Cluster Analysis I

MENGQIAN LU

Clustering is for?

For Understanding

For Utility

The big picture

- Clustering for Understanding
  - Human beings are skilled at dividing objects into groups – clustering and assigning objects to existing groups (classification), way before we have statistics, let alone data science.
  - For data analysis: clusters are potential groups, cluster analysis is the study of techniques for automatically finding groups.
  - Broad applications in various fields as a (exploratory) learning approach:

1. Biology: (1) Taxonomy of all living things: Kingdom – phylum – class – order – family – genus – species; (2) Gene expression;
2. Information retrieval: e.g. group results to your search query.
3. Medicine and Psychology: e.g. identify different types of depression
4. Business: e.g. segment customers for marketing

The big picture

- Clustering for Utility
  - Clustering techniques can characterize each cluster in terms of a cluster prototype – a data object representing of the other objects in the cluster
  1. Summarization – Use cluster prototypes instead entire data set to do analysis, with comparable results
  2. Compression – Tabulate data with "prototype index" – **Vector Quantization** – often applied to image/sound/video data
  3. Finding Nearest Neighbors much more efficiently – far away cluster prototypes tell that their cluster members are less likely (quite impossible) to be nearest neighbors

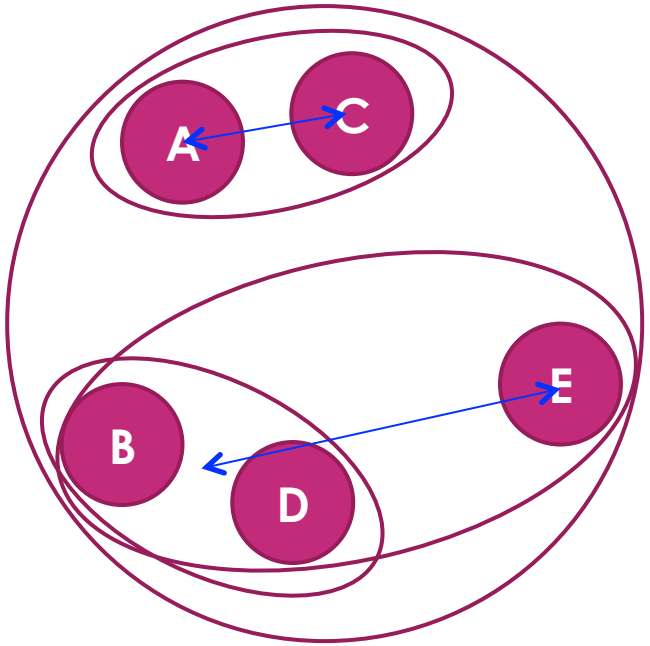
What is Cluster Analysis?

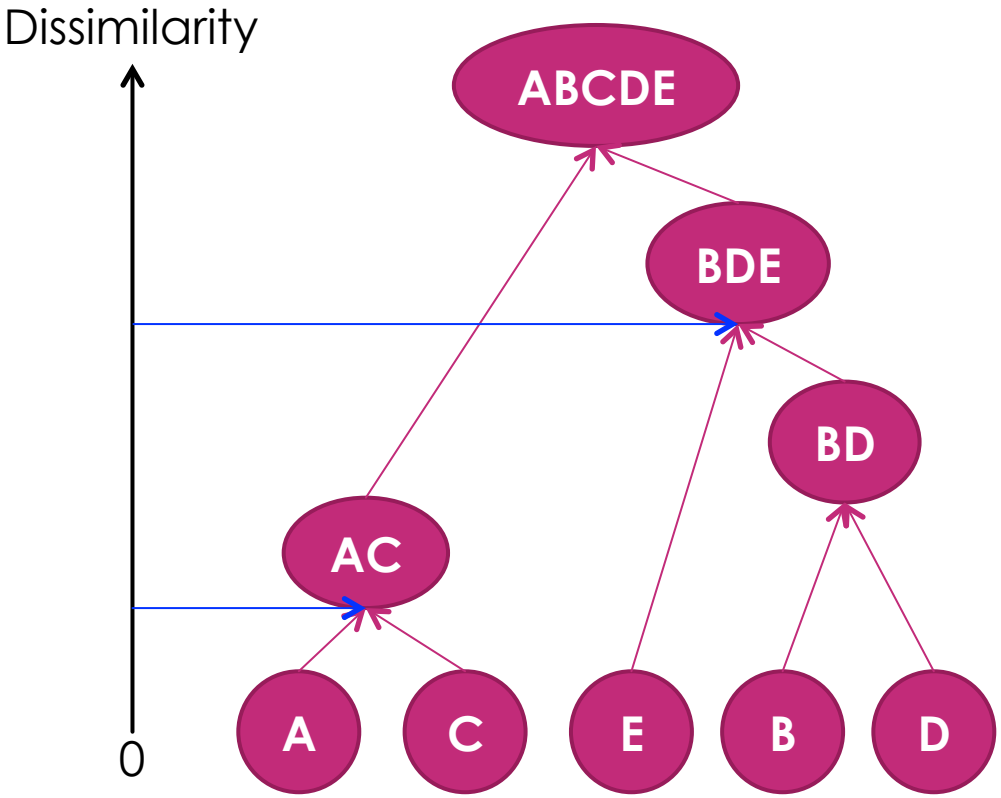
- Definition: To group data objects based on ONLY information found in the data, which tells what are the objects and what are their relationships.
- Goal: Objects in a group are similar to one another, AND different from the objects in other groups [two levels!]
  - The greater the within-group similarity + the greater the between-group difference → the better the clustering
- Cluster analysis vs. Classification – Unsupervised vs. Supervised

Types of Clustering of our focus

1. Hierarchical techniques
   - Agglomerative – build up clusters from individual observations – solve clustering for all possible numbers of cluster at once – choose desired number of cluster afterward
   - Vs. Divisive – start with one group for all and then split off clusters – computational burden
2. K-means clustering
3. Model-based clustering

Agglomerative Hierarchical Clustering





- **Join observations that are closest until only one cluster is left**

Measure dissimilarity between **observations**

- Any dissimilarity we have learnt before can be used
- **Euclidean**
- Manhattan
- Simple matching coefficient

$$SMC = \frac{\text{Number of Matching Attributes}}{\text{Number of Attributes}} = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

- Jaccard dissimilarity: Jaccard similarity coefficient is defined as $J(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$, and Jaccard distance is $d_J(\mathbf{x}, \mathbf{y}) = 1 - J(\mathbf{x}, \mathbf{y})$.
- Gower's dissimilarity
- More…

Measure dissimilarity between **groups**

- Inter-individual distance: $d_{ij} = \sqrt{\sum_{k=1}^{q}(x_{ik} - x_{jk})^2}$

- Two simple inter-group measures by the IAMA author:

1 $d_{AB} = \min_{\substack{i \in A \\ i \in B}}(d_{ij})$ *Single Linkage Clustering*

2 $d_{AB} = \max_{\substack{i \in A \\ i \in B}}(d_{ij})$ *Complete Linkage Clustering*

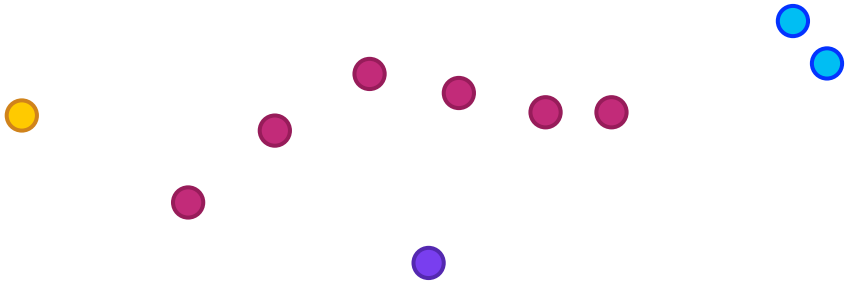
Invariant under monotone transformation of the original inter-individual distance

3 $d_{AB} = \frac{1}{n_A n_B}\sum\sum_{i \in A i \in B} d_{ij}$ *Average Linkage Clustering*

Measure dissimilarity between **groups** (cont'd)

- None is perfect, normally choose complete linkage to get started

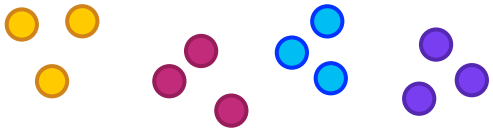
1. *Single Linkage:* suitable for finding elongated cluster



Measure dissimilarity between **groups** (cont'd)

- None is perfect, normally choose complete linkage to get started

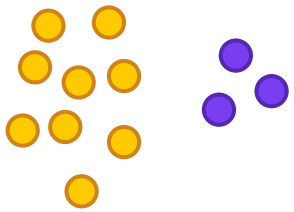
2. *Complete Linkage:* suitable for finding compact but not well separated cluster – "crowd clusters"



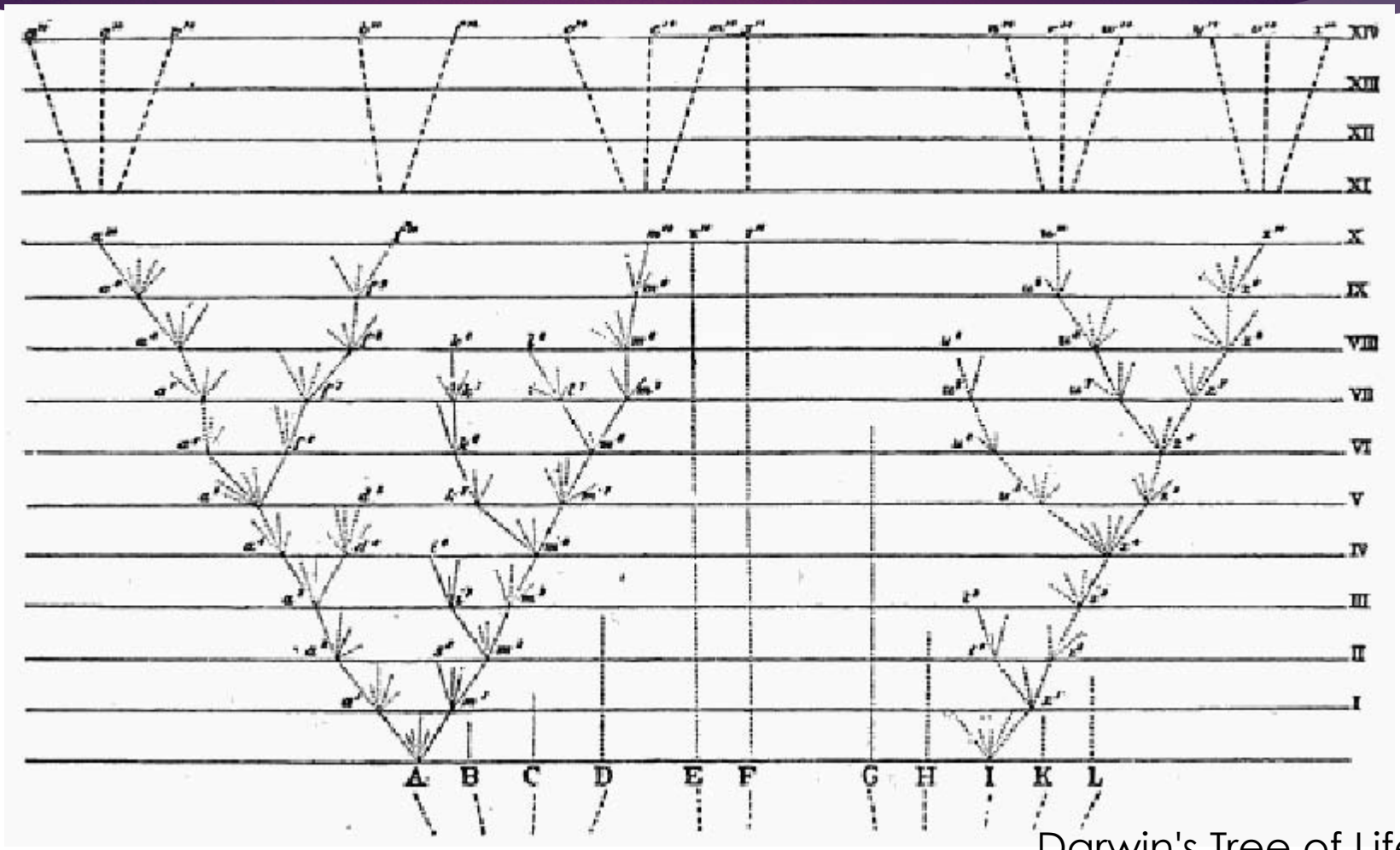
Measure dissimilarity between **groups** (cont'd)

- None is perfect, normally choose complete linkage to get started

3. *Average Linkage:* suitable for finding well separated, oval shaped cluster



Hierarchical clustering – Dengrogram



Darwin's Tree of Life.

hclust() in R

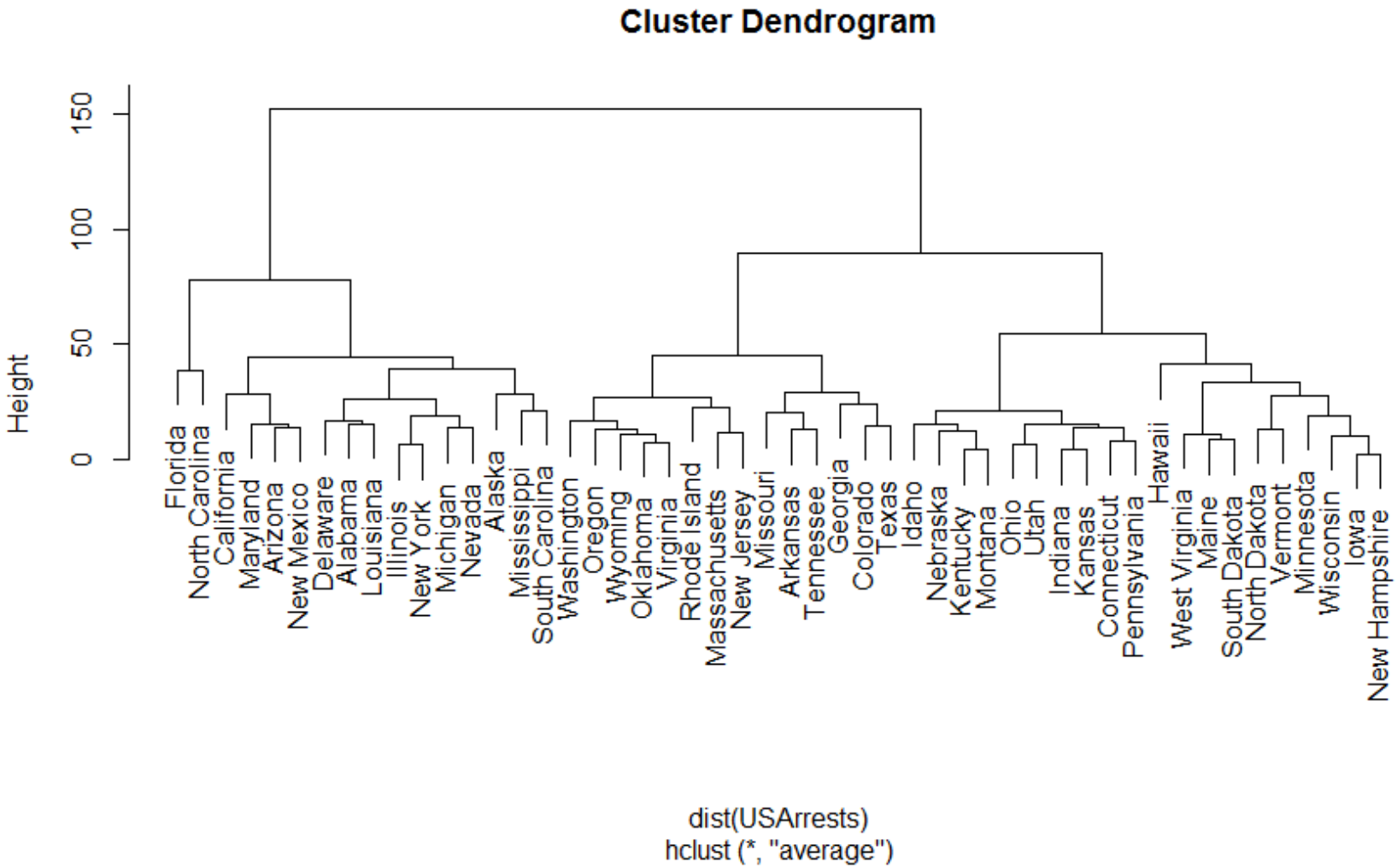
- data(USArrests): numbers are "arrests per 100,000 residents" in the four types of crimes in the 50 US states in 1973.

```
> head(USArrests)
```

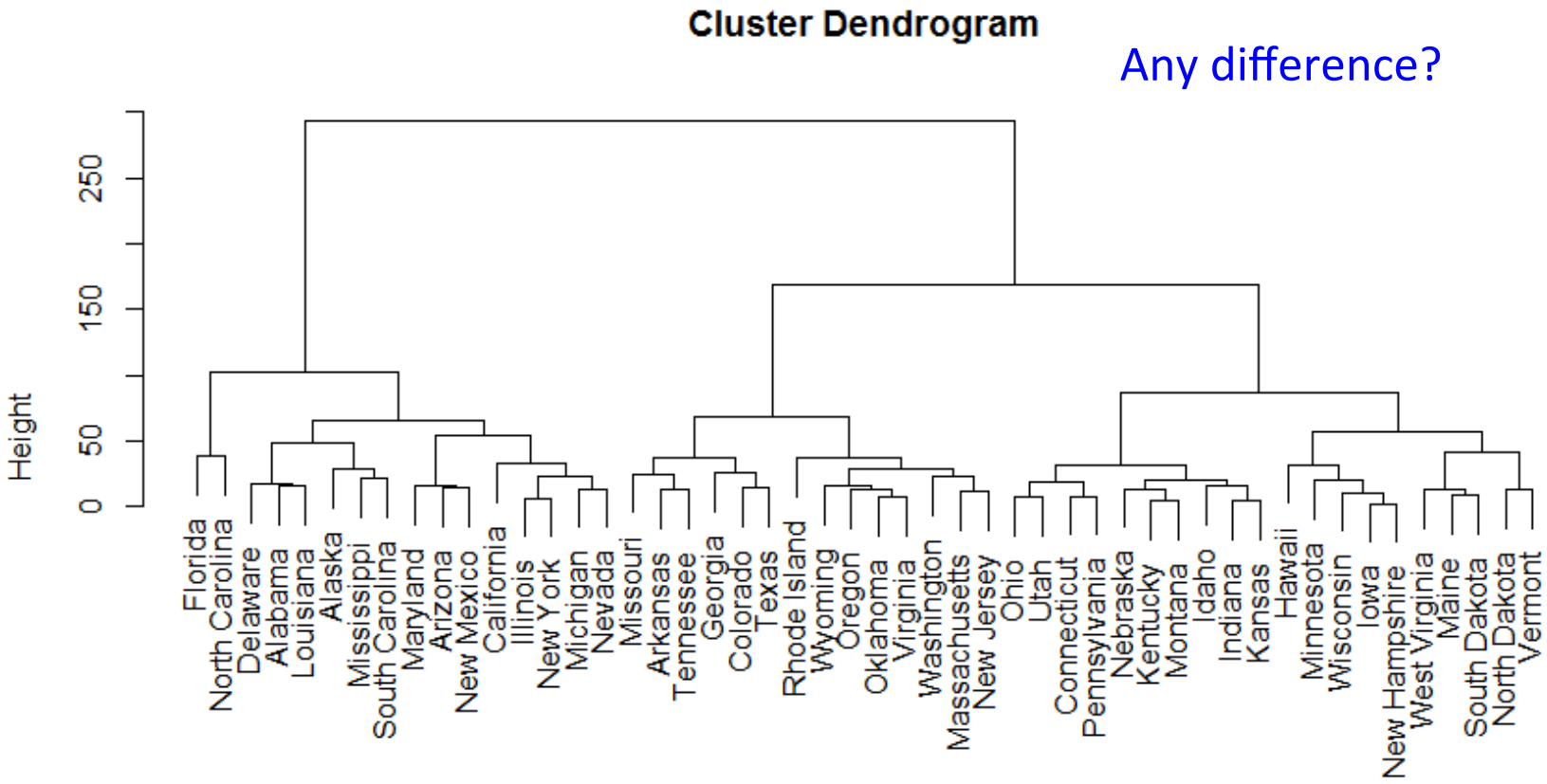
	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

```
hc = hclust(dist(USArrests), "ave")
plot(hc)
```

hclust() in R (cont'd)



hclust() in R (cont'd)



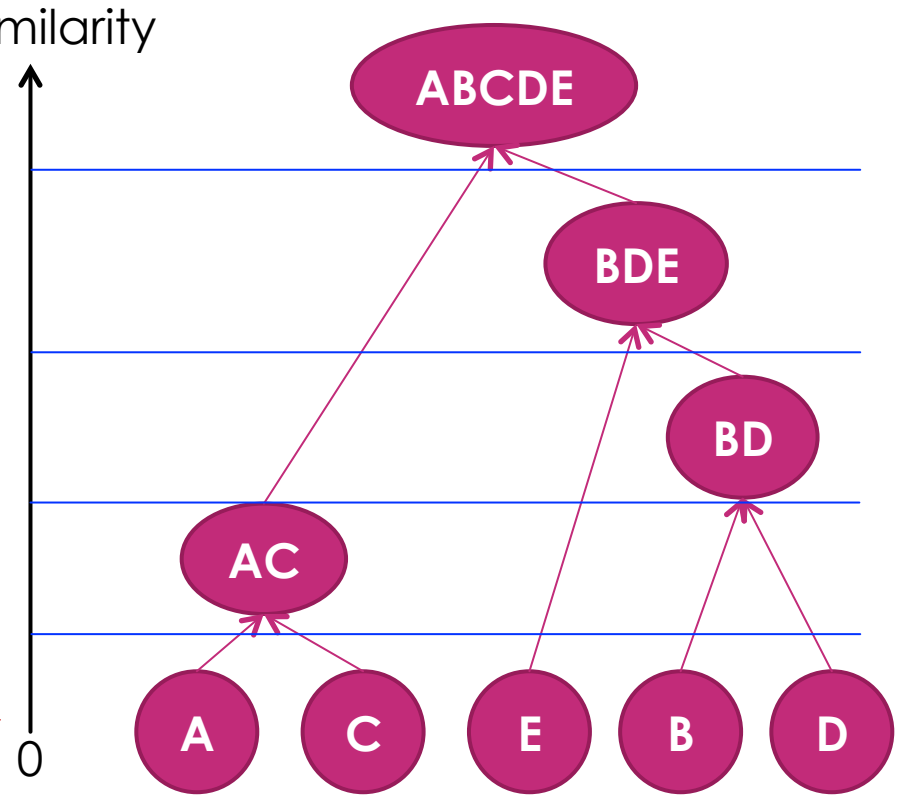
hc = hclust(dist(USArrests), "complete")
plot(hc)

dist(USArrests)
hclust (*, "complete")

Choosing the number of clusters

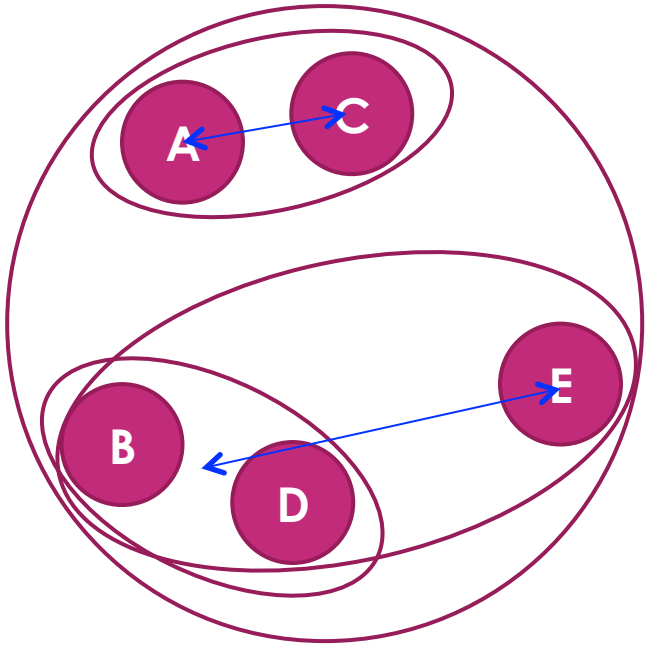
- Get cluster solutions by cutting the tree
- 1 cluster: ABCDE
- 2 clusters: AC – BDE
- 3 clusters: AC – E – BD
- 4 clusters: AC – E – B – D
- 5 clusters: A – C – E – B – D

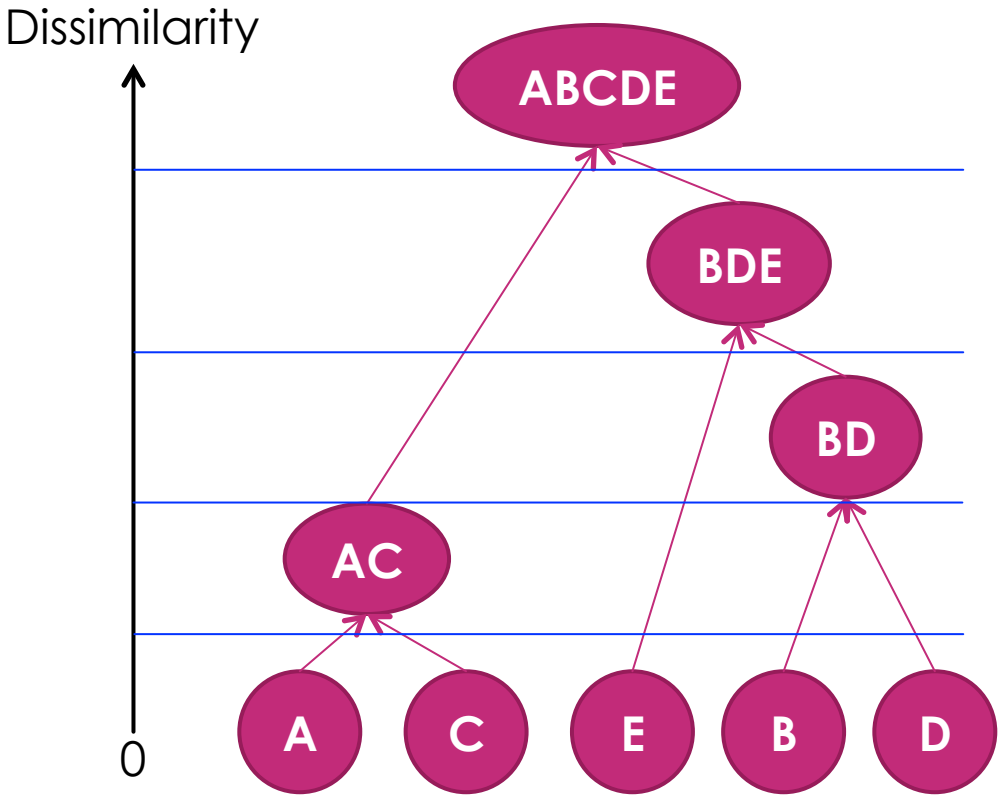
General rule: Find the largest vertical "drop" in the tree



Clustering tree = Dendrogramm

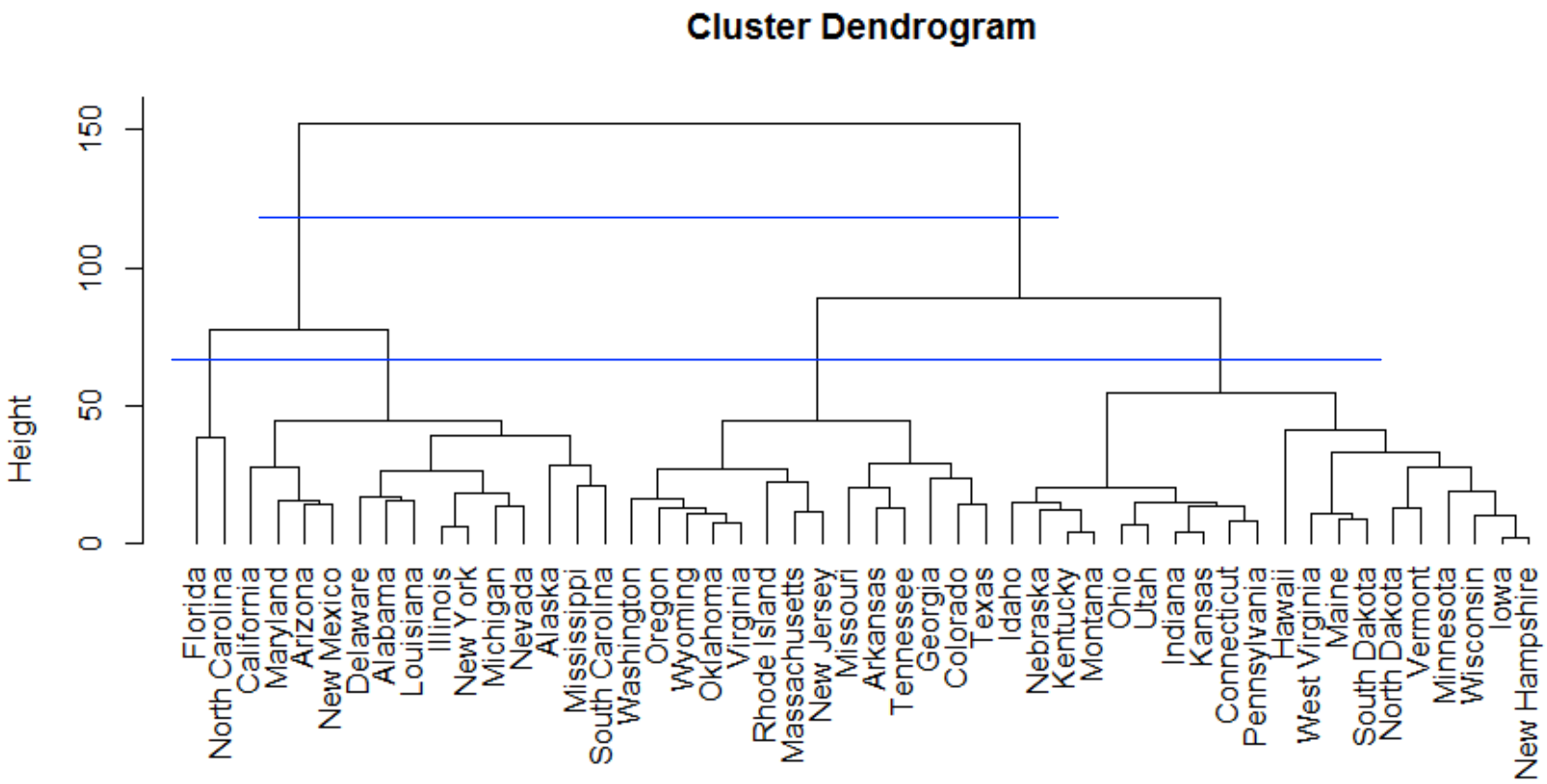
Choosing the number of clusters (cont'd)





Clustering tree = Dendrogramm

Choosing the number of clusters (cont'd)




```
hc = hclust(dist(USArrests), "complete")
plot(hc, hang = -1)
```

Choosing the number of clusters (cont'd)
`cutree()`

```
cutree(hc, k = 1:5)
cutree(hc, h = 250)

## Compare the 2 and 4 grouping:
g24 = cutree(hc, k = c(2,4))
table(grp2 = g24[,"2"], grp4 = g24[,"4"])
```

```
> table(grp2 = g24[,"2"], grp4 = g24[,"4"])
          grp4
grp2  1  2  3  4
1    14  0  0  2
2     0 14 20  0
```

```
> head(cutree(hc, k = 1:5))
           1 2 3 4 5
Alabama    1 1 1 1 1
Alaska     1 1 1 1 1
Arizona    1 1 1 1 1
Arkansas   1 2 2 2 2
California 1 1 1 1 1
Colorado   1 2 2 2 2
```