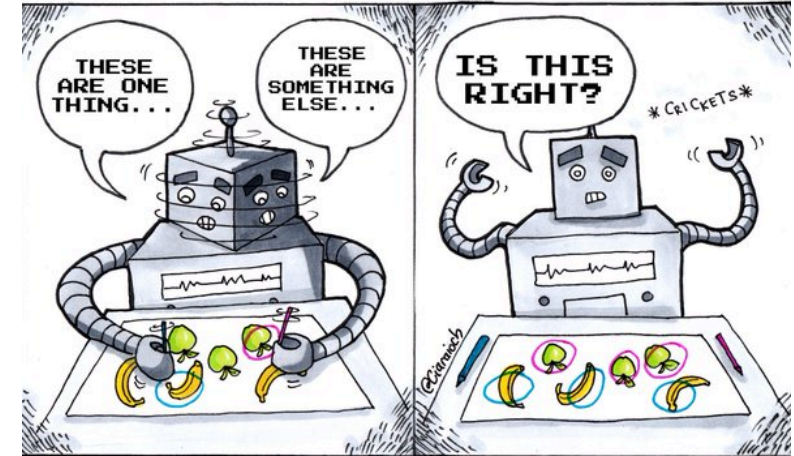
Hierarchical clustering example



Issues with clustering

Choices made can effect the results:

- Feature normalization and/or dissimilarity measure
- K-means: choice of K
- For hierarchical cluster: linkage and cut height



Unsupervised Learning

Potential approaches to deal with these issues:

- Try a few methods and see if one gives interesting/useful results
- Validate that you get similar results on a second set of data

Let's try clustering in R...