GENERAL ASSEMBLY

HIERARCHICAL CLUSTERING

Aymeric Flaisler
Adapted from Joseph Nelson, GA - DSI - DC

AGENDA

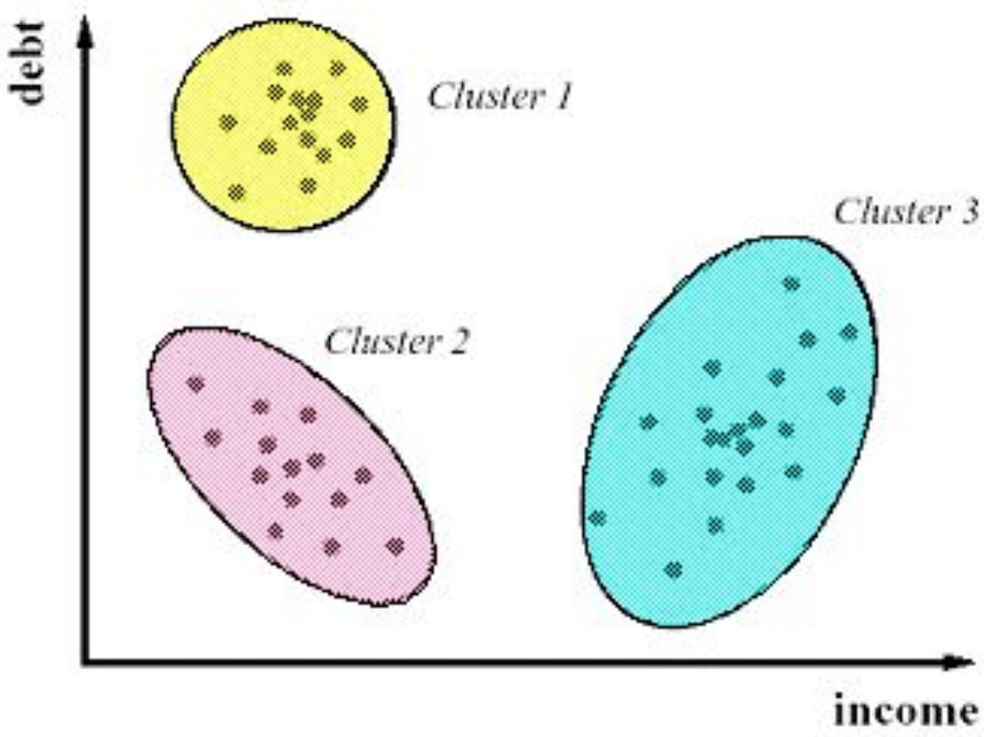
- What is hierarchical clustering?
- How does hierarchical clustering work?
- Code Implementation

WHAT IS HIERARCHICAL CLUSTERING?

- Review: what is clustering?

WHAT IS CLUSTERING?

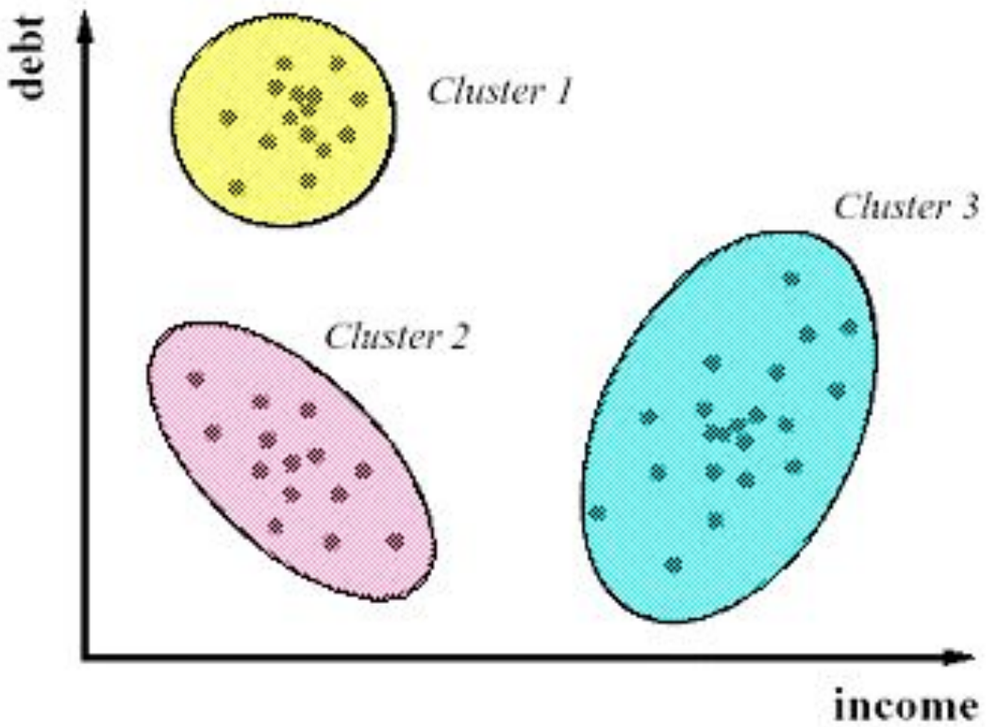
- Review: what is clustering?
- Clustering is an **unsupervised learning** technique we employ to group “similar” data points together
- We use the features space to understand the similarities between data points
- With unsupervised learning, remember: there is no **clear** objective, there is no “right answer” (hard to tell how we’re doing), there is no response variable, just observations with features, and labeled data is not required



WHAT IS K-MEANS CLUSTERING?

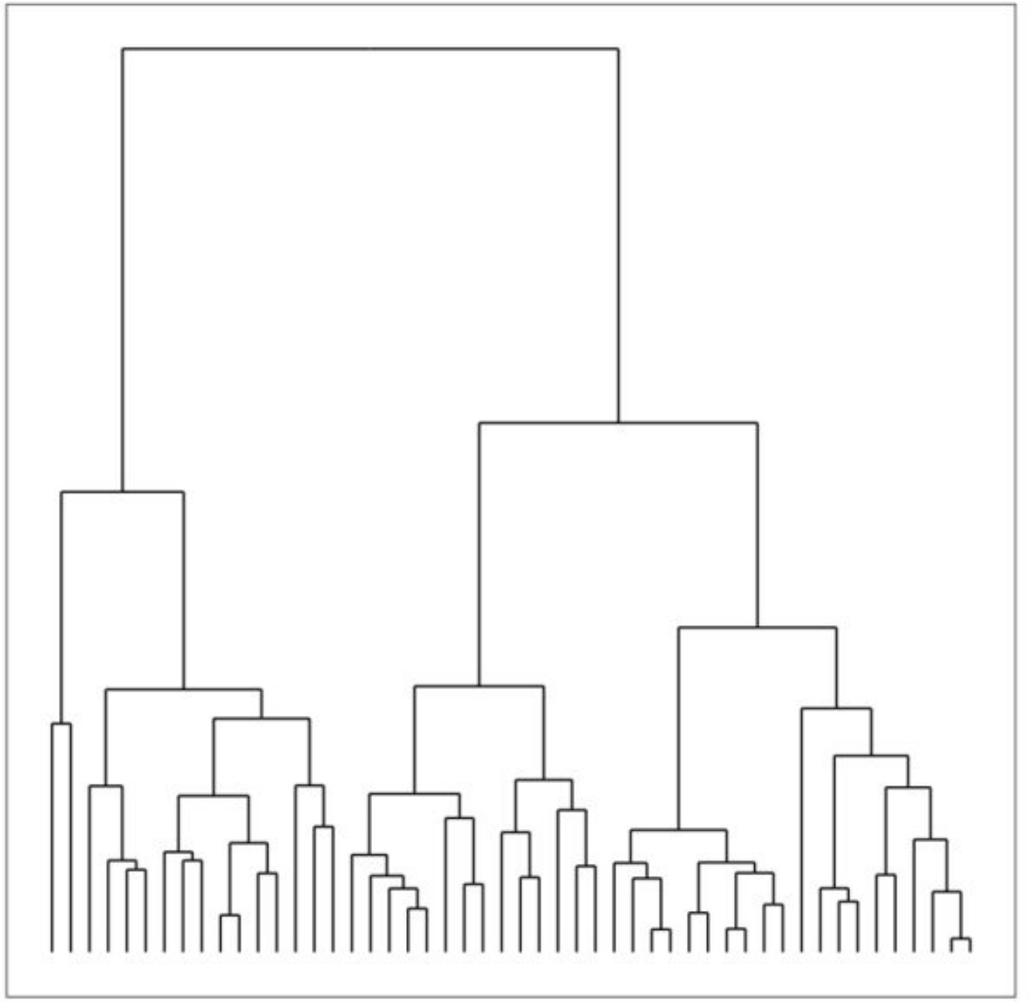
- What's KMeans?
- How does it work?
- What are its weaknesses?

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/



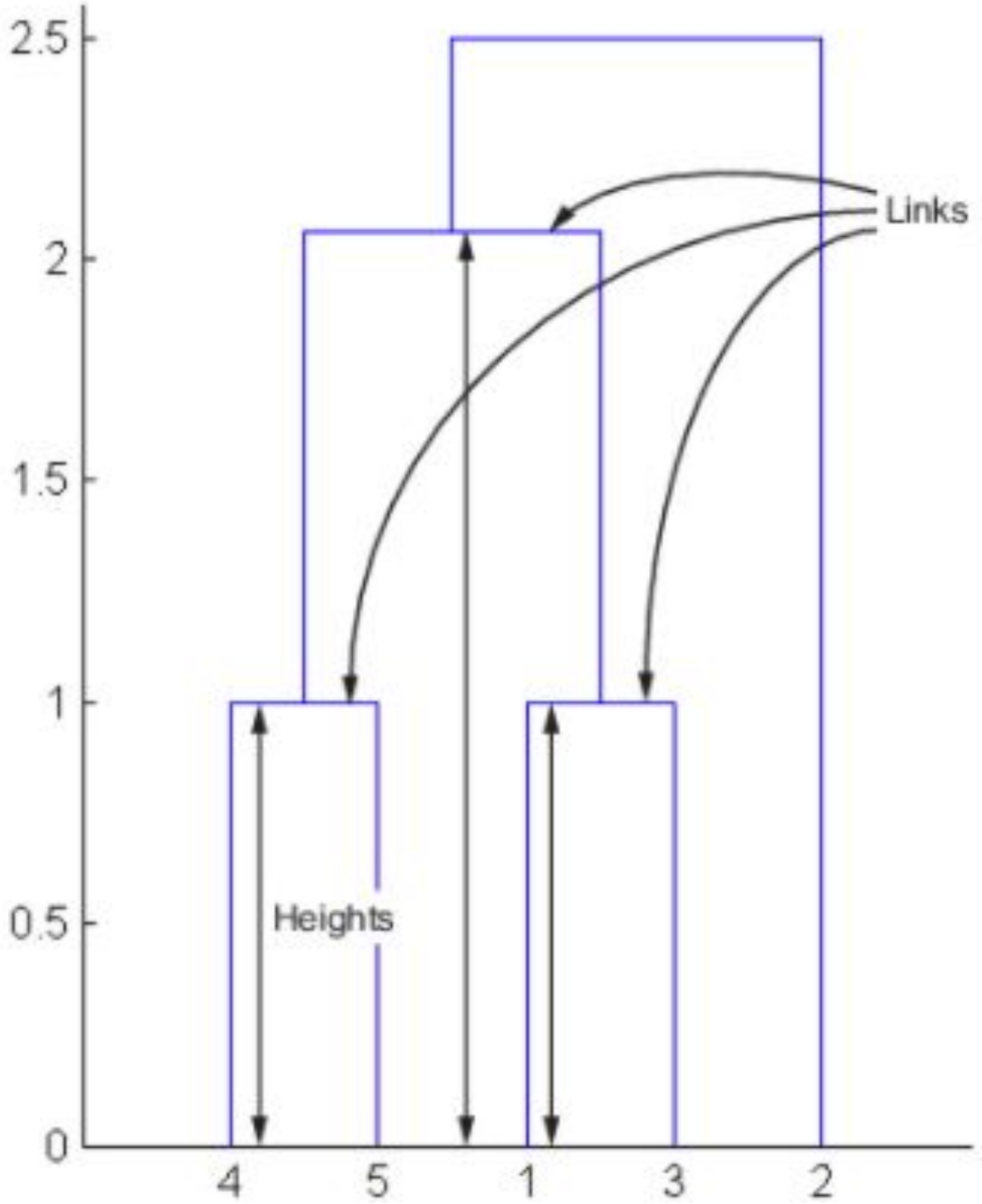
WHAT IS HIERARCHICAL CLUSTERING?

- Hierarchical clustering, like k-means clustering, is another common form of **clustering analysis**. With this type of clustering - we seek to do exactly what the name suggests: **build hierarchies of links** that ultimately form clusters.
- Once these links are determined, they are displayed in what is called a **dendrogram** - a graph that displays all of these links in a hierarchical manner.



WHAT IS HIERARCHICAL CLUSTERING?

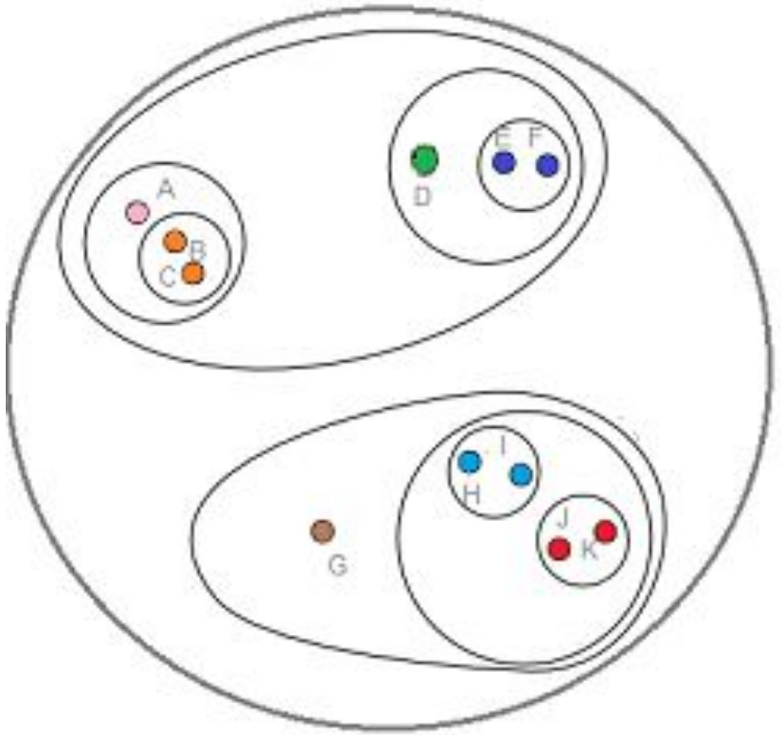
- A **dendrogram** is a branching diagram that represents the relationships of similarity among a group of entities
- The arrangement of the clades tells us which **leaves are most similar** to each other. The **height of the branch** points indicates how similar or different they are from each other: the **greater the height, the greater the difference**
- Because of this, hierarchical clustering is best served on **SMALLER** datasets

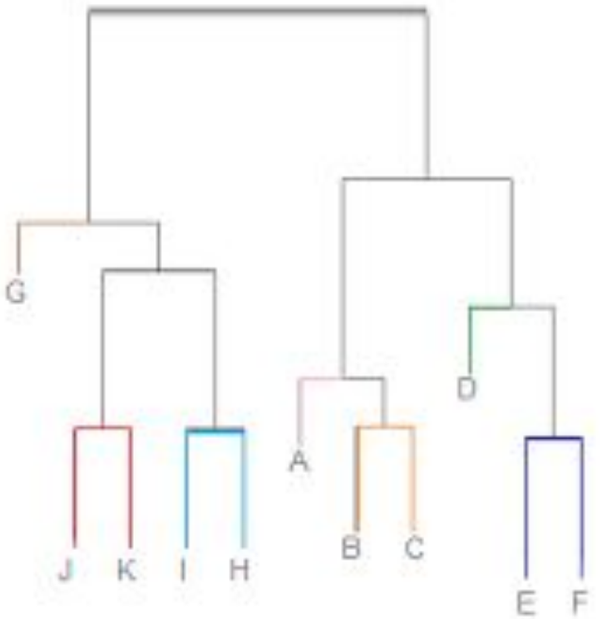


HOW DOES HIERARCHICAL CLUSTERING WORK?

- In hierarchical clustering, the clusters are determined in the a varying number of partitions. At each step, **it makes the best choice based on the surrounding datapoints**, with the ultimate goal that these best choices will lead to the best choice of clusters overall.
- Given the algorithm's method of calculating linkages based on immediate datapoints, it's known as a **greedy algorithm**.

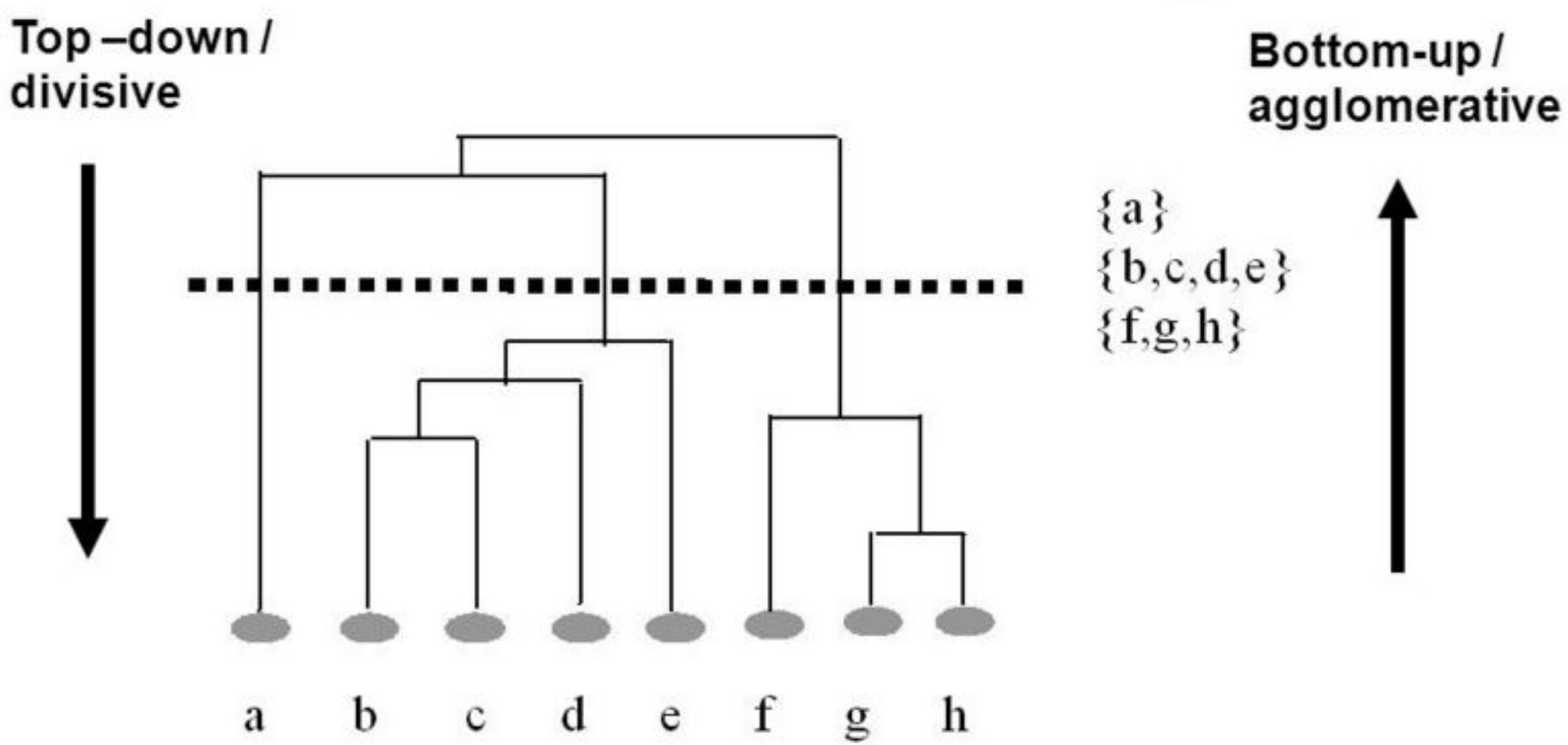
- **Checkout (in pair):** how is that different from the K-means algorithm?





HOW DOES HIERARCHICAL CLUSTERING WORK?

- There are two forms of hierarchical clustering; **agglomerative hierarchical** clustering and **divisive hierarchica**l clustering.



HOW DOES HIERARCHICAL CLUSTERING WORK?

- We're going to look at one of the most fundamental methods for agglomerative hierarchical cluster, known as **linkage clustering**. Linkage clustering iterates through datapoints and computes the distance between groups by computing the distance between two neighboring datapoints, using a distance metric.
- To think about the difference between **agglomerative vs divisive**, with the former we start with the leaves of the tree and build the trunk, and with the latter we start with the trunk of the tree and build the leaves. Both methods are applicable when using hierarchical clustering, it's just a matter of computational preference

IMPLEMENTATION

- Implementing hierarchical clustering in python is as simple as calling a function from the SciPy toolbox:
- `Z = linkage(X, 'ward')`
- Here, "X" represents the matrix of data that we are clustering, and "ward" tells our algorithm which method to use to calculate distance between our newly formed clusters - in this case **Ward's Method** which seeks to minimize the variance when forming clusters. When calculating distance, the default is **Euclidean distance** which we learned about in our dimensionality reduction lesson.

SELECTING K (CREATING CLUSTER)

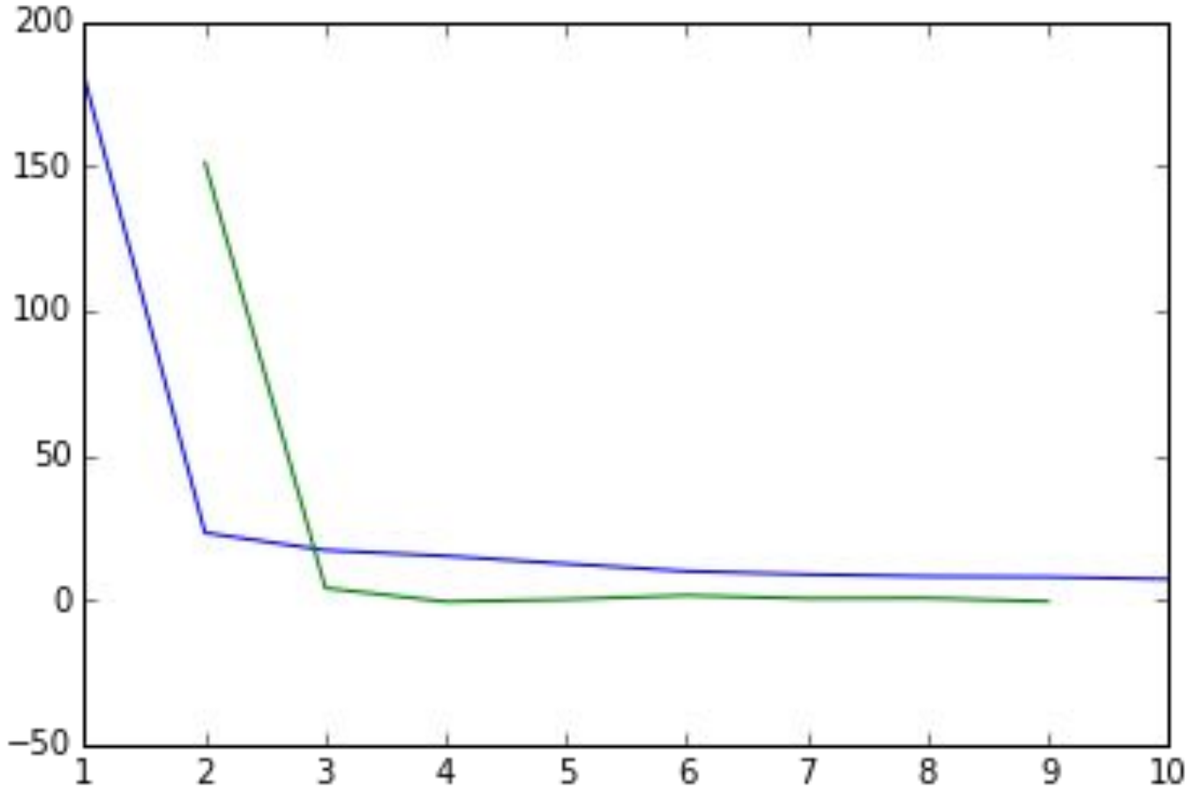
Find Natural Divisions in Data

- The hierarchical cluster tree naturally divide the data into distinct, well-separated clusters. This can be particularly evident in a dendrogram diagram created from data where groups of objects are densely packed in certain areas and not in others.
- The **inconsistency coefficient** of the links in the cluster tree can identify these divisions where the similarities between objects change abruptly (the inconsistency compares each cluster merge's height h to the average avg and normalize it by the standard deviation std formed over the depth previous levels).

$$inconsistency = \frac{h - avg}{std}$$

SELECTING K (CREATING CLUSTER)

- For the **inconsistency coefficient** we want the **clustering step** where the **acceleration of distance growth** is the **biggest**. In other word we are going to look for the highest delta in the distances as K decreases



- Blue line: the inconsistency values
- Green line. the acceleration (or delta)

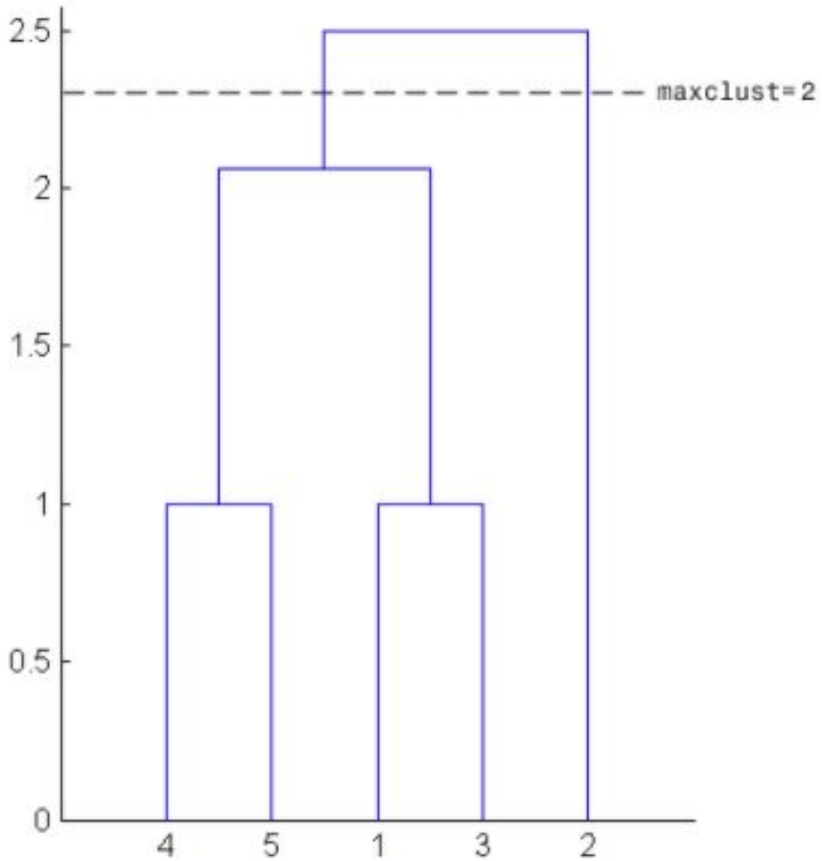
- Result ?

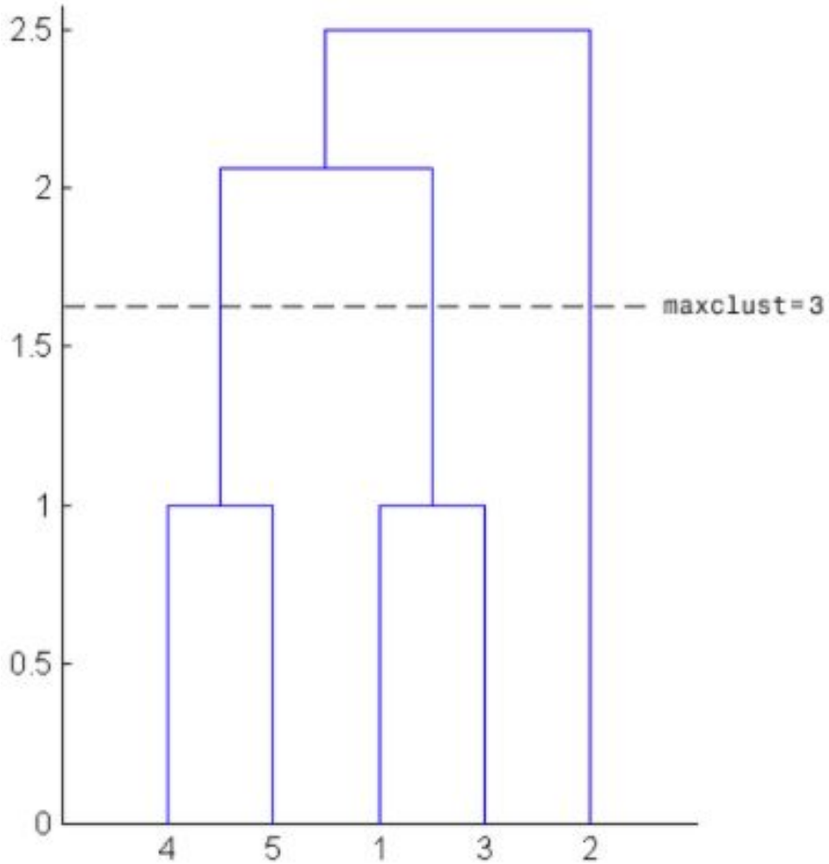
SELECTING K (CREATING CLUSTER)

- The final decision on the K number of cluster is - just like the for the K-means algorithm - a mix of the elbow method with the **inconsistency coefficient, a visual analysis of the distributions and an understanding of your data.**

SELECTING K (CREATING CLUSTER)

- Once from those 3 methods you select a **threshold** which will be the level where you separate your data.





IMPLEMENTATION

- After we cluster, we can calculate the dendrogram using a simple *dendrogram*() function from SciPy, which we can then draw using our handy plt from matplotlib.
- To check how well our algorithm has measured distance, we can calculate the **cophenetic correlation coefficient.** This metric, which measures the height of the dendrogram at the point where two branches merge, can tell us how well the dendrogram has measured the distance between data points in the original dataset and is a helpful measure to see how well our clustering test has run.
- `c, coph_dists = cophenet(Z, pdist(X))`