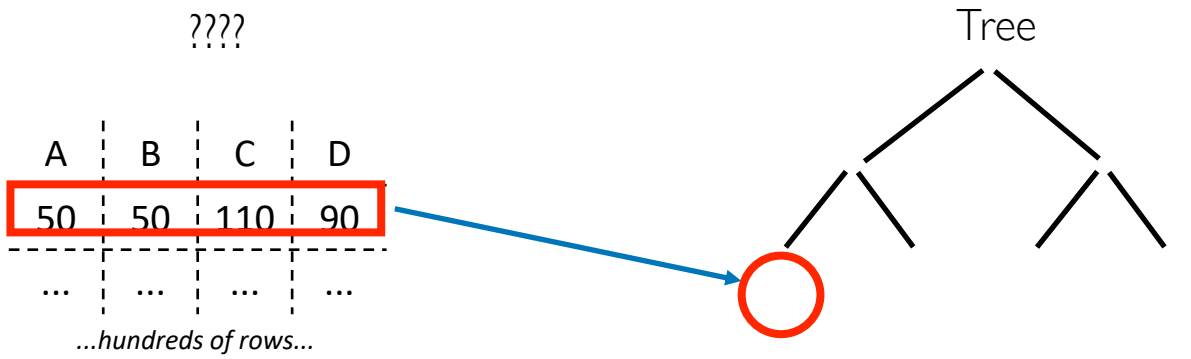
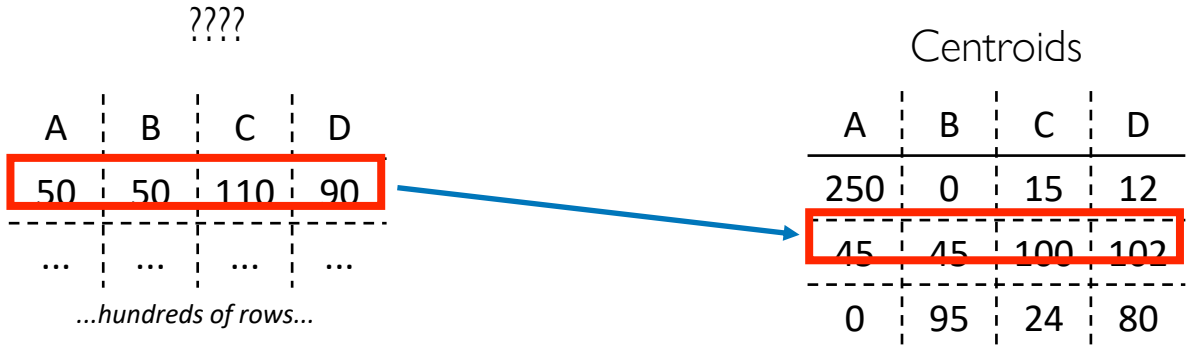
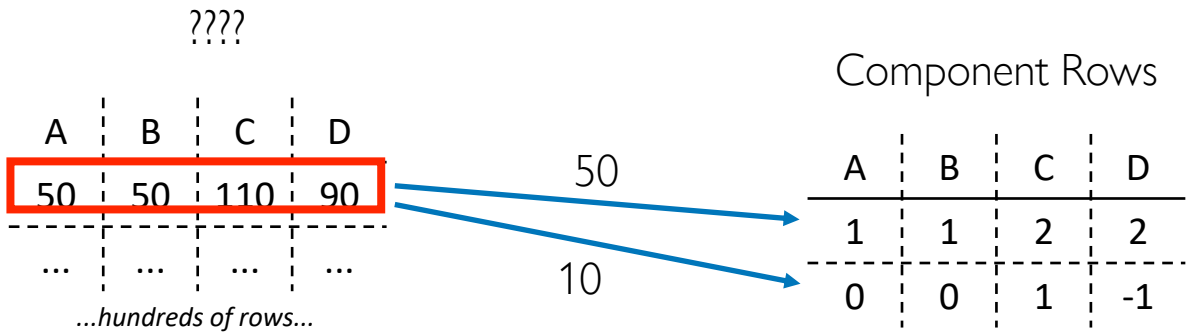
# [320] Unsupervised ML Recap

Department of Computer Sciences
University of Wisconsin-Madison


????
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
Tree


????
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
Centroids
A B C D
250 0 15 12
45 45 100 102
0 95 24 80

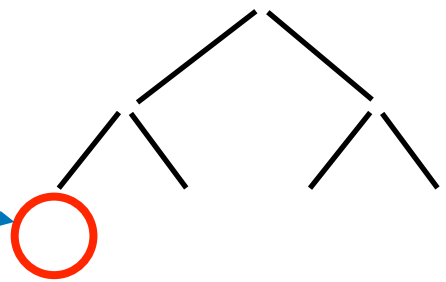

????
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
50
10
Component Rows
A B C D
1 1 2 2
0 0 1 -1

## Hierarchical Clustering
(for example, AgglomerativeClustering)

| A | B | C | D |
|---|---|---|---|
| 50 | 50 | 110 | 90 |
| ... | ... | ... | ... |

*...hundreds of rows...*

Tree



## Non-Hierarchical Clustering
(for example, KMeans)

| A | B | C | D |
|---|---|---|---|
| 50 | 50 | 110 | 90 |
| ... | ... | ... | ... |

*...hundreds of rows...*

Centroids

| A | B | C | D |
|---|---|---|---|
| 250 | 0 | 15 | 12 |
| 45 | 45 | 100 | 102 |
| 0 | 95 | 24 | 80 |

## Decomposition
(for example, PCA)

| A | B | C | D |
|---|---|---|---|
| 50 | 50 | 110 | 90 |
| ... | ... | ... | ... |

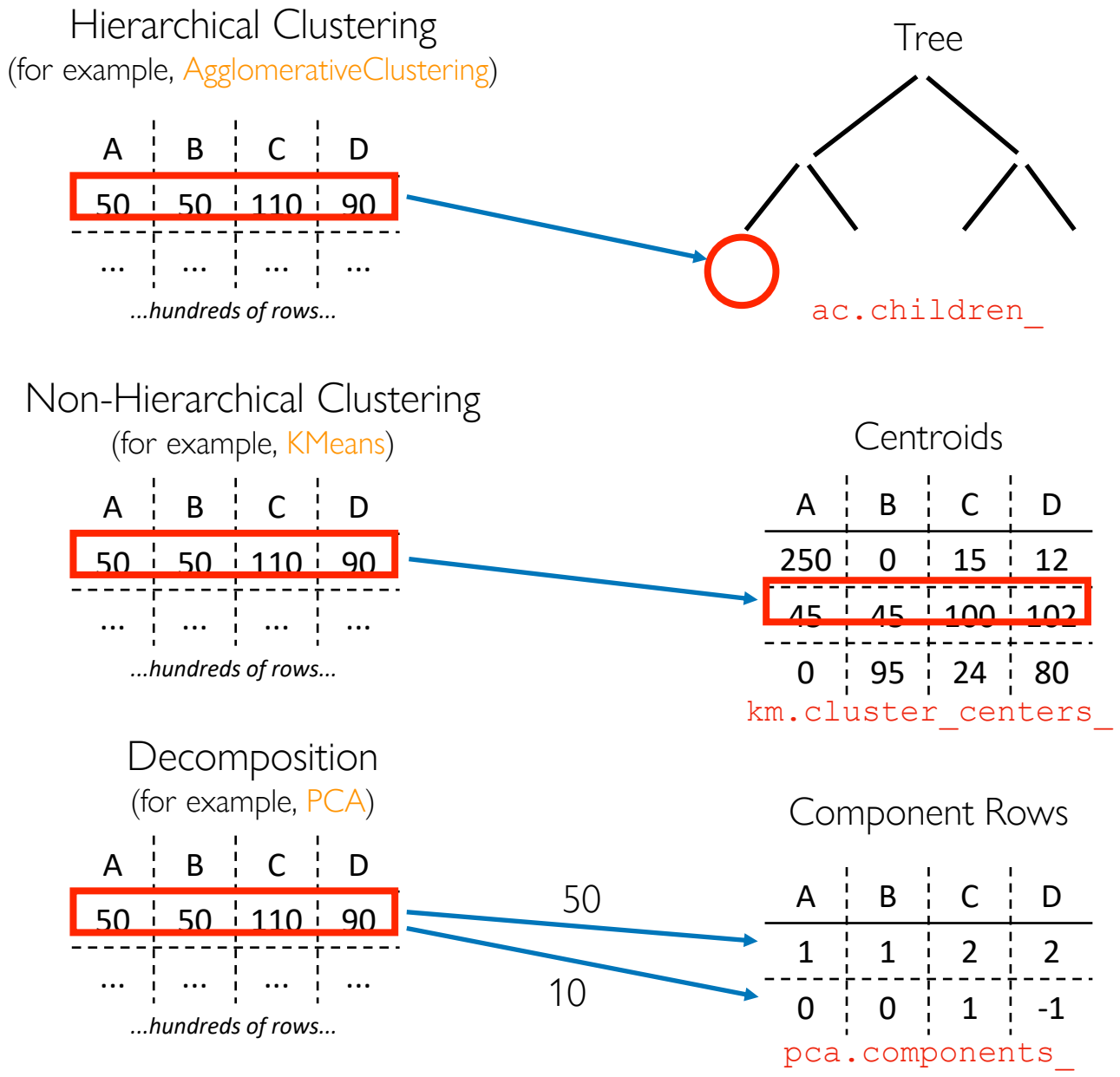
*...hundreds of rows...*

50

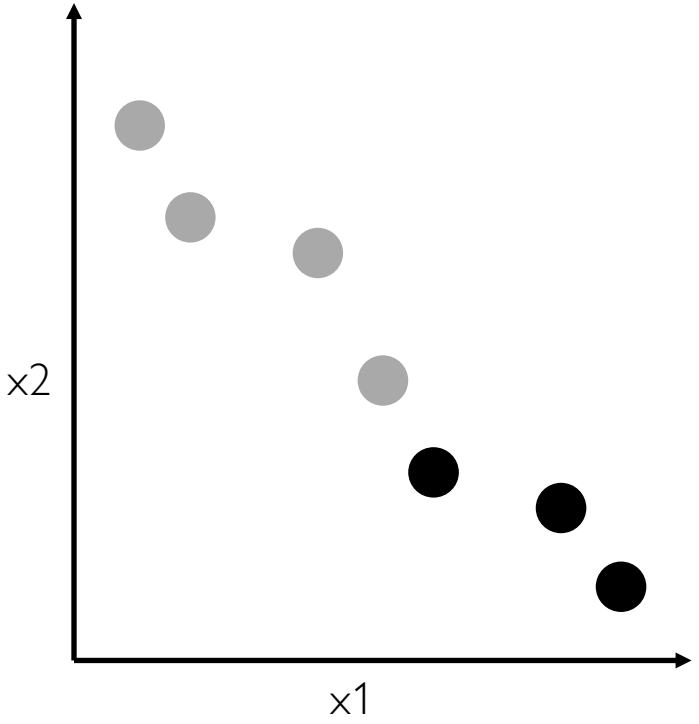
10

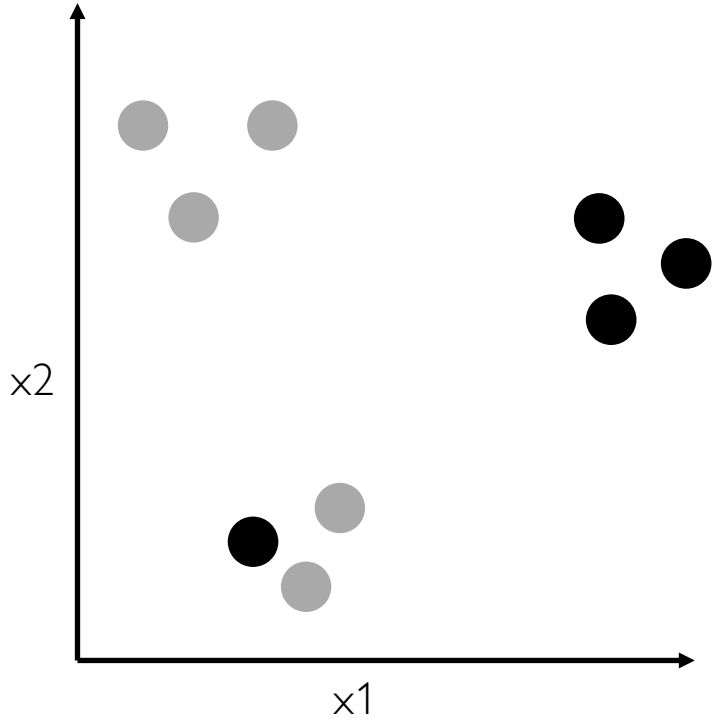
Component Rows

| A | B | C | D |
|---|---|---|---|
| 1 | 1 | 2 | 2 |
| 0 | 0 | 1 | -1 |


Hierarchical Clustering
(for example, AgglomerativeClustering)
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
Tree
ac.children_
Non-Hierarchical Clustering
(for example, KMeans)
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
Centroids
A B C D
250 0 15 12
45 45 100 102
0 95 24 80
km.cluster_centers_
Decomposition
(for example, PCA)
A B C D
50 50 110 90
... ... ... ...
...hundreds of rows...
50
10
Component Rows
A B C D
1 1 2 2
0 0 1 -1
pca.components_

# Preprocessing: Clustering or Decomposition?

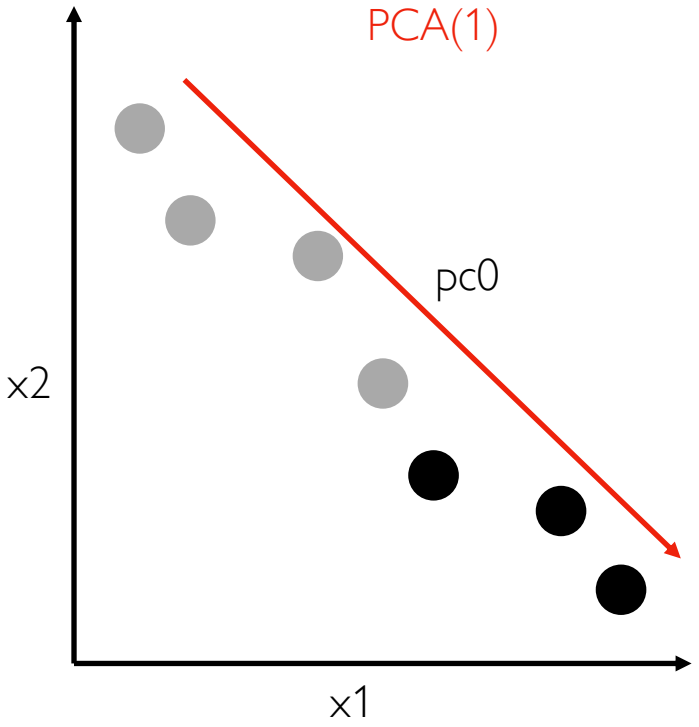


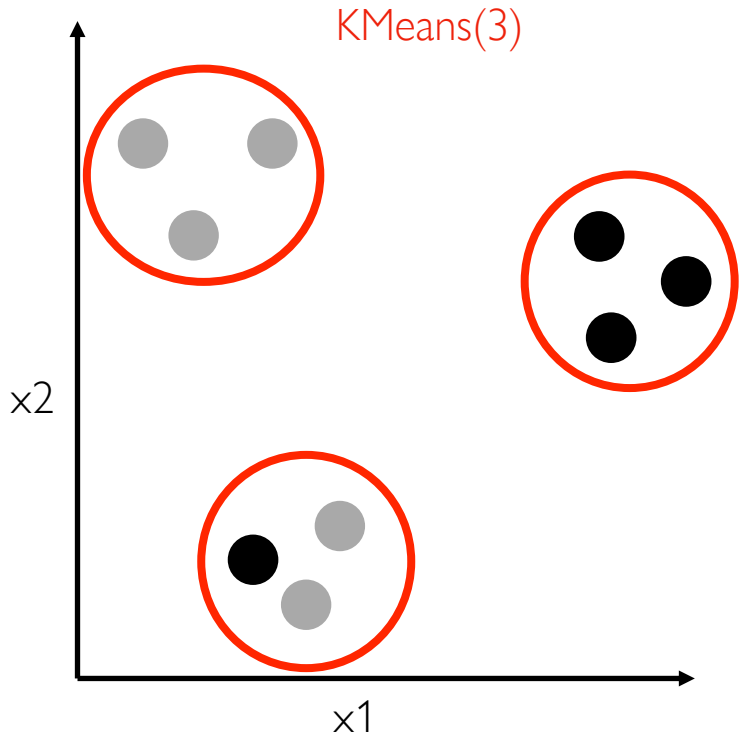




| x1 | x2 | y |
|---|---|---|
| 10 | 5 | TRUE |
| … | … | … |

```
model = Pipeline([
    ????,
    ("lr", LogisticRegression())
])
```

# Preprocessing: Clustering or Decomposition?

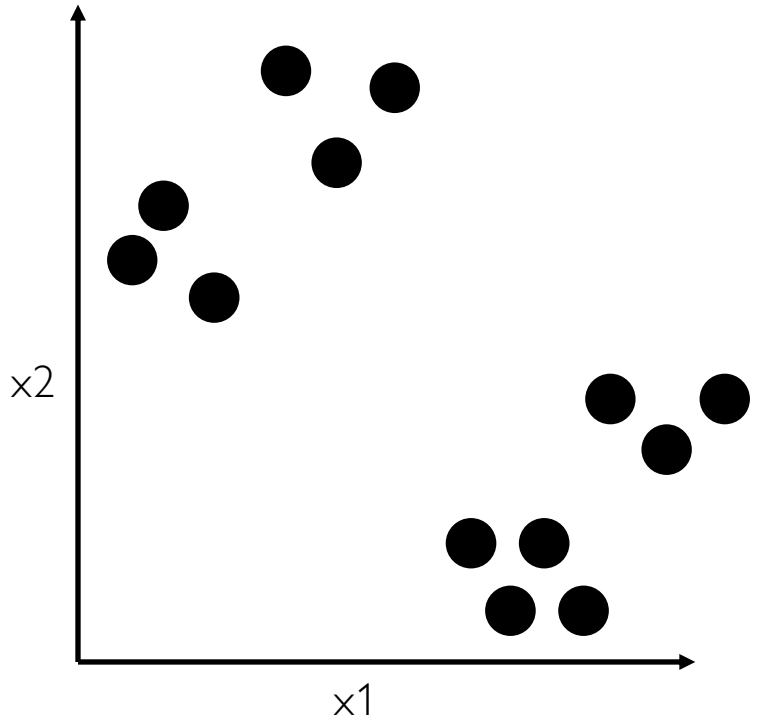


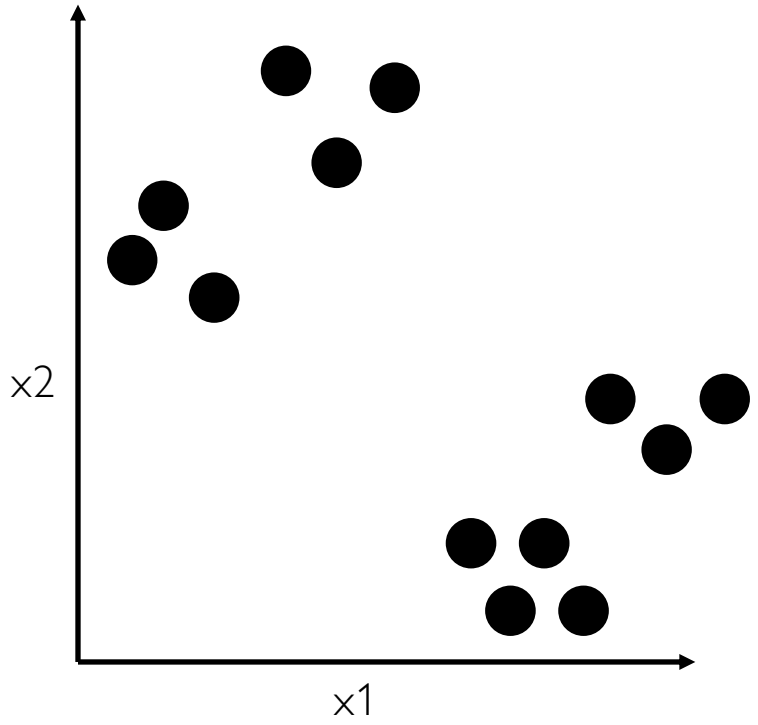

| x1 | x2 | y |
|---|---|---|
| 10 | 5 | TRUE |
| … | … | … |

```
model = Pipeline([
    ????,
    ("lr", LogisticRegression())
])
```

# KMeans or Agglomerative Clustering?

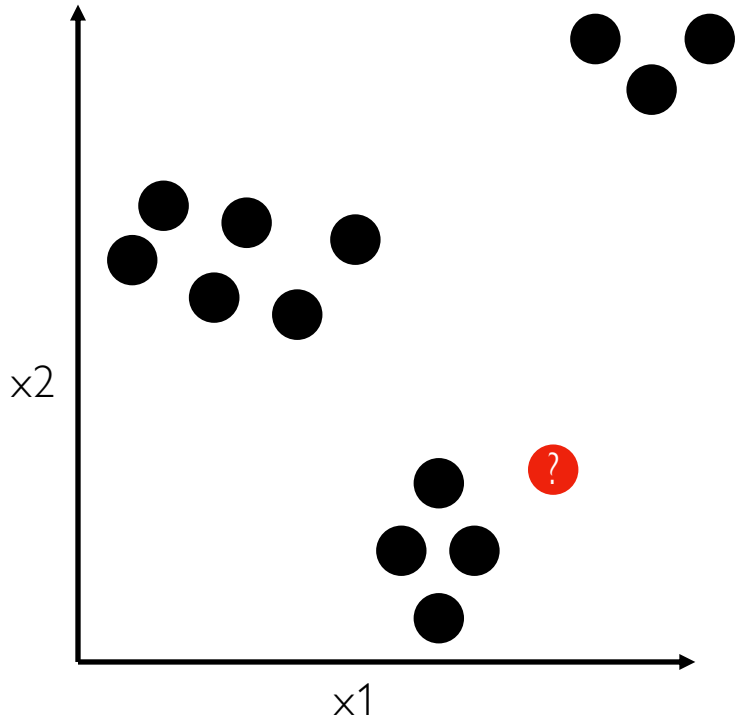


# KMeans or Agglomerative Clustering?



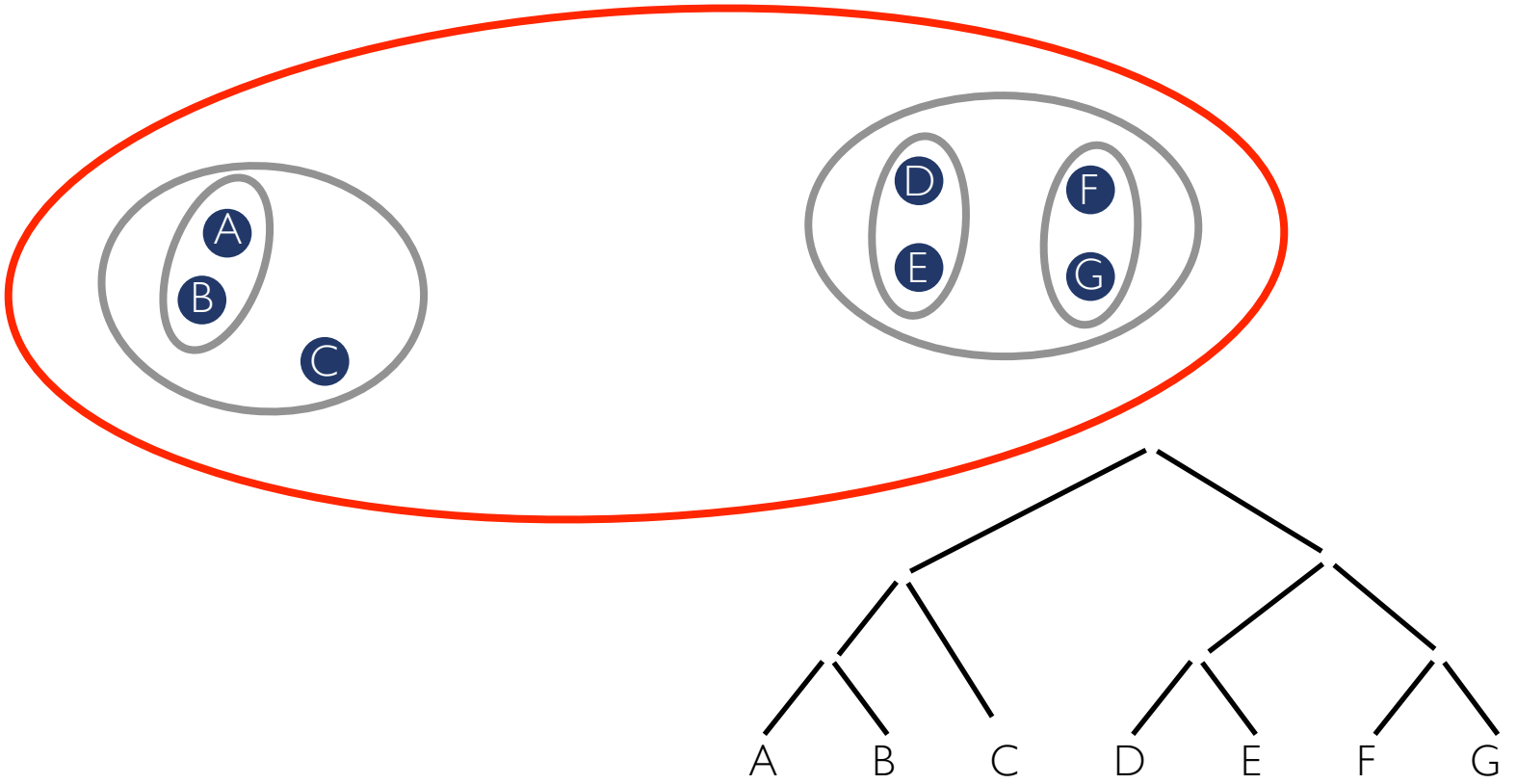

**AgglomerativeClustering** can show us that the two big clusters contain sub clusters.

# KMeans or Agglomerative Clustering?

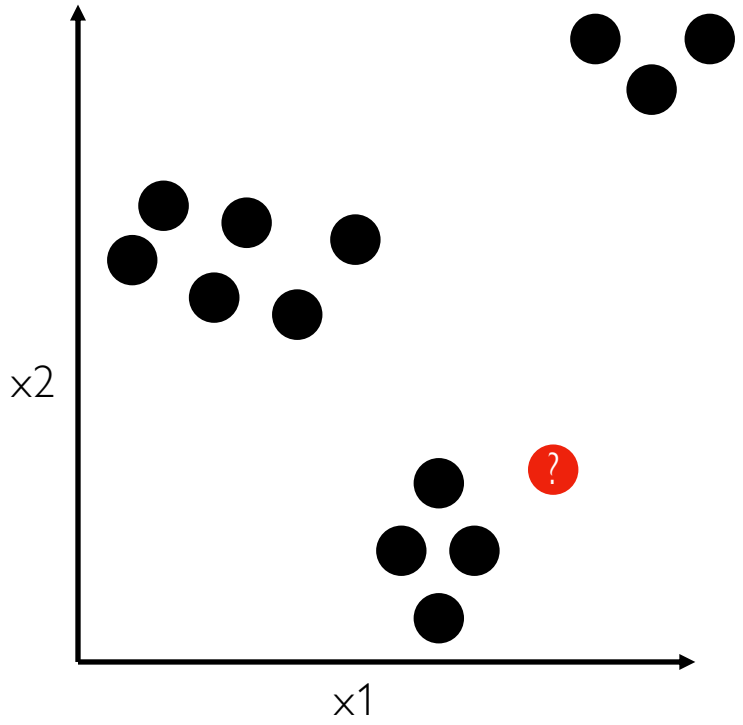



After identifying some clusters from initial data, we will need to look at new data points and find what cluster is the best match

# Strategy: Combine Nearby Points/Groups (and repeat!)



# KMeans or Agglomerative Clustering?




Use **KMeans**, because it can do fit and predict on separate datasets.
AgglomerativeClustering can only do fit_predict on a single dataset.

Non-hierarchical clusters cannot contain other custers (example: KMeans)

Hierarchical clusters can contain other custers (example: AgglomerativeClustering)

# Hierarchical clusters: AgglomerativeClustering

# Strategy: Combine Nearby Points/Groups (and repeat!)

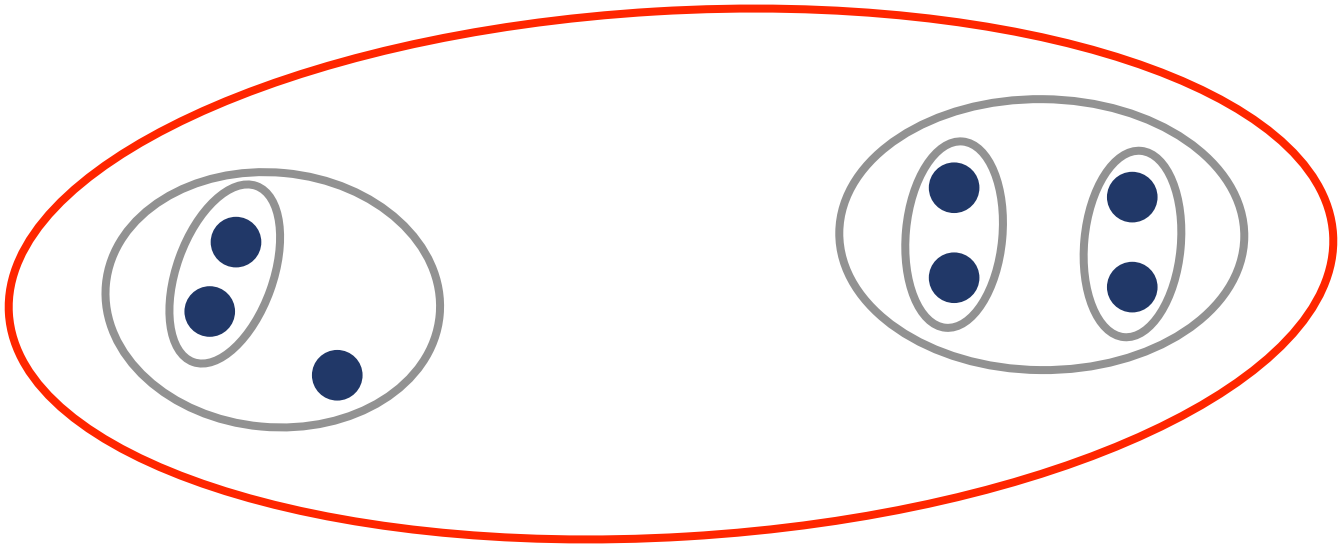
# Strategy: Combine Nearby Points/Groups (and repeat!)

# Strategy: Combine Nearby Points/Groups (and repeat!)

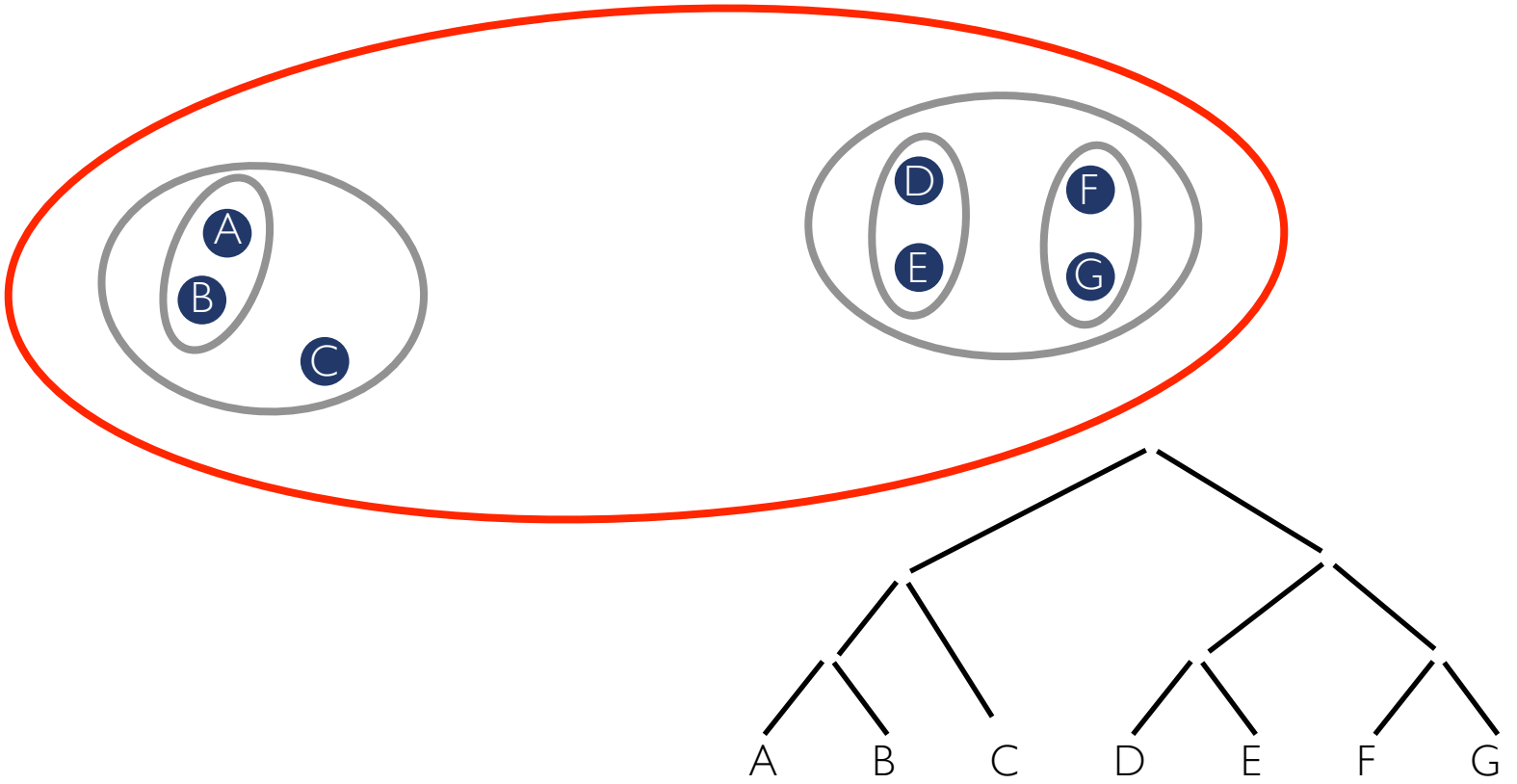
# Strategy: Combine Nearby Points/Groups (and repeat!)

# Strategy: Combine Nearby Points/Groups (and repeat!)

# Strategy: Combine Nearby Points/Groups (and repeat!)



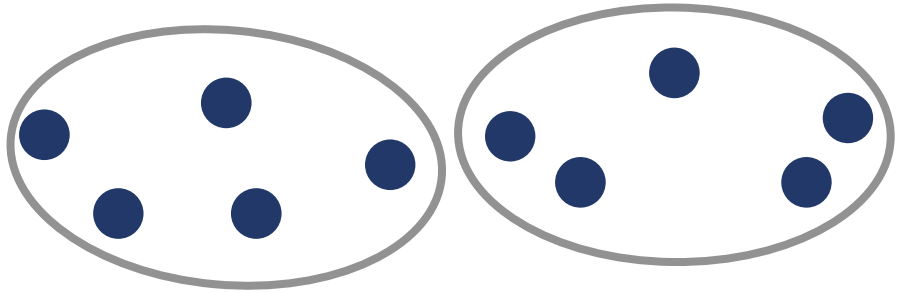
# Strategy: Combine Nearby Points/Groups (and repeat!)



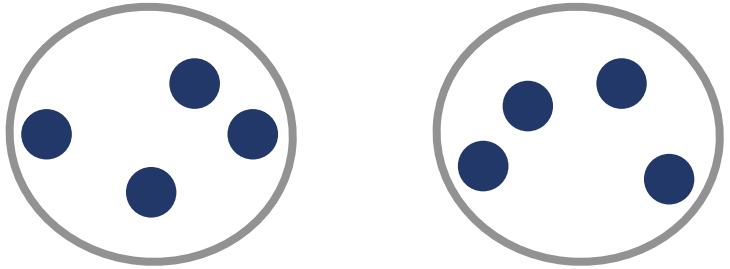
# Configuration: what is "nearest"?

option: `linkage`

# Configuration: what is "nearest"?

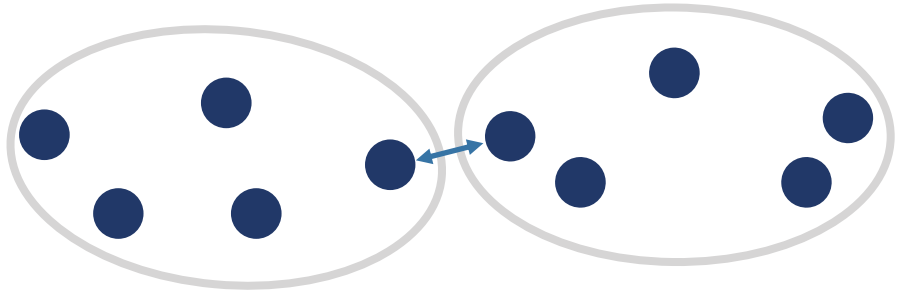


OR...

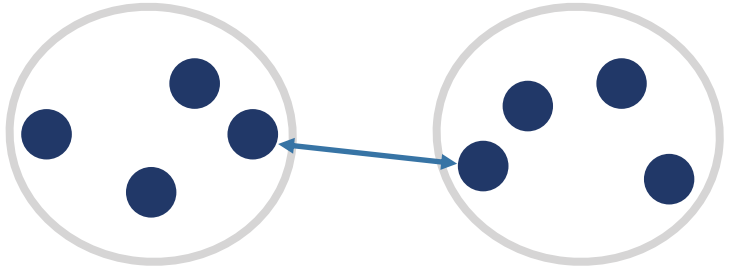


# Configuration: what is "nearest"?

linkage="single"

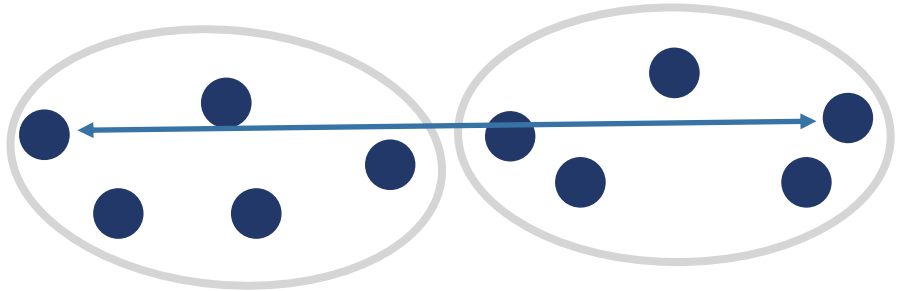


OR...

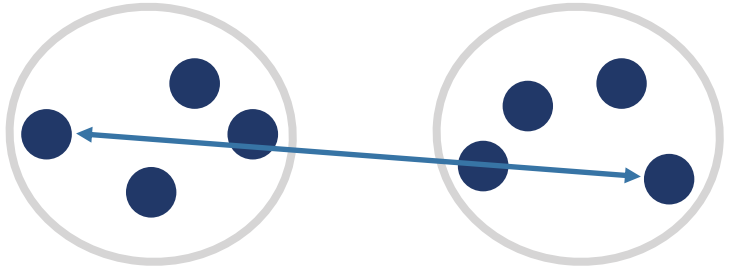


# Configuration: what is "nearest"?

linkage="complete"



OR...



# Configuration: what is "nearest"?

linkage="????"

From docs: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

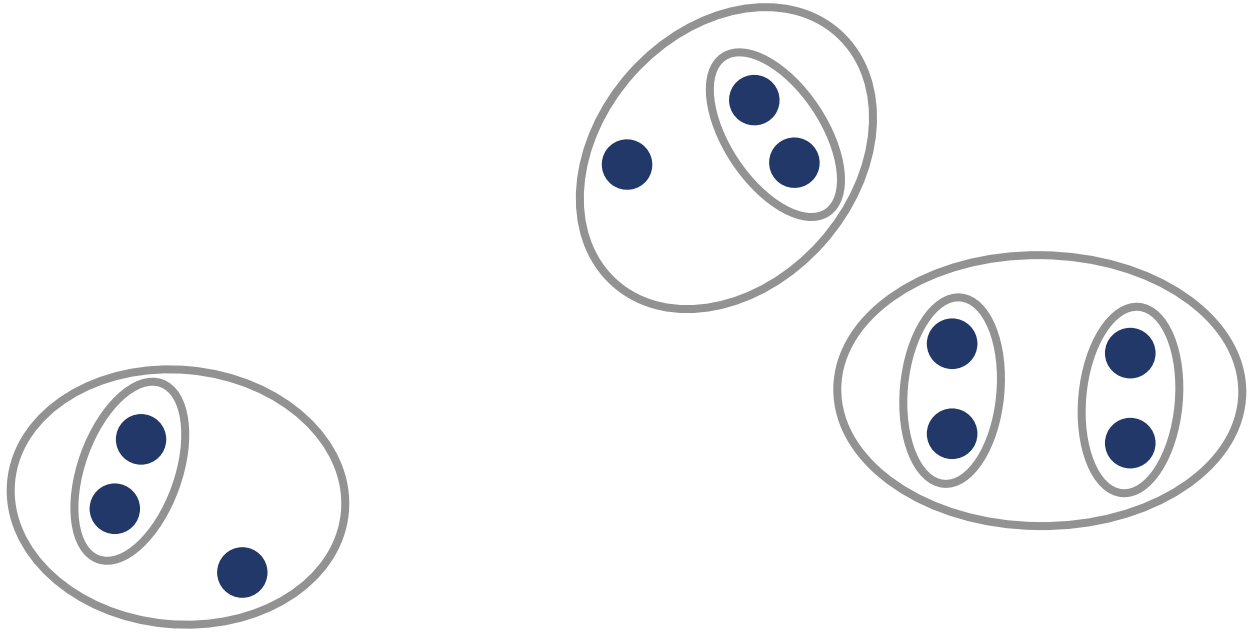
- ward minimizes the variance of the clusters being merged.
- average uses the average of the distances of each observation of the two sets.
- complete or maximum linkage uses the maximum distances between all observations of the two sets.
- single uses the minimum of the distances between all observations of the two sets.

# Configuration: when to stop?

option: `n_clusters` or `distance_threshold`

# Configuration: when to stop?

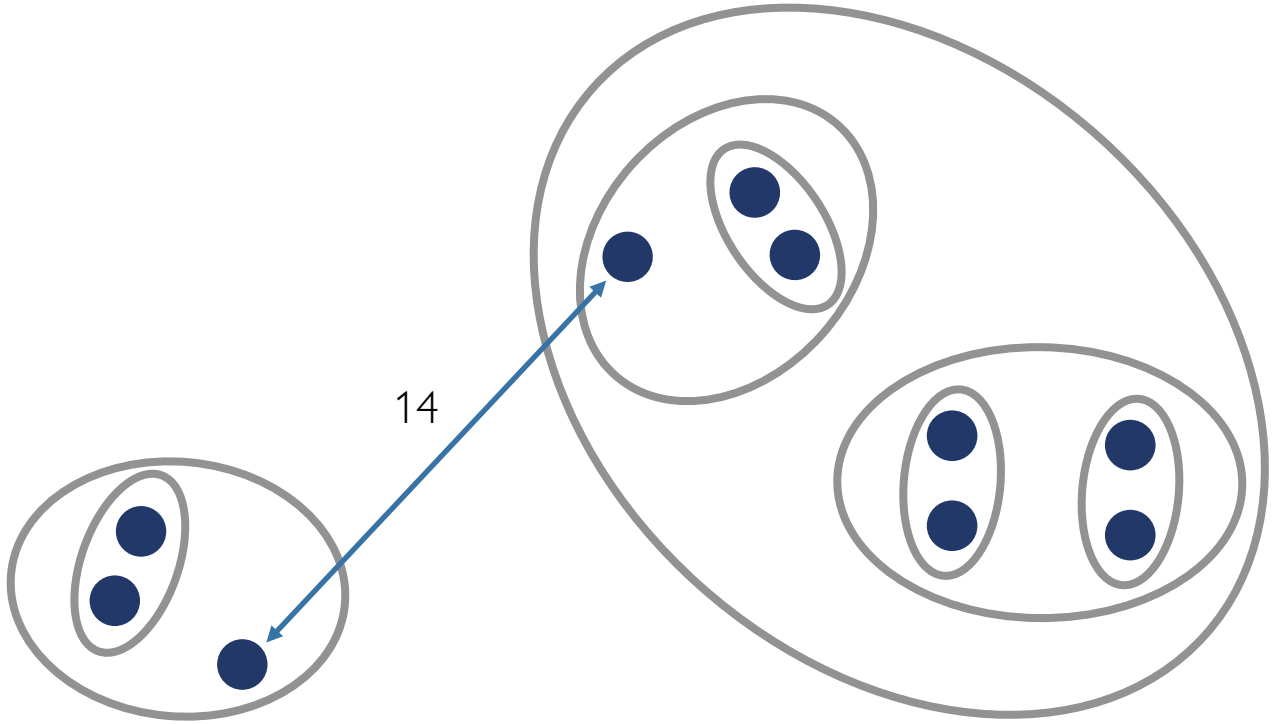
n_clusters=3



each cluster is it's own tree!

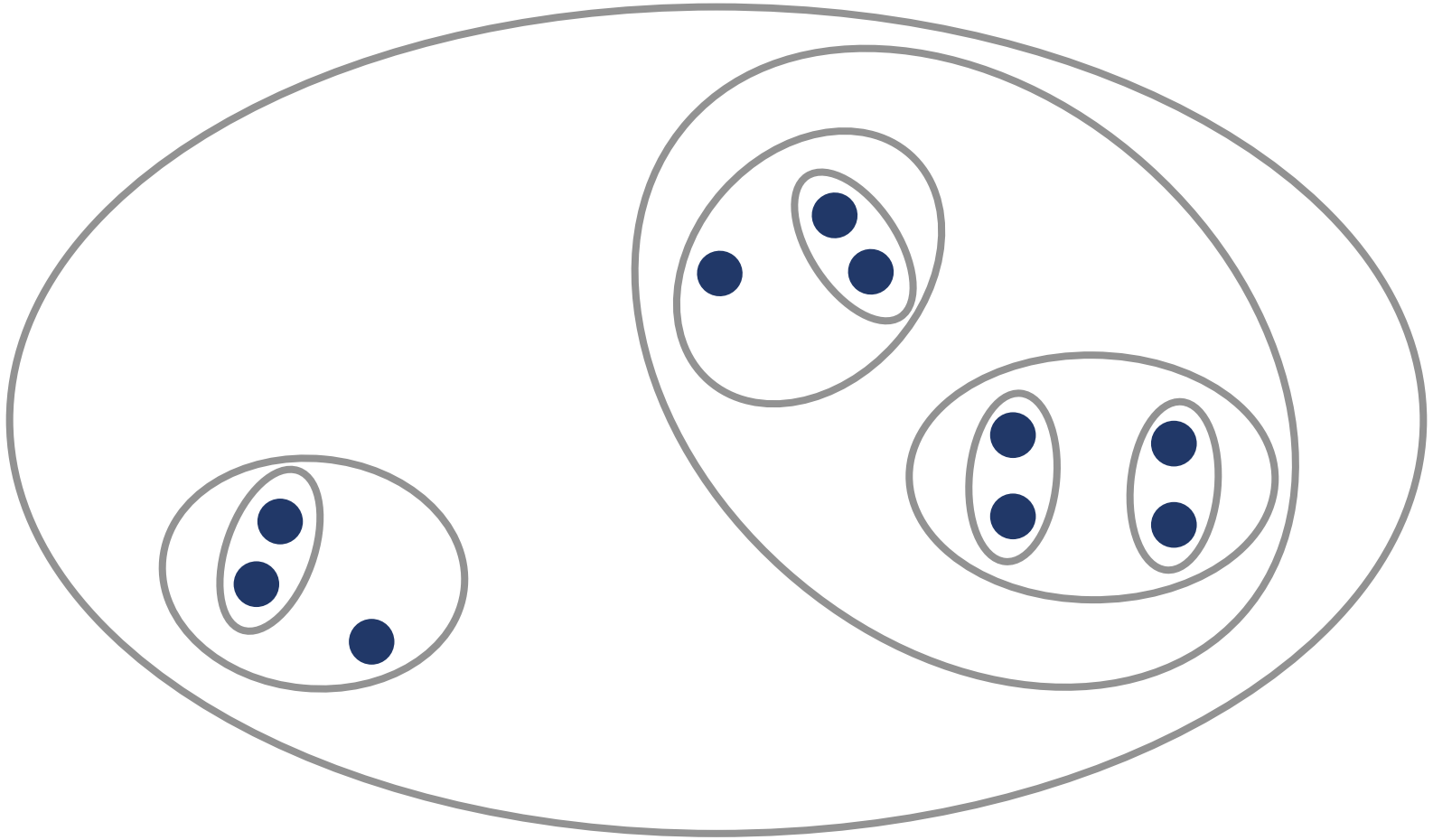
# Configuration: when to stop?

distance_threshold=10

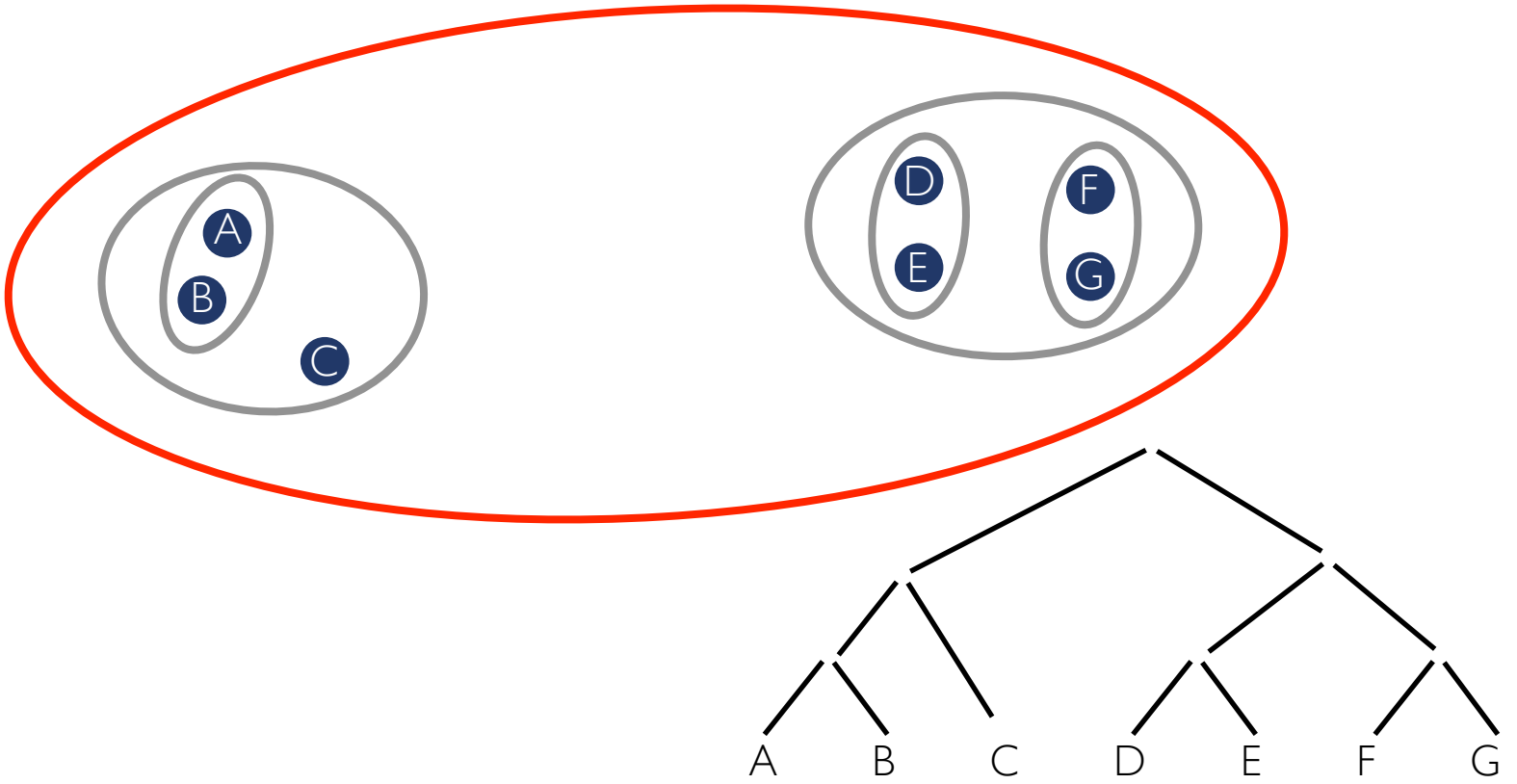


# Configuration: when to stop?

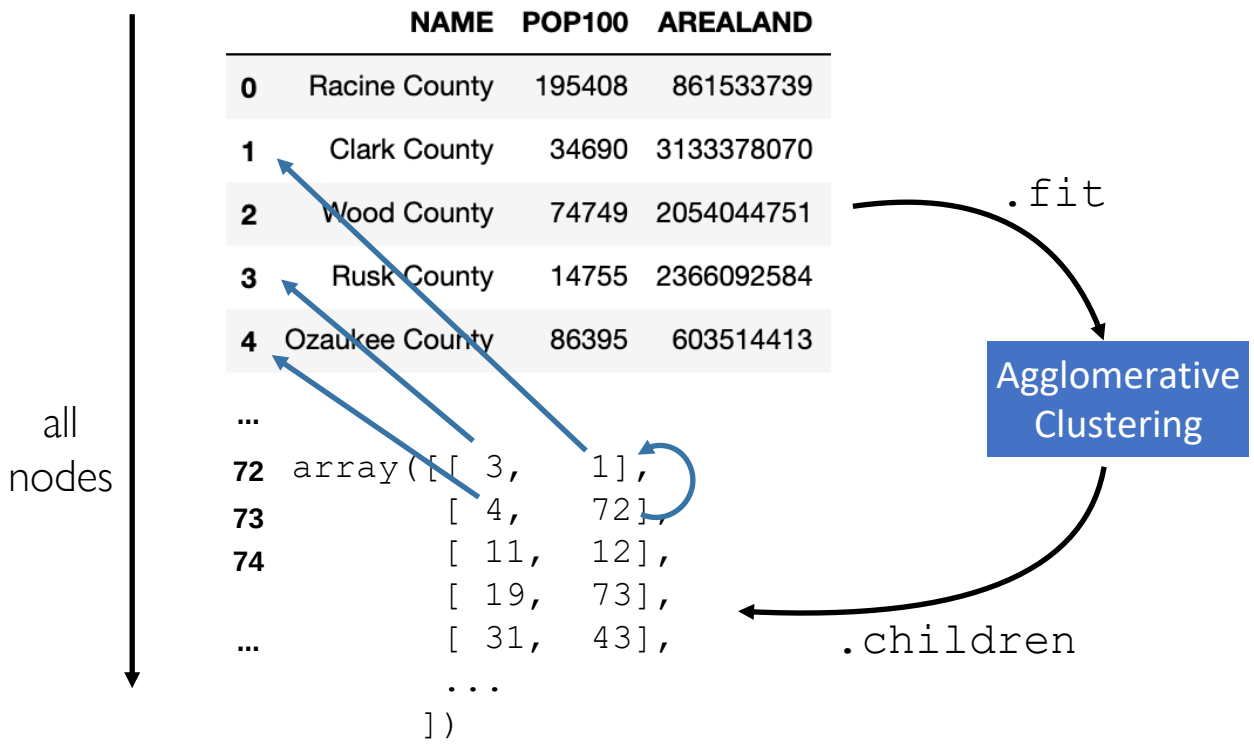
distance_threshold=0



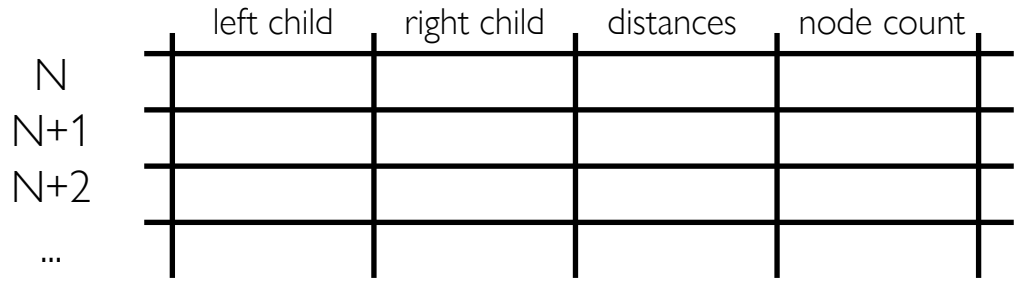
# Strategy: Combine Nearby Points/Groups (and repeat!)



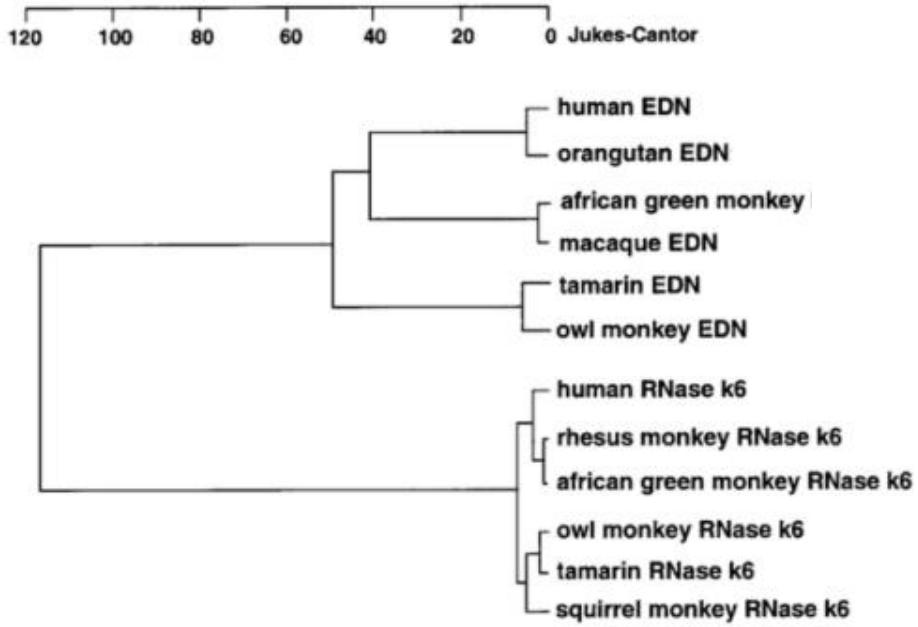
# Node Representation



# Linkage Matrix

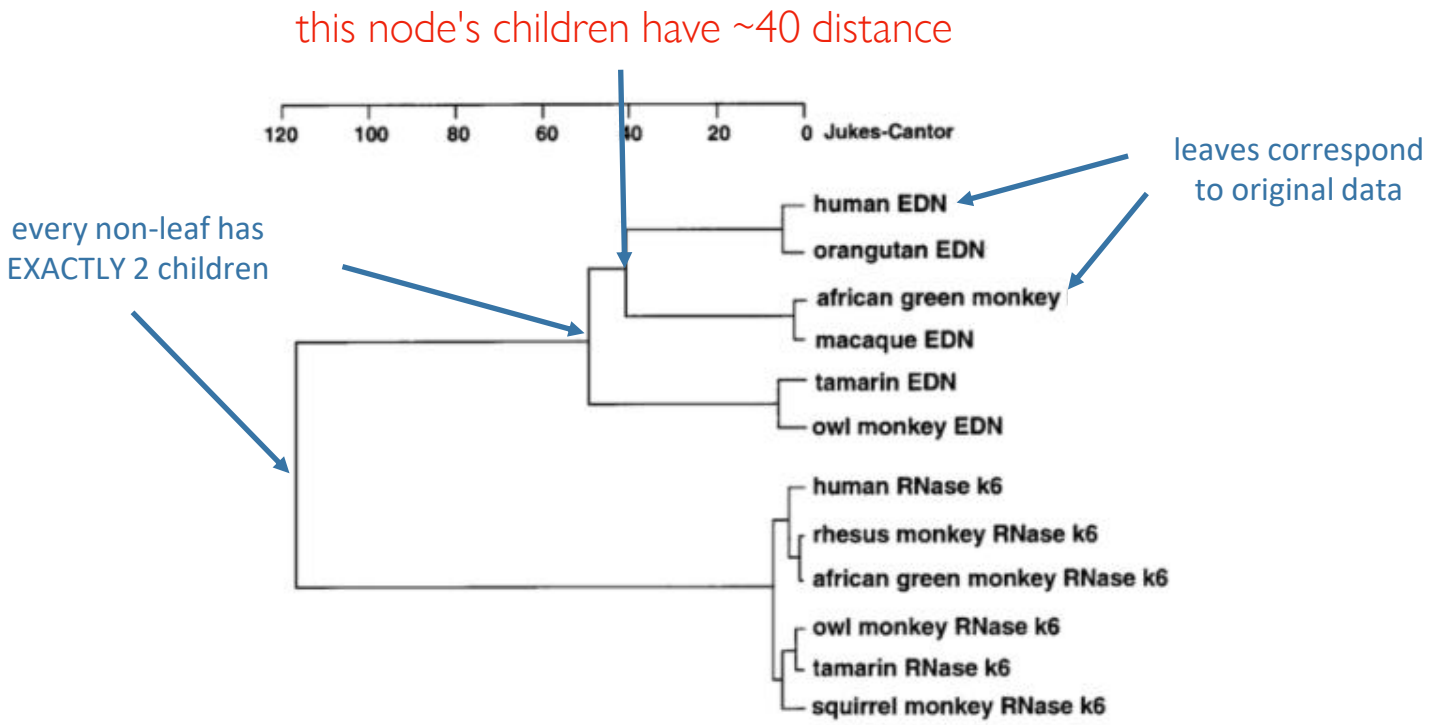


# Hierarchical Clusters with Dendrograms




https://www.researchgate.net/figure/A-Dendrogram-depicting-the-relationships-among-human-and-non-human-primate-EDNs-and_fig1_13459488

# Hierarchical Clusters with Dendrograms




https://www.researchgate.net/figure/A-Dendrogram-depicting-the-relationships-among-human-and-non-human-primate-EDNs-and_fig1_13459488

We'll represent hierarchies as special binary trees.

Demos...