# Pre-class progress check

## By now, you should have

1. Started thinking about your group project proposals (due 11.06 by 11.59pm)
2. Reviewed slides and code from classes 1-5

1. Reviewed slides and code from class 6
2. Reviewed graded assignment 1 (will release answer key shortly)

1. Started working on group project
2. Reviewed slides from class 7 (today)

CLASS 1

# Mapping what we have learnt so far…

| | Features (input) | Targets/ labels (output) | Task type | ML type | Metric |
|---|---|---|---|---|---|
| *Prediction of brain volume from gestational age* | ✅ | ✅ (Numeric) | Regression | **Supervised learning** | *RMSE, R-squared* |
| *Classification of heart failure using ultrasound or magnetic resonance imaging* | ✅ | ✅ (Categorical) | Classification | | *Accuracy, Precision, Recall, AUC* |
| *Patient health profiles using wearable device data* | ✅ | 🚫 | Clustering | **Unsupervised learning** | *Silhouette score* |
| *Compressing information from 30 features to study similarities/differences in benign and malignant breast tumors* | ✅ | 🚫 | Dimensionality reduction | | *Explained Variance* |

University of Pittsburgh | School of Public Health

# Unsupervised learning (I): Clustering (I)
## - k-means
## - hierarchical clustering

**BIOST 2155**

**Class 7**
**October 16th, 2026**

# On clustering

## Can we discover subgroups among the observations or variables?

When we cluster the observations of a data set, we want to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.

**Market Segmentation:** Finding groups of customers with similar purchasing behavior.

**Biology:** Grouping genes with similar expression patterns, or patients by disease subtype.

**Image Recognition:** Grouping similar images together.

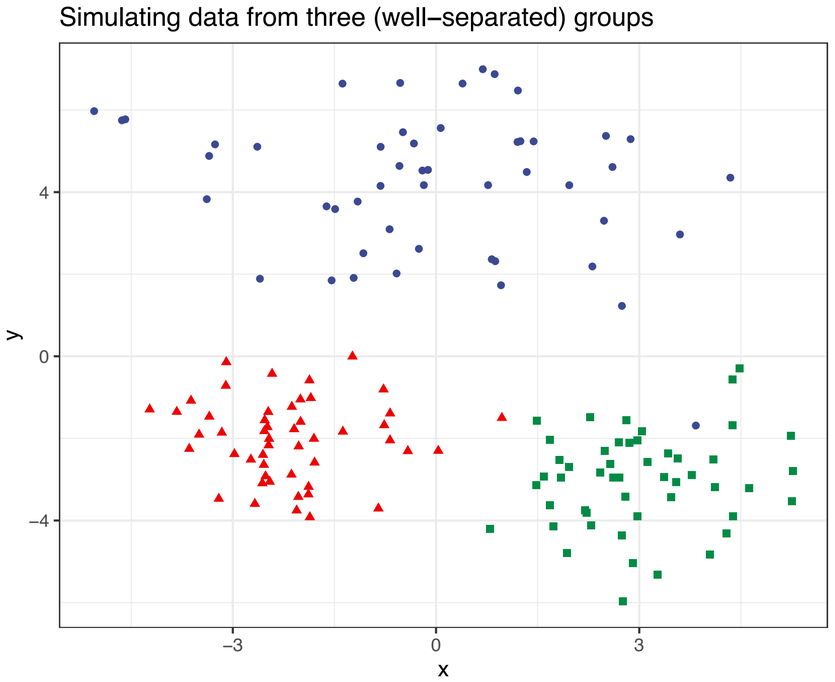
**Different approaches:**

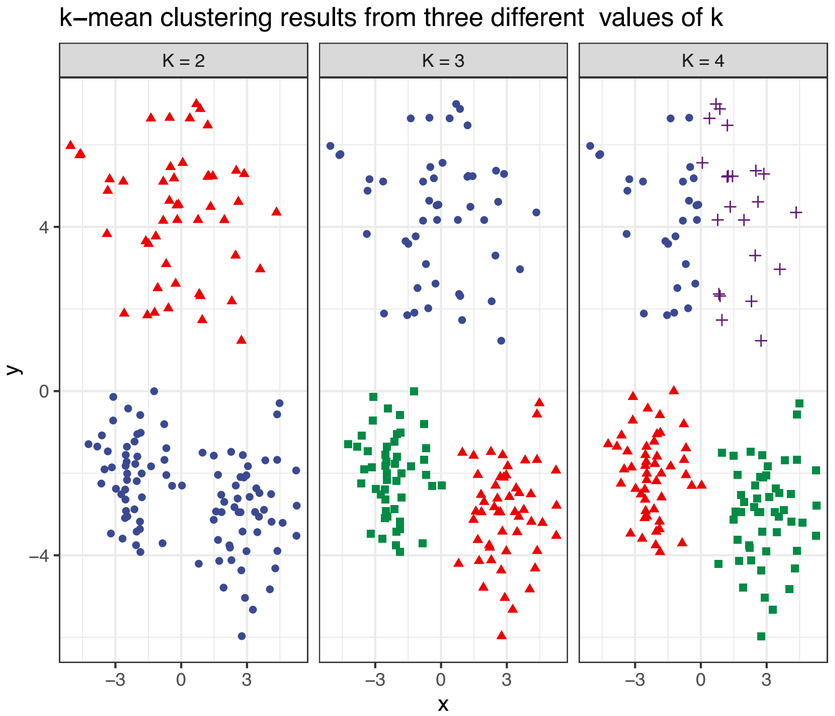
1. **K-means clustering:** partition the observations into a **pre-specified** number of clusters.
2. **Hierarchical clustering:** *do not know in advance how many clusters we want*. Get a tree-like visual representation of the $n$—many observations, called a dendrogram, that allows us to view at once the clusterings obtained for dendrogram each possible number of clusters, from 1 to $n$.

# K-means clustering

## Overview

1. Simple and elegant approach for partitioning a data set into $K$ distinct, non-overlapping clusters.
2. To perform $K$-means clustering, we must first specify the desired number of clusters $K$
3. Then the $K$-means algorithm will assign each observation to exactly one of the $K$ clusters.
4. Good clustering = within-cluster variation is as small as possible.



# K-means clustering

## Defining notation

1. Let $C_1, C_2, \ldots, C_K$ denote sets containing the indices of the observations in each cluster.
2. Each observation belongs to at least one of the $K$ clusters

$$\cup_{i=1}^{K} C_i = \{1,2,\ldots,n\}$$

3. No observation belongs to more than one cluster

$$C_k \cap C_l = \emptyset \text{ for } 1 \leq k \neq l \leq K$$

$i-$th observation is in the $l-$th cluster: $i \in C_l$

Good clustering: within-cluster variation is as small as possible.

Within-cluster variation for cluster $C_l$ = $W(C_l)$

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 2 | **3** | **4** | **5** | *6* | *7* | *8* | 9 | 10 |

$\underset{\sim}{x} = (x_1, x_2, \ldots, x_n)^T$ ; $\mathscr{L}_p$ norm : $\|\underset{\sim}{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$.

# K-means clustering

↳ so, $\|\underset{\sim}{x}\|_p^p = \sum_{i=1}^{n} |x_i|^p$ and $\|\underset{\sim}{x}_i - \underset{\sim}{x}_j\|_2^2 = \sum_{m=1}^{n} (x_{im} - x_{jm})^2$

## Clustering is an optimization problem

1. The within-cluster variation for cluster $C_k$:
   $W(C_k)$ - the amount by which the observations within a cluster differ from each other.
2. "Good clustering" = solving the problem

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

3. Many different ways to define $W(C_k)$….

$$W(C_k) = \frac{1}{n(C_k)} \sum_{i,j \in C_k} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|_0^0$$

$$W(C_k) = \frac{1}{n(C_k)} \sum_{i,j \in C_k} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|_1^1$$

$$W(C_k) = \frac{1}{n(C_k)} \sum_{i,j \in C_k} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|_2^2$$

The general $\mathscr{L}_p$ norm for a vector $\boldsymbol{x} = (x_1, x_2, \ldots x_n)^T$ is $\|\boldsymbol{x}\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}$

① Remember $\bar{x}_n = \text{argmin}_a \sum_{i=1}^{n} (x_i - a)^2$ ("sample mean" = a) "sample mean minimizes mean squared error"



# Optimizing for $\mathscr{L}_2$ norm

## Lloyd's algorithm

② similary, sample median minimizes mean absolute deviation

- Our optimization problem now is:

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{n(C_k)} \sum_{j,j' \in C_k} \| x_j - x_{j'} \|_2^2 \right\}$$

- **Step 1:** randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.
- **Step 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 $\mathscr{L}_2$ norm)

→ will guarantee convergence for $\mathscr{L}_2$ norm

For $\mathscr{L}_1$ norm, need K-MEDIOD

**What is a good $K$?**

This algorithm is guaranteed to decrease the value of the objective

It is not guaranteed to give the global minimum

Expect an assignment problem on this….

How would the algorithm change for other penalties?

(doable for $\mathscr{L}_1$ or $\mathscr{L}_\infty$)
(not an assignment problem)

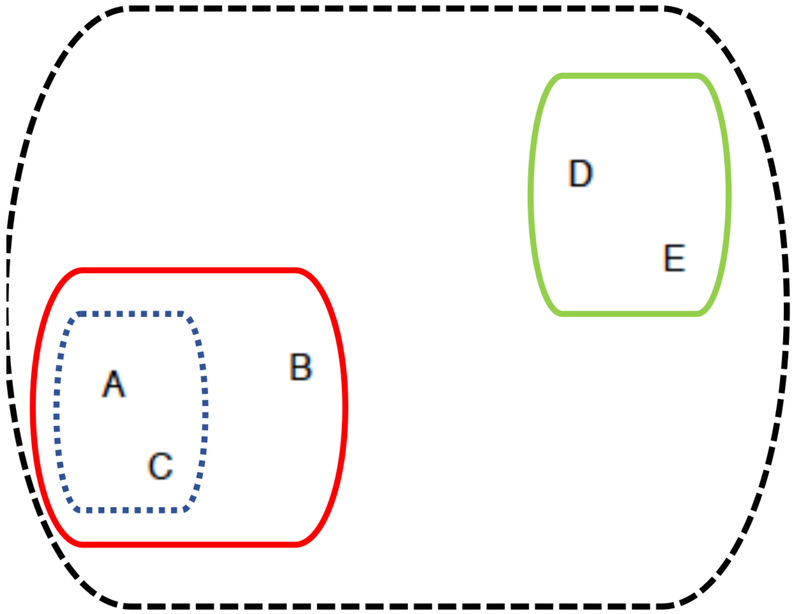
FOR EACH CLUSTER, COMPUTE VECTOR: EACH COMPONENT IS MEDIAN OF FEATURE IN THAT CLUSTER

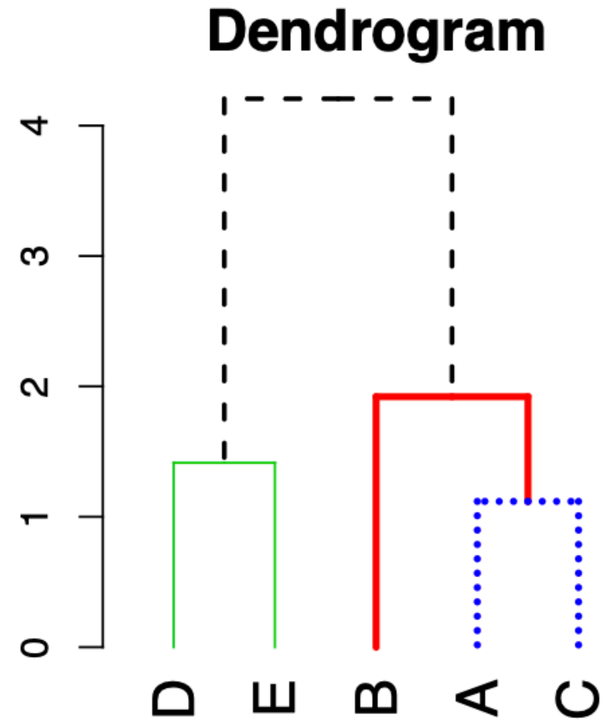
# Hierarchical clustering

## If you can't commit to $K$ just yet…

- Begin with $n$ observations and measure (using $\mathscr{L}_p$ norm) all $n(n-1)/2$ **inter-observation distances**. Treat each observation as it's own cluster.
- For $i = n, n-1, \ldots, 2$ :
  - Examine all pairwise **inter-cluster dissimilarities** among the $i$ clusters and identify the pair of clusters that are **most similar** and fuse them.
  - Dissimilarity between two clusters indicates the height at which in the dendrogram at which the fusion is placed.
  - Compute new pairwise **inter-cluster dissimilarities** among the $i-1$ remaining clusters. Repeat step above.

$$W(C_k) = \frac{1}{n(C_k)} \sum_{i,j \in C_k} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|_2^2$$

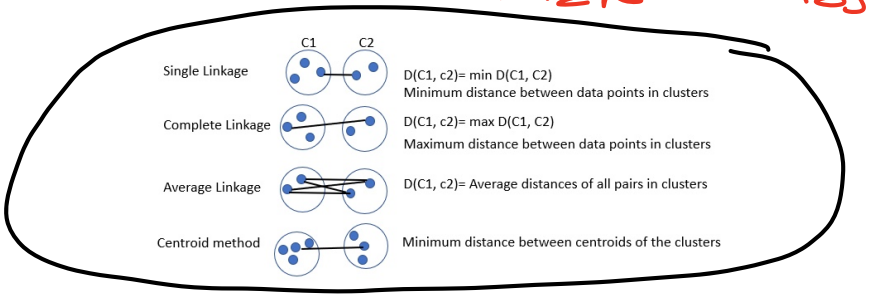


ADDED AFTER CLASS.

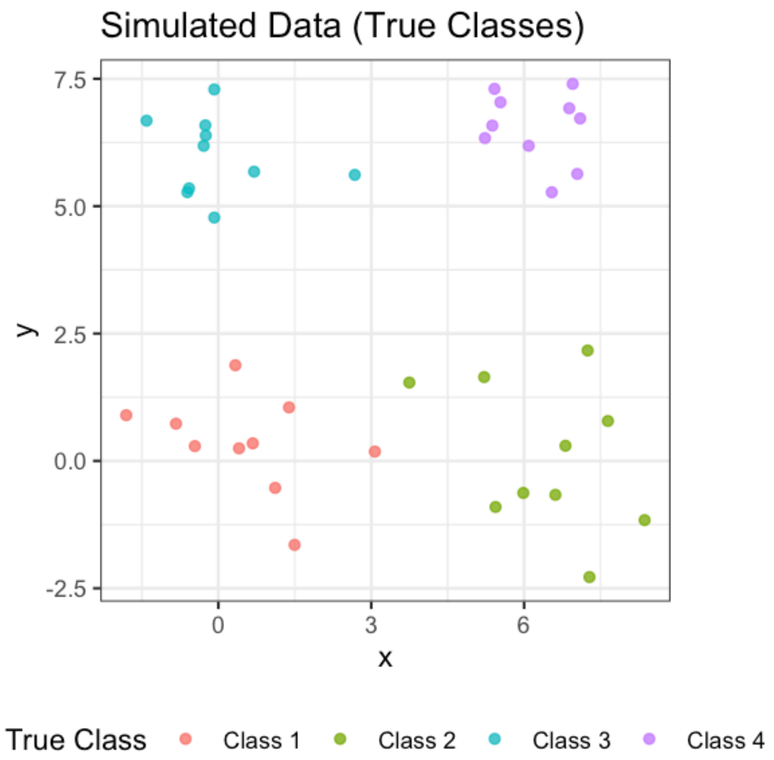
# Hierarchical clustering

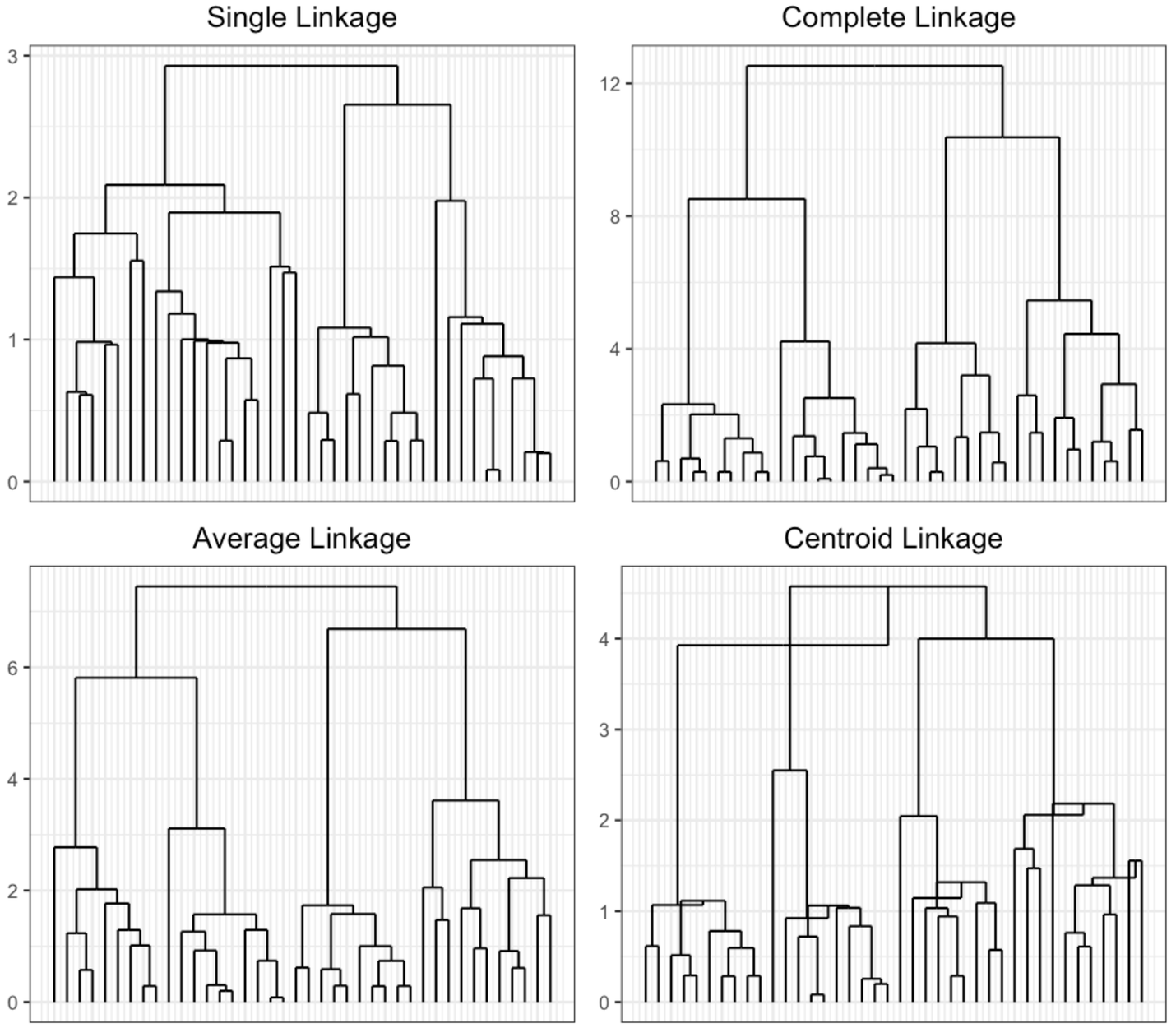
## If you can't commit to $K$ just yet…




- Begin with $n$ observations and measure (using $\mathscr{L}_p$ norm) all $n(n-1)/2$ **inter-observation distances**. Treat each observation as it's own cluster.
- For $i = n, n-1, \ldots, 2$ :
  - Examine all pairwise **inter-cluster dissimilarities** among the $i$ clusters and identify the pair of clusters that are **most similar** and fuse them.
  - Dissimilarity between two clusters indicates the height at which in the dendrogram at which the fusion is placed.
  - Compute new pairwise **inter-cluster dissimilarities** among the $i-1$ remaining clusters. Repeat step above.

| **Dissimilarity type** | Description |
|---|---|
| Complete | Maximal intercluster dissimilarity. Compute all pairwise **distances** between the observations in cluster A and the observations in cluster B, and record the largest of these **distances**. |
| Single | Minimal intercluster dissimilarity. Compute all pairwise **distances** between the observations in cluster A and the observations in cluster B, and record the smallest of these **distances**. |
| Average | Mean intercluster dissimilarity. Compute all pairwise **distances** between the observations in cluster A and the observations in cluster B, and record the average of these **distances**. |
| Centroid | **Distance** between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. |


Simulated Data (True Classes)
7.5
5.0
2.5
0.0
-2.5
y
0
3
6
x
True Class
Class 1
Class 2
Class 3
Class 4


Single Linkage
3
2
1
0
Complete Linkage
12
8
4
0
Average Linkage
6
4
2
0
Centroid Linkage
4
3
2
1
0

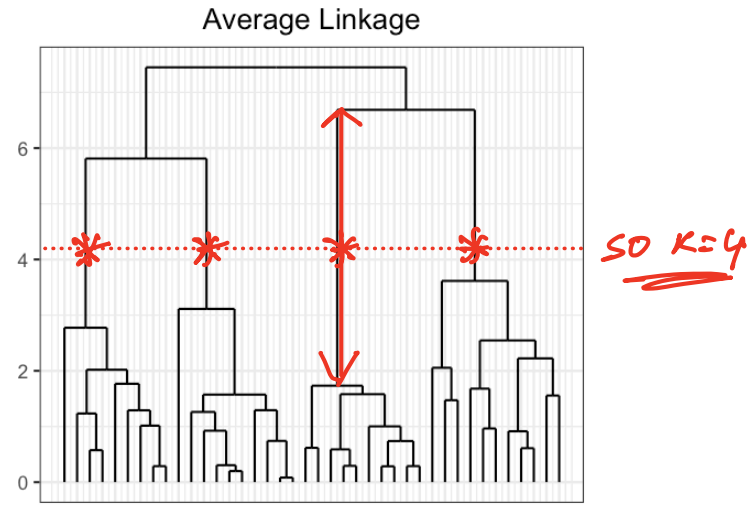
# Hierarchical clustering

## Using dendrograms to commit to $K$

- Look for the Longest Vertical Line: On your dendrogram look for the longest vertical line that doesn't get "crossed" by any other cluster merges (horizontal lines).
- Draw a Horizontal Line: Imagine drawing a horizontal line that cuts right through the middle of that longest vertical segment.
- Count the Intersections: Count how many vertical lines your new horizontal line crosses.

**Scaling matters! Always center and scale!!**

- The Y-axis on the dendrogram represents the distance or "dissimilarity" between the clusters being merged.
- A short vertical line means two clusters that were already "pretty close" got merged.
- A long vertical line represents a "big jump." It means the algorithm had to merge two clusters that were very far apart (i.e., very dissimilar).
- By drawing your horizontal "cut line" below that big jump, we're saying, "not willing to accept any merges between clusters that are **this** different."



# The Gap Statistic

## A structured way of choosing $K$…

**Gap Statistic** measures total within-cluster sum of squares:

$$W_K = \sum_{k=1}^{K} \sum_{i \in C_k} \| \boldsymbol{x}_i - \boldsymbol{\mu}_k \|_2^2$$, where $\boldsymbol{\mu}_k$ is the mean vector for $C_k$

We compute $\log(W_K)$ from **real data** and compare with $\mathbb{E}^\star\left(\log(W_K)\right)$, which measures behavior of $W_K$ under '**null reference data**'

For $k = 1,2,\ldots, K_{\max}$:

- Cluster observed data $\mathscr{D}$ with $k$ clusters and compute $\log(W_k)$.
- Generate $B$ "null" reference datasets by sampling uniformly from the bounding box of $\mathscr{D}$: $\mathscr{D}_1^\star, \mathscr{D}_2^\star, \ldots, \mathscr{D}_B^\star$
  - Get $\log(W_{k,b}^\star)$, $1 \le b \le B$
  - Estimate $\hat{\mathbb{E}}^\star\left(\log(W_k)\right) = B^{-1} \sum_{b=1}^{B} \log(W_{k,b}^\star)$
  - $\text{Gap}_k = \hat{\mathbb{E}}^\star\left(\log(W_k)\right) - \log(W_k)$
- Choose value of $k$ that maximizes $\text{Gap}_k$

The crux of the gap statistic and **permutation tests** are very similar: you set up a "contest" between two datasets:

- Observed data
- Simulated null data: create a fake dataset with no clusters (uniform random sample). Run the exact same k-means algorithm on this fake data. This gives you a "baseline clustering" score.

# Post-class debrief

## Today we discussed…

The fundamental goal of clustering is to discover subgroups within data. We want to partition observations into distinct groups so that items within the same group are very similar to each other, while items in different groups are very dissimilar. This is a core technique in unsupervised learning and has many applications, such as market segmentation, grouping genes with similar expression patterns, and organizing images.

- **K-Means Clustering:** This method partitions the dataset into a pre-specified number of K distinct, non-overlapping clusters by trying to make the within-cluster variation as small as possible.
  - Advantages: It is a simple and elegant approach to partitioning a dataset.
  - Disadvantages: Must pre-specify the number of clusters, K, which is often unknown. Further, the algorithm is not guaranteed to find the single best (global minimum) solution and can get stuck in a sub-optimal local minimum.

- **Hierarchical Clustering:** This method does not require specification of the number of clusters in advance; instead, it builds a tree-like visual representation called a dendrogram that shows the cluster merges at every level.
  - Advantages: Its main advantage is that it does not require pre-specified K. It produces a dendrogram, which is a rich visualization that depicts see the clustering structure at all levels and can be used to select a good K.
  - Disadvantages: The choice of "linkage" (e.g., complete, single, average) can significantly affect the results. The algorithm is more computationally intensive, as it must compute all pairwise distances between observations.

- The Gap Statistic is a formal, structured method for choosing the optimal number of clusters, K, for an algorithm like k-means.
- Intuition: It answers the question, "How much better is the clustering than the clustering we get on 'null' data with no structure?"


That's all Folks!