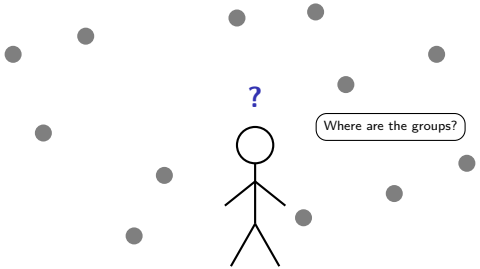
Cluster Analysis: A Visual Guide

A Formula-Free Introduction

Statistical Data Analysis

Lesson 4

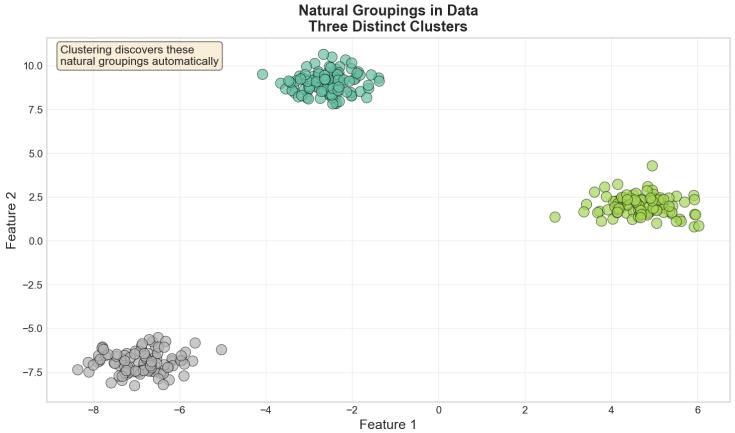
Can You Find the Groups?



Data arrives without labels – can you spot the natural groups?

Clustering = finding structure in unlabeled data.

Do Groups Really Exist in Data?



- Data often contains natural groupings that are invisible until you look
- Clustering algorithms find these groups without any labels or supervision

Unsupervised learning discovers hidden structure automatically.

What Will You Learn Today?

1. **Split data into groups** with k-Means – the magnet method
2. **Build group hierarchies** with dendrograms – the family tree
3. **Choose the right number of groups** – the elbow test

No formulas. Just pictures, analogies, and intuition.

Three acts: k-Means, hierarchical clustering, and choosing k.

What Is k-Means in One Sentence?

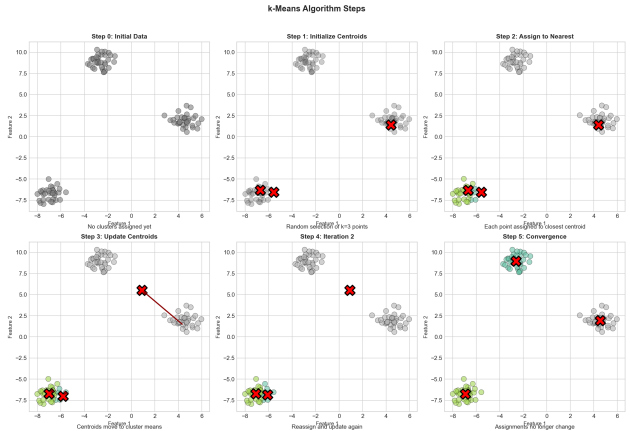
k-Means = "Place magnets, let data snap to the nearest one."

Three everyday analogies:

- **Magnets on a whiteboard** – place k magnets; each data point sticks to the closest one
- **Sorting laundry by color** – toss each item into the nearest color pile
- **Seating guests at wedding tables** – assign each guest to the nearest table, then adjust tables

k-Means assigns every point to its nearest center, then moves centers to the group average.

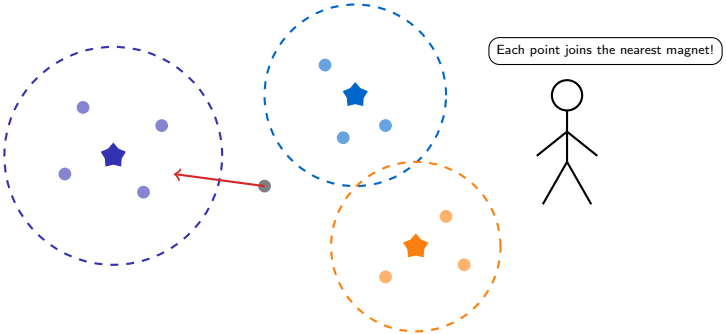
How Does k-Means Work Step by Step?



- Place k random centers in the data space
- Assign each point to its nearest center
- Move each center to the average of its assigned points – repeat until stable

The algorithm converges when no point changes its assignment.

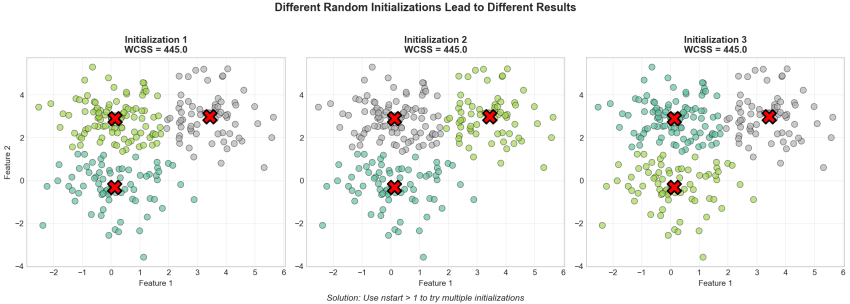
Why Is It Like a Magnet Game?



k-Means keeps moving the magnets until every point is happy.

Stars = centroids. Dashed circles = cluster boundaries. Arrows = reassignment.

Does the Starting Position Matter?



- A bad starting position can trap k-Means in a wrong answer
- Run the algorithm multiple times and pick the best result

Modern implementations use k-Means++ to choose smarter starting positions.

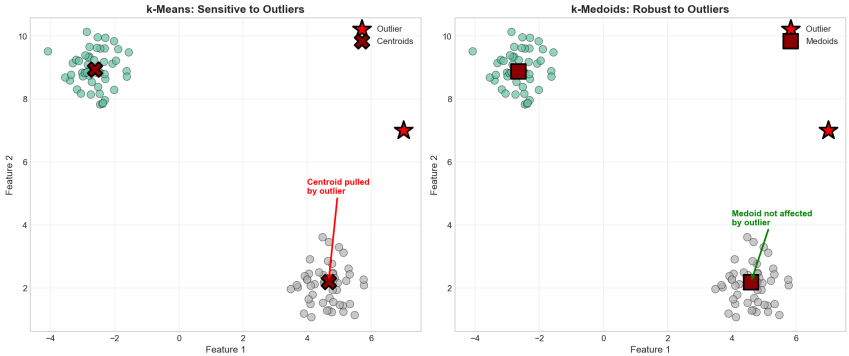
How Do You Pick the Right k?

- You must choose k **before** running the algorithm – it will not tell you how many groups exist
- **Too few** groups (k too small) = oversimplified, real differences get hidden
- **Too many** groups (k too large) = noise disguised as structure

We will see systematic tools for choosing k in Act 4.

Choosing k is the hardest part of k-Means – there is no single right answer.

What Happens When Outliers Appear?



- A single extreme point can drag a centroid far off center
- This is k-Means' biggest weakness – it treats every point equally

k-Medoids uses actual data points as centers, making it robust to outliers.

What If You Do Not Know k?

- k-Means demands that you pick k upfront – but what if you have no idea?
- **Hierarchical clustering** does not need k at all
- It builds a tree of nested groups and lets you cut at any height

Think of it as growing a family tree from the data – you decide where to cut the branches.

Hierarchical clustering trades speed for flexibility: no k needed upfront.

How Does Bottom-Up Merging Work?

- **Start**: every data point is its own tiny cluster
- **Merge**: find the two closest clusters and combine them into one
- **Repeat**: keep merging until everything belongs to a single big cluster

The result is a tree called a **dendrogram** *– a map of every merge that happened.*

Agglomerative (bottom-up) clustering is the most common hierarchical approach.

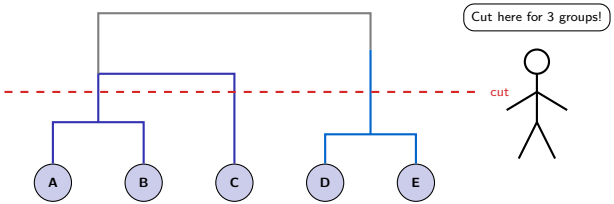
What Are the Merging Steps?

- **Step 1:** Find the two closest clusters in the data
- **Step 2:** Merge them into one new cluster
- **Step 3:** Update all distances and repeat – every merge adds a branch to the tree

After $n - 1$ merges, you have a complete dendrogram with n leaves.

Each horizontal bar in a dendrogram records exactly one merge event.

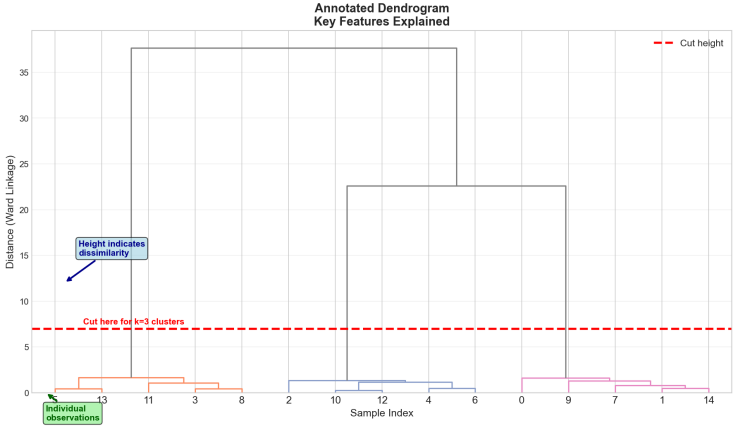
Can You Build a Family Tree of Data?



Merge from the bottom, cut where you want your groups.

A dendrogram lets you choose the number of clusters after building the tree.

How Do You Read a Dendrogram?



- **Height** of a bar = the distance at which two groups were merged
- A **horizontal cut line** splits the tree into clusters
- The number of vertical lines crossing the cut = the number of clusters

Tall bars mean big jumps – a natural place to cut.

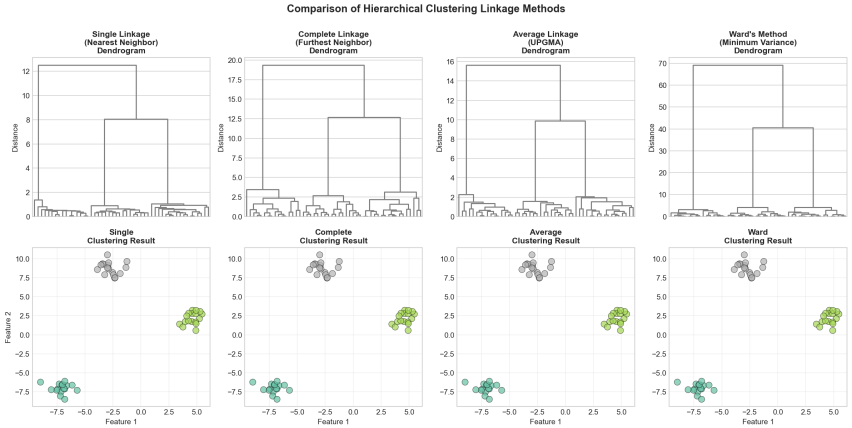
Does It Matter How You Measure "Closest"?

- **Single linkage** = nearest-neighbor distance (can produce long chains)
- **Complete linkage** = farthest-neighbor distance (produces compact clusters)
- **Ward's method** = minimizes the increase in within-cluster spread (most popular)

Same data, different linkage rule = very different tree shapes.

Ward's linkage tends to give the most balanced, evenly-sized clusters.

See How Linkage Changes the Tree!



- Same data, four different trees – linkage choice matters
- Ward's method usually gives the most balanced result

Always try more than one linkage method and compare the trees.

When Should You Use Hierarchical Clustering?

- When you **do not know** k in advance and want to explore different cuts
- When you want a **visual tree** of relationships (e.g., gene expression, taxonomy)
- When your dataset is **not too large** – hierarchical methods scale poorly beyond a few thousand points

For large datasets, use k-Means first, then hierarchical on the centroids.

Hierarchical clustering shines when the tree structure itself is informative.

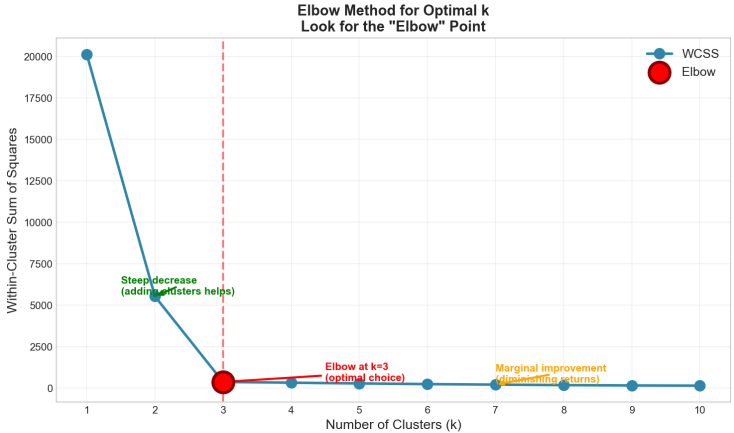
How Do You Know the Right Number of Groups?

- There is **no single perfect answer** – clustering is an art as much as a science
- Use visual diagnostics: the **elbow plot** and **silhouette scores**
- Combine evidence from multiple methods before deciding

If two methods agree on the same k, you can be more confident.

Never trust a single metric – always cross-validate with domain knowledge.

Where Is the Elbow?



- The x-axis shows the number of clusters, the y-axis shows total within-cluster distance
- The "elbow" is where adding more clusters stops helping much
- Pick the k at the bend – diminishing returns beyond that point

The elbow method is simple but sometimes the bend is not obvious.

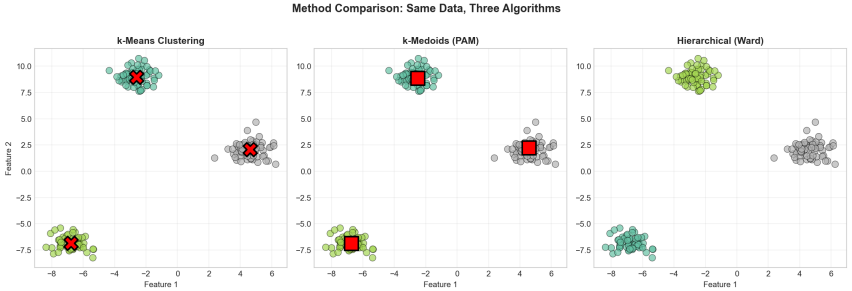
Which Method Should You Choose?

- k-**Means** – fast, works well for round clusters, needs k upfront
- **Hierarchical** – no k needed, gives a tree view, but slow on large data
- k-**Medoids** – like k-Means but robust to outliers (uses real data points as centers)

No single method wins everywhere – match the method to your data shape.

Start with k-Means for speed, switch to alternatives when assumptions break.

Compare All Methods at a Glance!



- Each method has strengths and weaknesses – the table summarizes them
- No single method wins everywhere; non-round clusters need special care
- Match the method to your data shape and research question

In practice, try 2–3 methods and compare results before reporting.

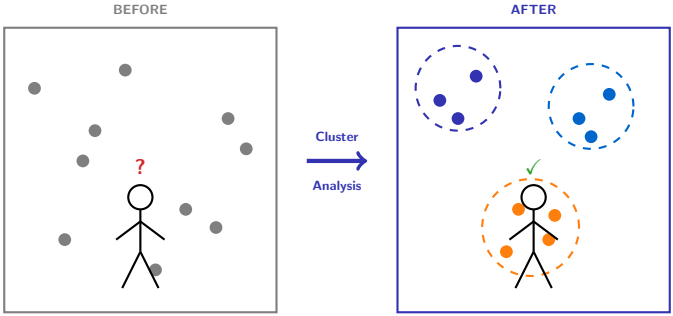
Remember the Three Big Takeaways!

1. *k*-**Means** is fast but needs *k* upfront and assumes round clusters
2. **Hierarchical clustering** builds a tree – cut where you want your groups
3. **Always validate** with elbow or silhouette before trusting your clusters

Clustering finds structure – but only you can judge whether it makes sense.

Combine algorithmic evidence with domain knowledge for the best results.

Can You See the Groups Now?



From scattered dots to clear groups – that is cluster analysis.

Clustering turns chaos into clarity – but always validate your groups.

Where Do You Go From Here?

- Try k-Means on a real dataset: `kmeans(data, centers = 3)` in R
- Experiment with dendrograms to explore group structure visually
- Read the full technical lecture for distance metrics, formal validation, and advanced methods

The best way to learn clustering is to cluster something.

Practice with real data – start simple, add complexity as you gain confidence.