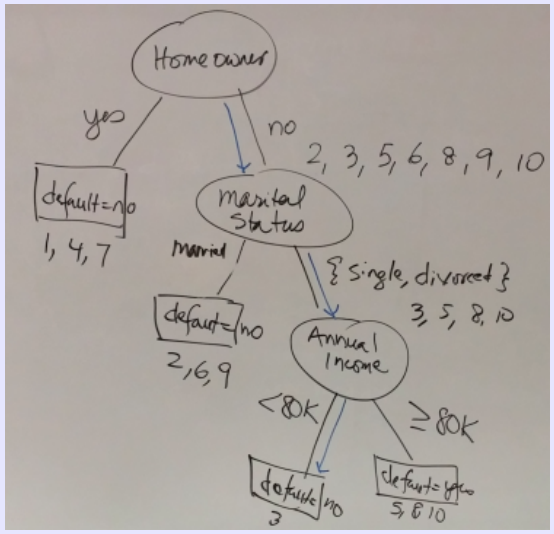
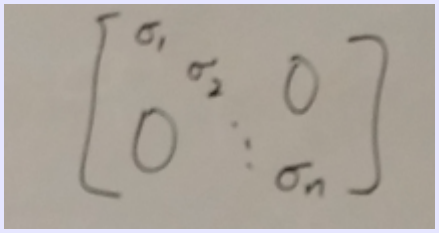
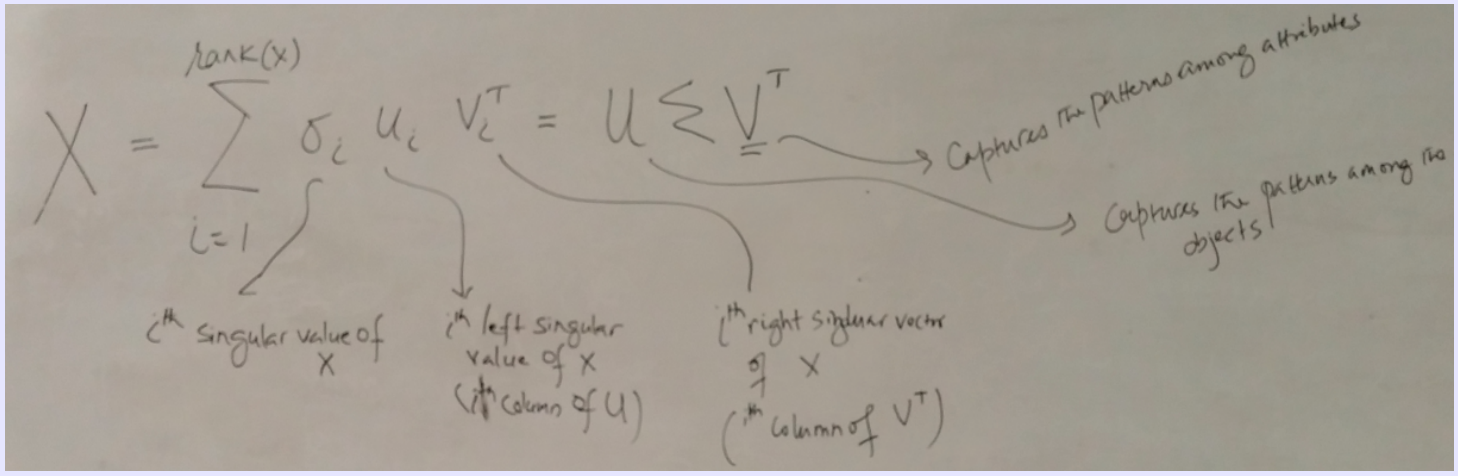
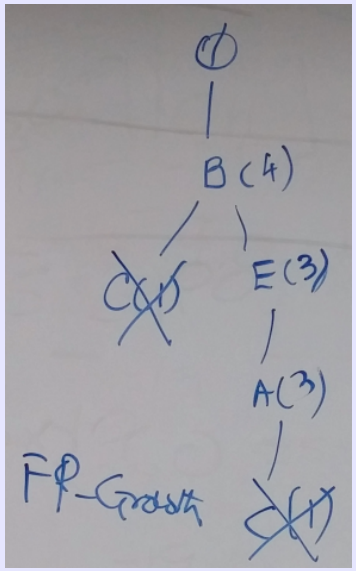





CS 422: Data Mining
Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

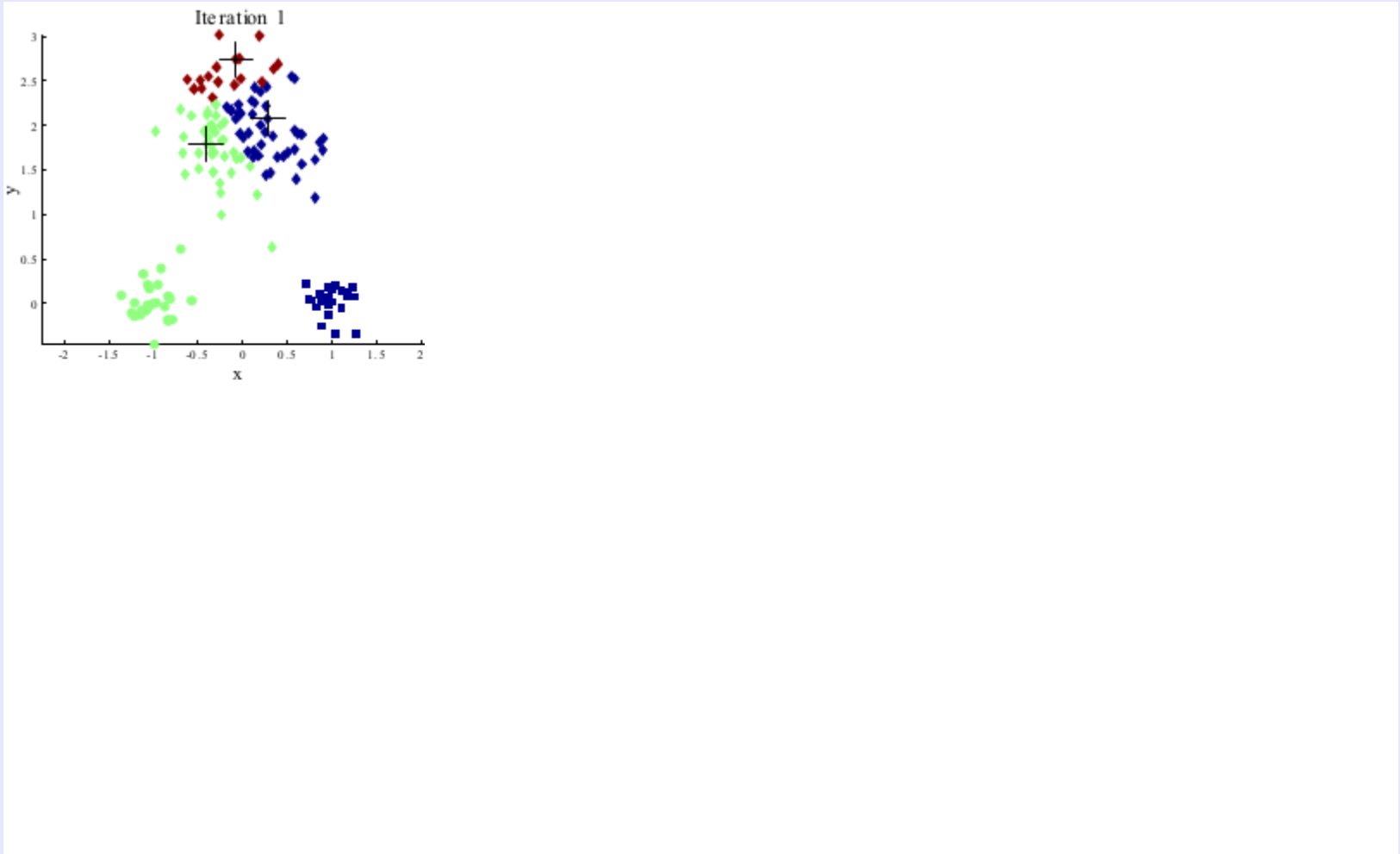
**Clustering I**

# Clustering: K-Means

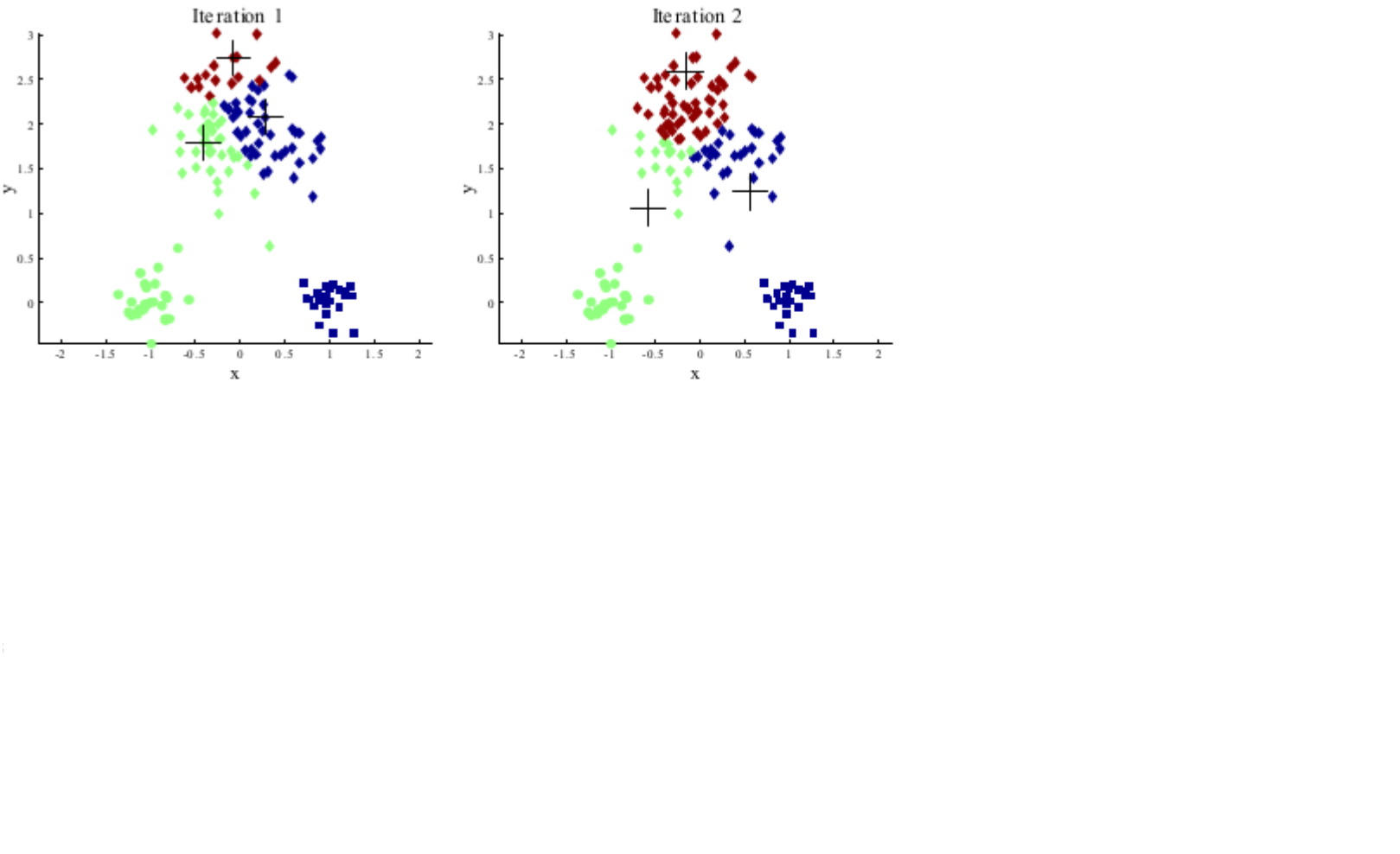
- Computing distances:
  - Euclidean distance (Minkowski, R=2)
  - Manhattan distance (Minkowski, R=1)
  - Jaccard similarity measure
  - Cosine similarity measure
  - …

Used for document data; covered later

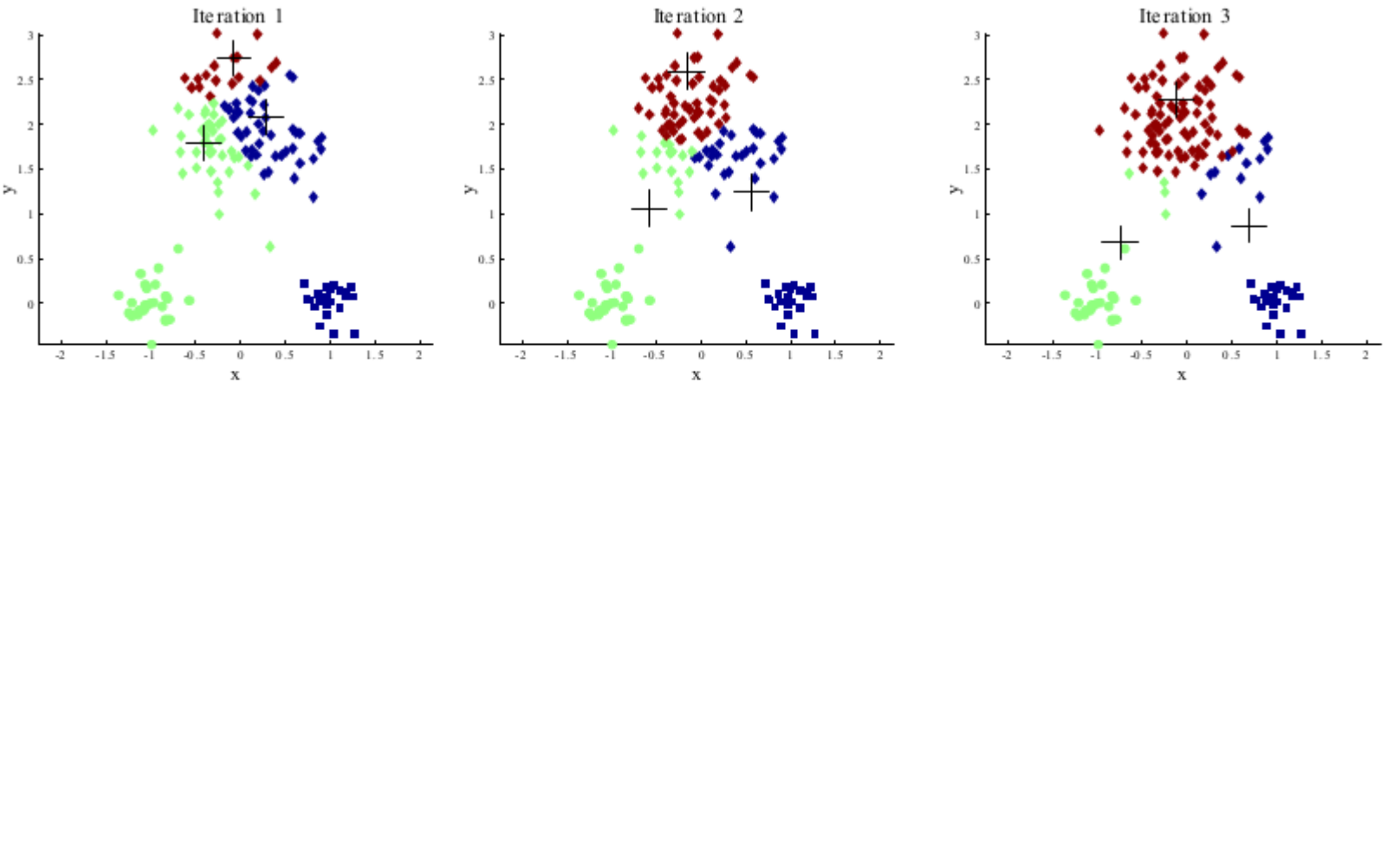
# Clustering: K-Means



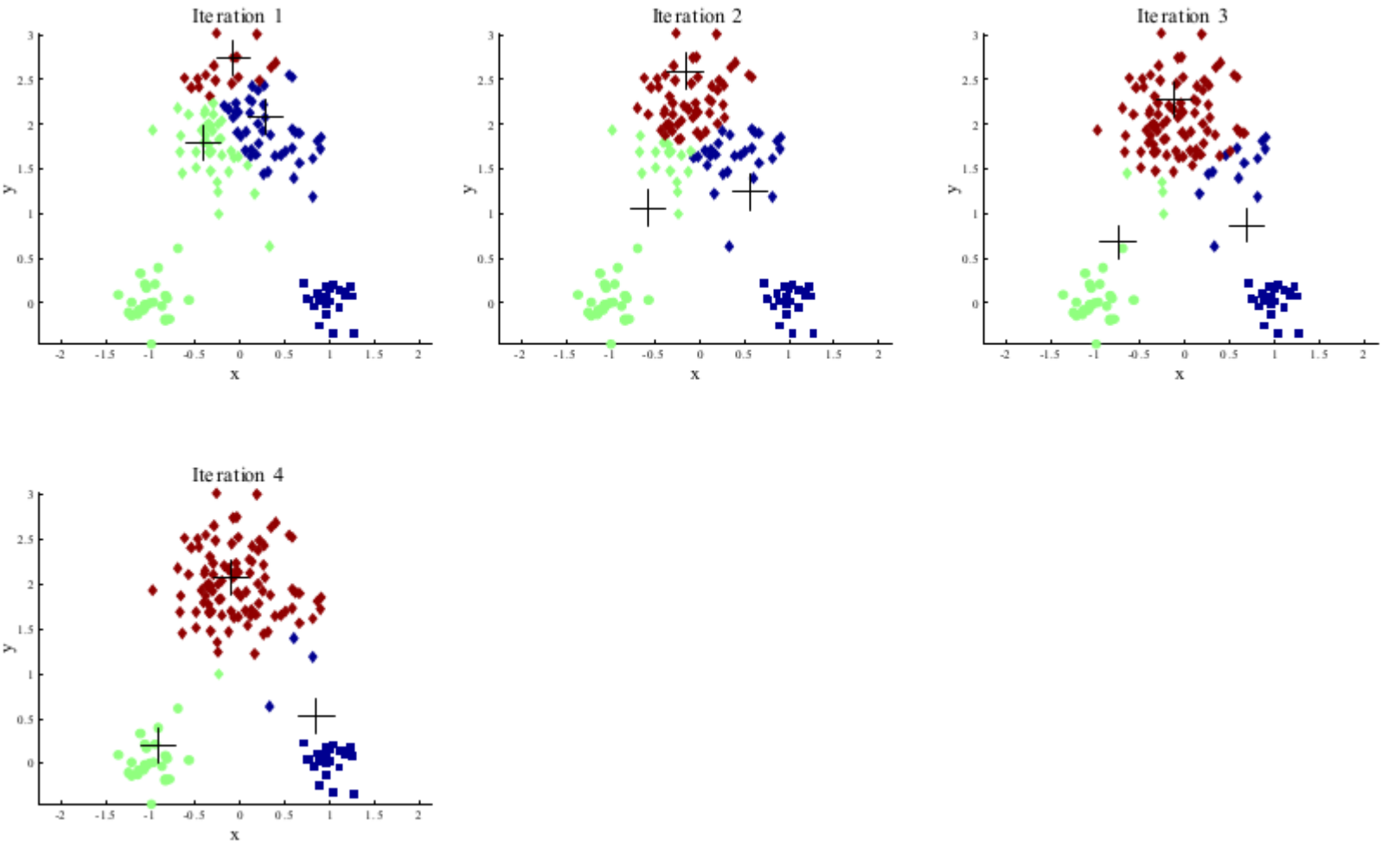
# Clustering: K-Means



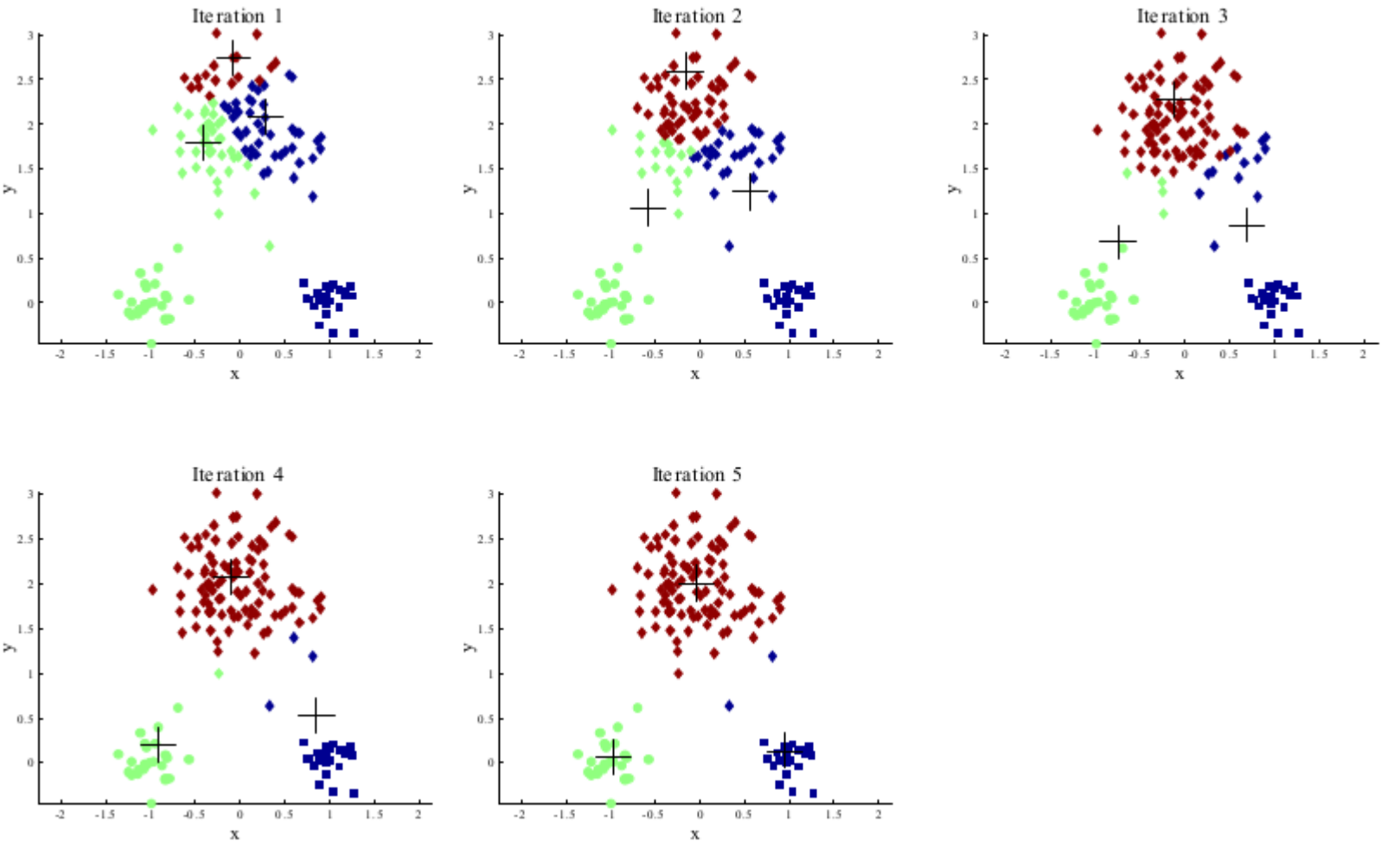
# Clustering: K-Means



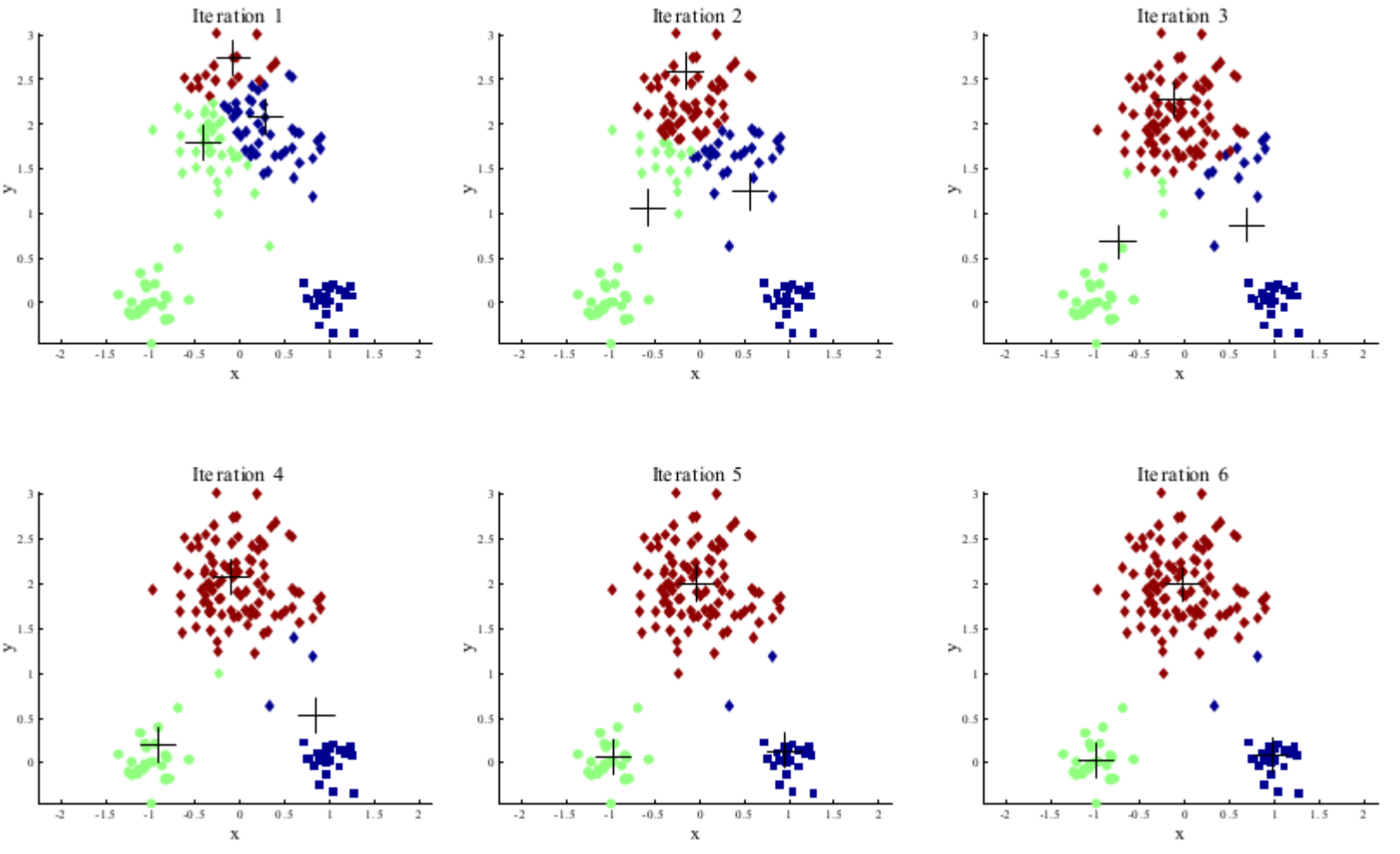
# Clustering: K-Means



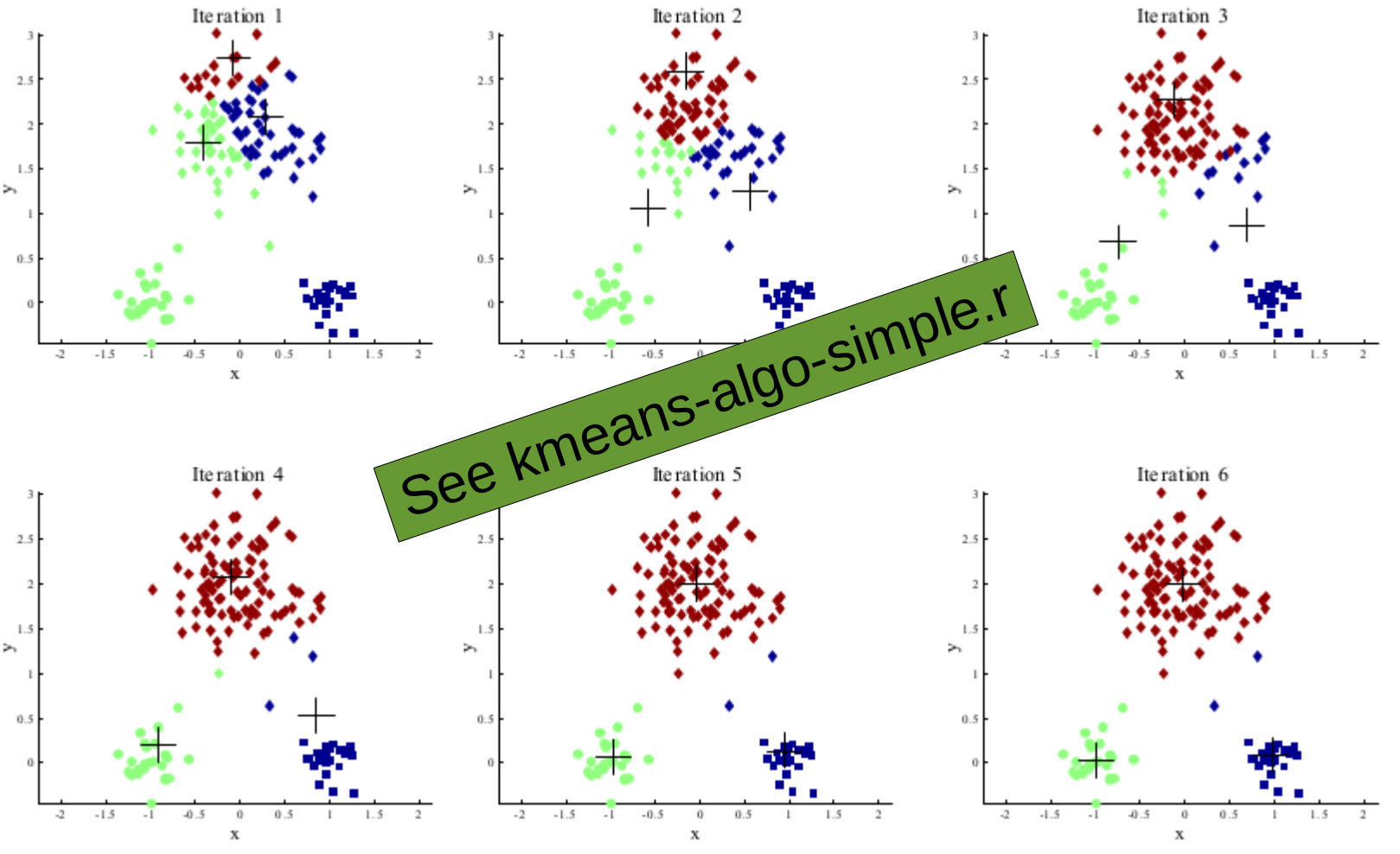
# Clustering: K-Means



# Clustering: K-Means



# Clustering: K-Means



# Clustering: K-Means

- When do we stop?

**Algorithm 8.1** Basic K-means algorithm.

1: Select $K$ points as initial centroids.
2: **repeat**
3: Form $K$ clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

$$\underset{j}{\text{argmin}}\ Dist(x_i, c_j)\ \forall \text{ observations } i \text{ and clusters } j\ \in\ \{1..K\}$$

$$min(SSE\ =\ \sum_{j=1}^{K}\sum_{x \in c_j} Dist(c_j, x)^2)$$

# Clustering: K-Means

- When do we stop?

**Algorithm 8.1** Basic K-means algorithm.

1: Select $K$ points as initial centroids.
2: **repeat**
3: Form $K$ clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

$$\underset{j}{\text{argmin}}\ Dist(x_i, c_j)\ \forall \text{ observations } i \text{ and clusters } j\ \in\ \{1..K\}$$

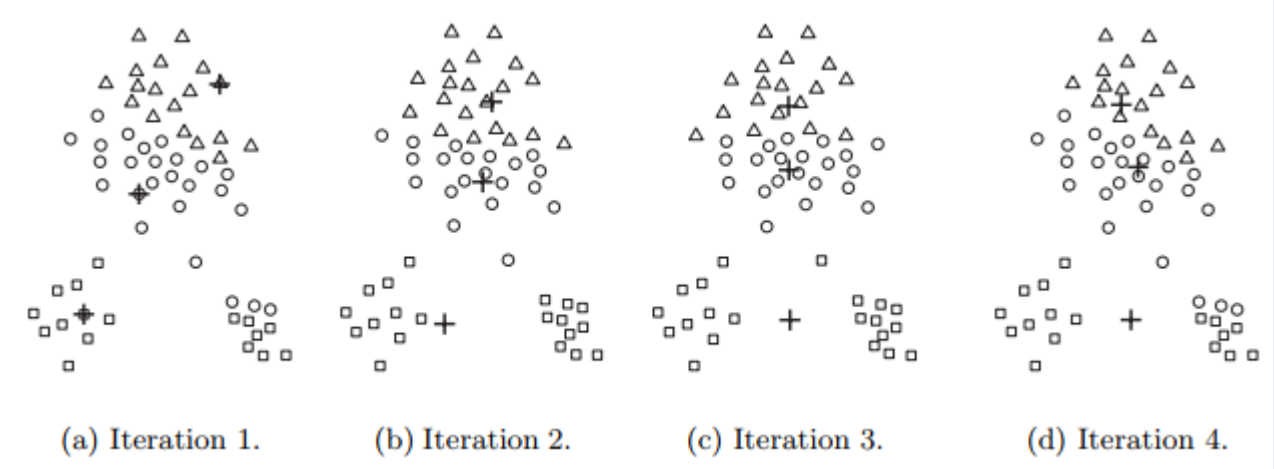
$$min(SSE\ =\ \sum_{j=1}^{K}\sum_{x \in c_j} Dist(c_j, x)^2)$$

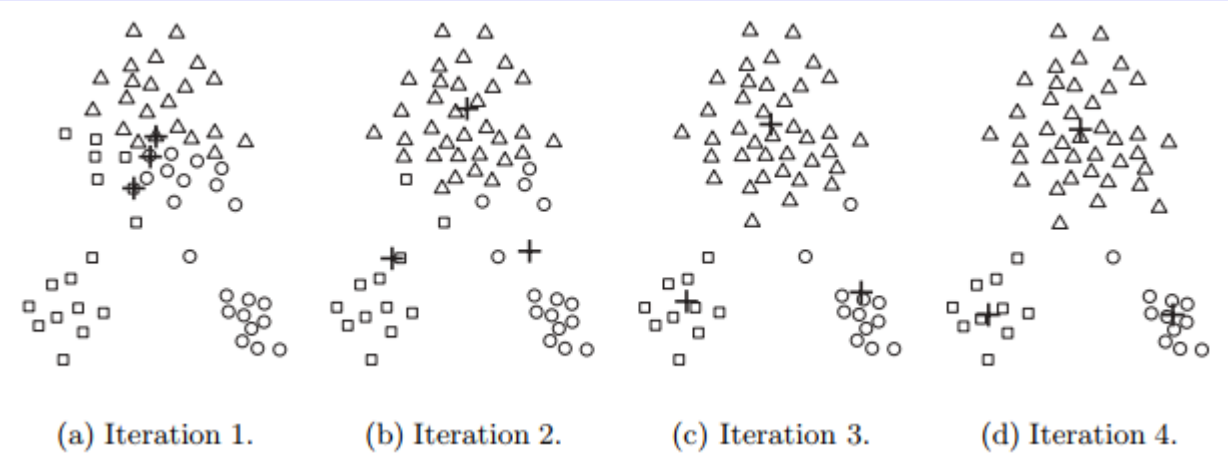
- Does K-Means always converge?

# Clustering: K-Means

- How to choose the initial centroids?
  - Random selection of initial centroids may lead to sub-optimal clustering.

All three initial centroids randomly distributed. Leads to suboptimal solution.




All three initial random centroids in one natural cluster. Leads to good solution.

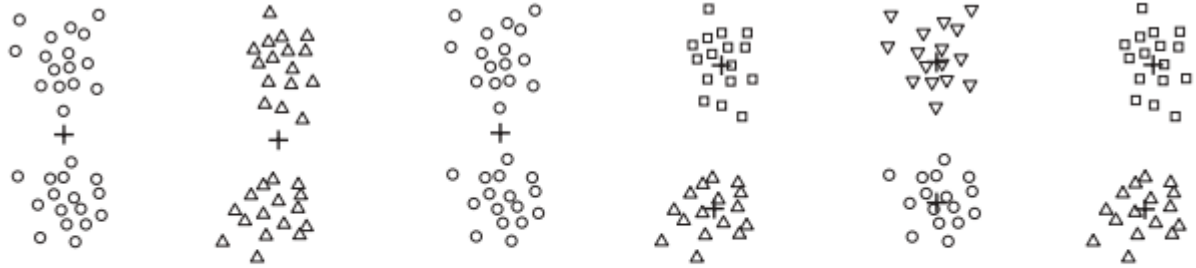
# Clustering: Bisecting K-means

- Goal: Form best (optimal) clusters.
- Simple idea: to obtain *K* clusters, split the set of all points into two clusters, select one of the clusters to split, and so on until you have *K* clusters.
  - Which cluster to split?
    - The largest one, or
    - The one with largest error (SSE), or
    - ...

# Clustering: Bisecting K-means

**Algorithm 8.2** Bisecting K-means algorithm.

1: Initialize the list of clusters to contain the cluster consisting of all points.
2: **repeat**
3: Remove a cluster from the list of clusters.
4: {Perform several "trial" bisections of the chosen cluster.}
5: **for** $i = 1$ to *number of trials* **do**
6: Bisect the selected cluster using basic K-means.
7: **end for**
8: Select the two clusters from the bisection with the lowest total SSE.
9: Add these two clusters to the list of clusters.
10: **until** Until the list of clusters contains $K$ clusters.

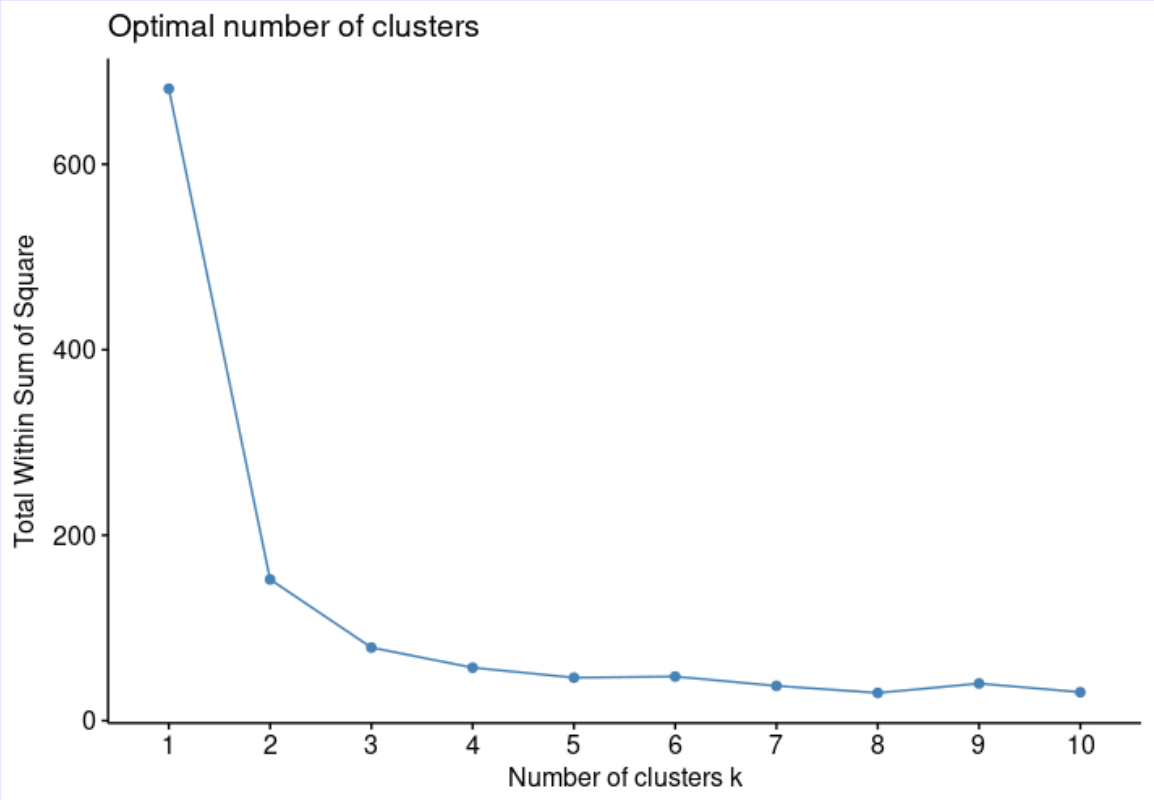


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

**Figure 8.8.** Bisecting K-means on the four clusters example.

# Clustering: K-Means

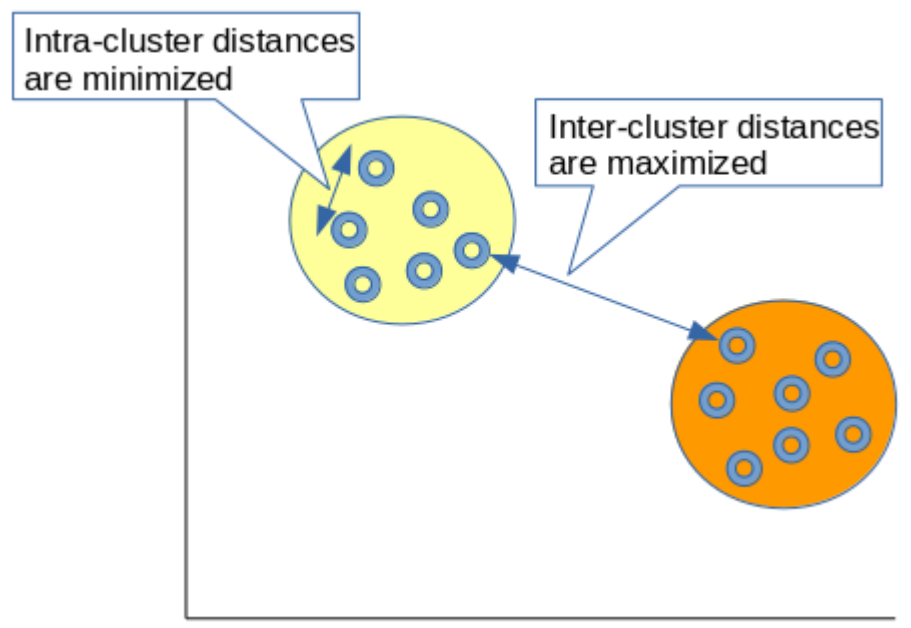
- How to choose the *k* in K-Means?
  - See kmeans-how-many-clusters.r




Within cluster sum of squares is the SSE we saw earlier. We want the total within cluster sum of squares to be minimal.

# Cluster analysis

- How do we evaluate the goodness of fit of clustering?
- Clustering should exhibit **low** intra-cluster SS (also called cohesion) and **high** inter-cluster SS (also called separation).
- Metrics of interest:
  - Within cluster SS
  - Total SS
  - Between SS
  - Variance explained



# Cluster analysis

- Within cluster SS
  - Sum of squares of each point in the cluster to the cluster centroid: $\sum_{x \in C_i} Dist(c_i, x)^2$, where $C_i$ is cluster i and $c_i$ is centroid of cluster i
- Total SS
  - Sum of squares of each point in the dataset to the **global cluster** mean:
    $\sum_{x \in D} Dist(C_g, x)^2$, where $C_g$ is global cluster mean, and D is the clustering dataset
- Between SS
  - Total SS - total within cluster SS, where total within cluster SS =
    $\sum_{j=1}^{k} \sum_{x \in C_j} Dist(c_j, x)^2$, where k is number of clusters, $C_j$ is cluster j, and $c_j$ is centroid of cluster j

# Cluster analysis

- Variance
  - Between SS / Total SS (between 0.0 and 1.0)
  - The higher this ratio, the more variance is explained by the clusters.

# Clustering: R implementation

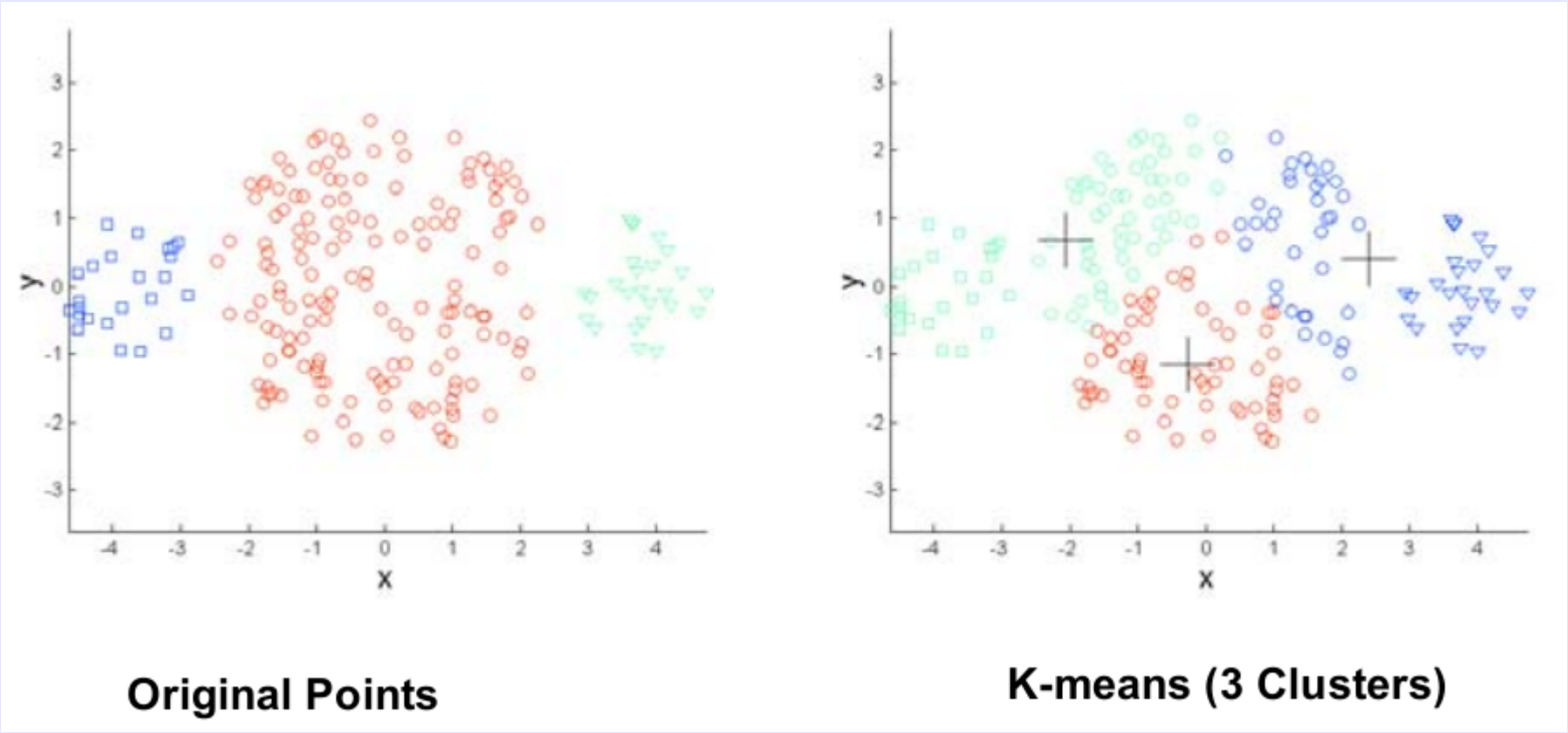
- Clustering in R:
  - Package: cluster, factoextra
  - How to cluster, or using R's clustering libraries.
  - Visualizing the clusters.
  - Why scale (standardize)?
  - See cluster-and-scaling.r

# K-means: Practical issues

- K-means does not handle outliers gracefully; it will try to include these in a cluster.
  - Outlier detection and removal prior to K-means can help.
- What is the right value of *k*?
- K-means has problems when clusters are of differing...

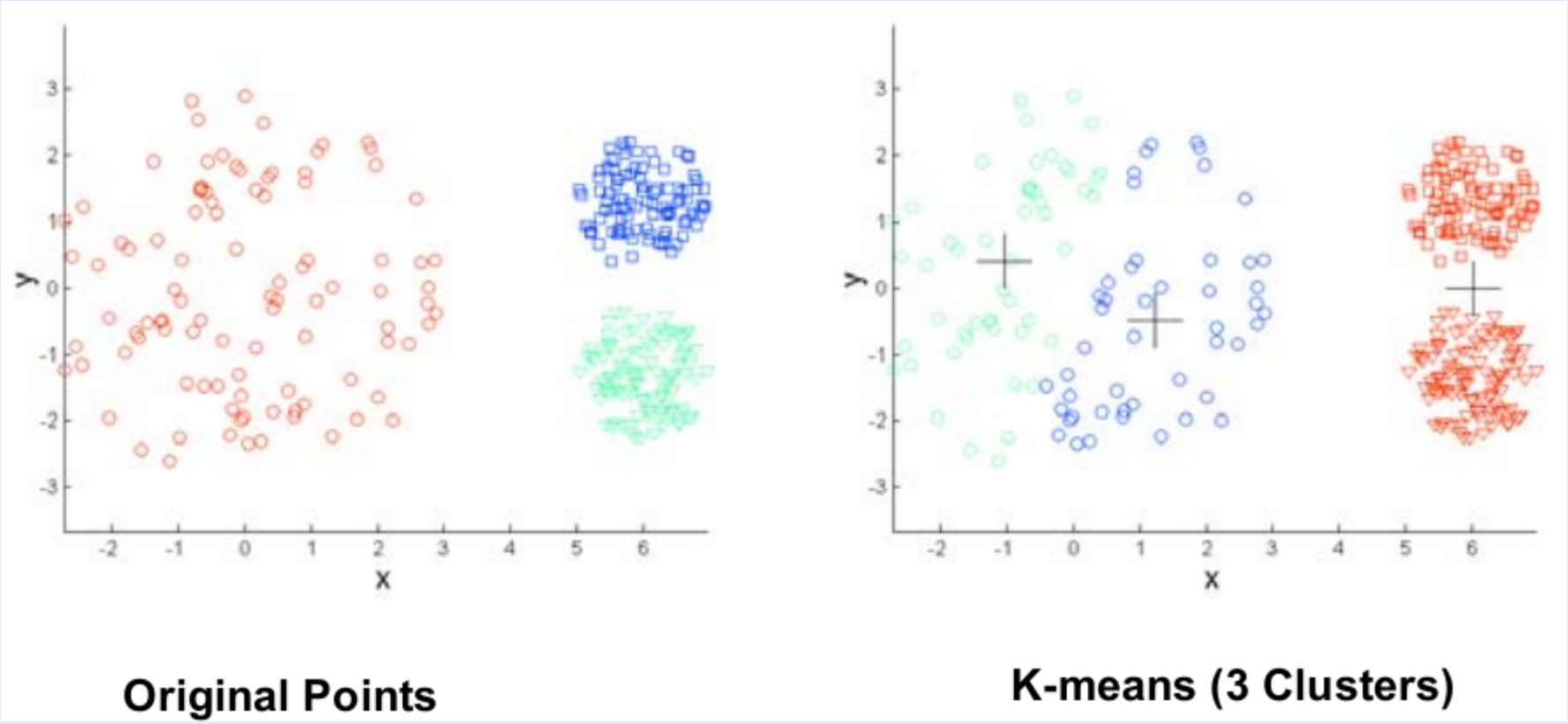
# K-means: Practical issues

- K-means: differing sizes.



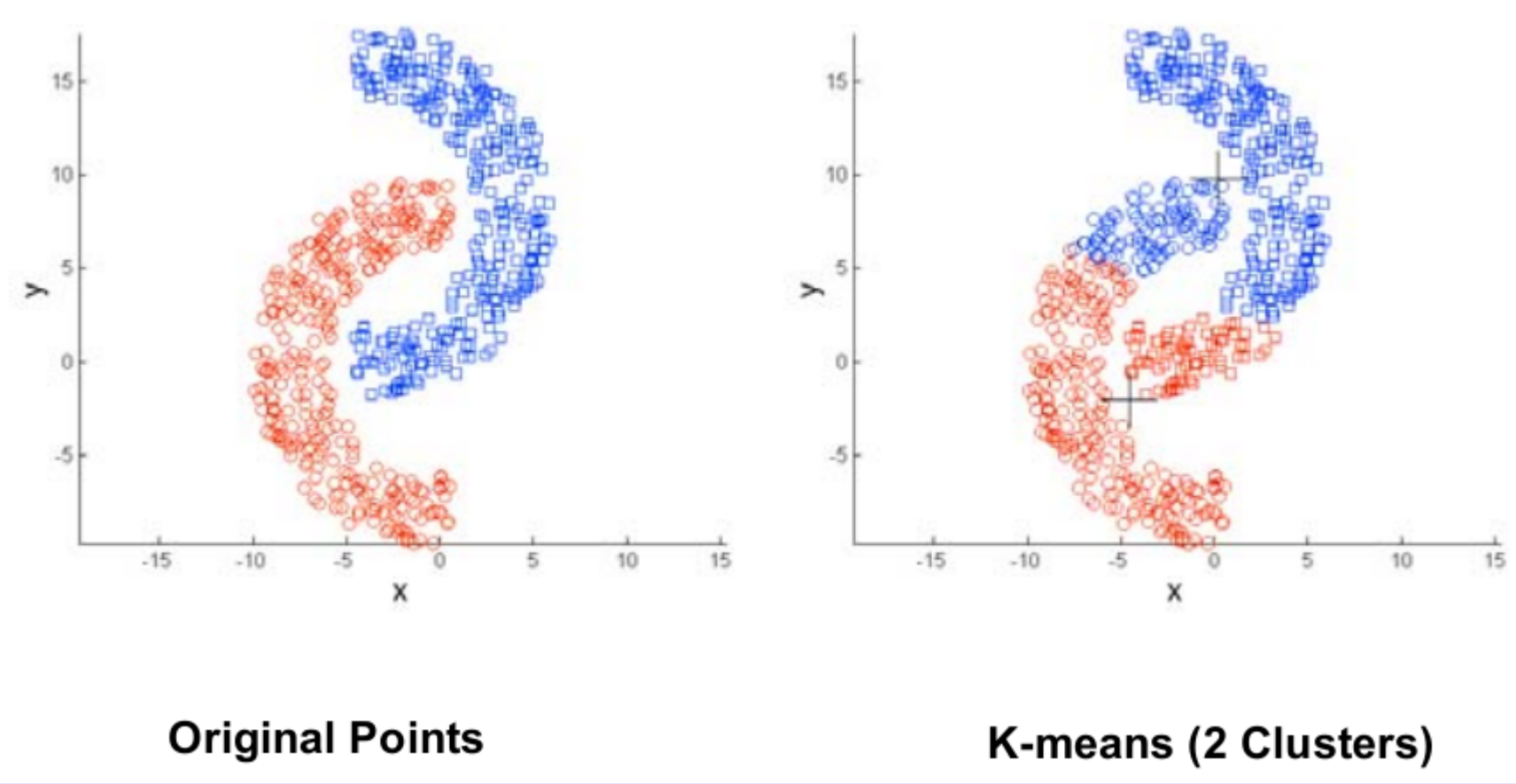
# K-means: Practical issues

- K-means: differing densities.



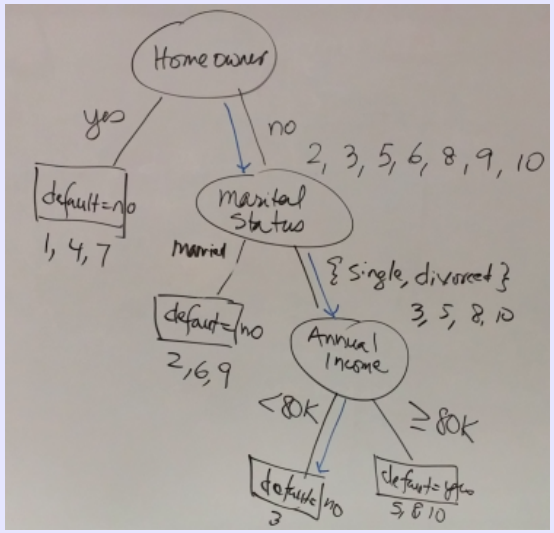
# K-means: Practical issues

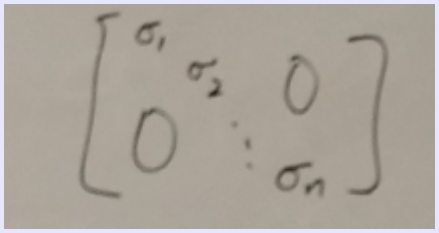
- K-means: problem with non-globular formations.

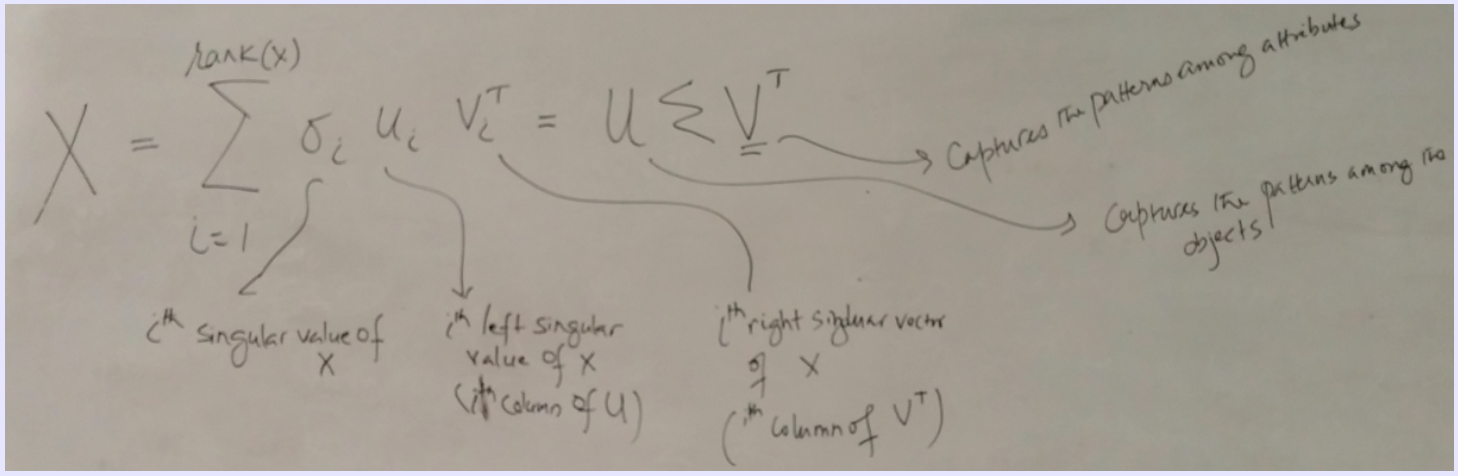


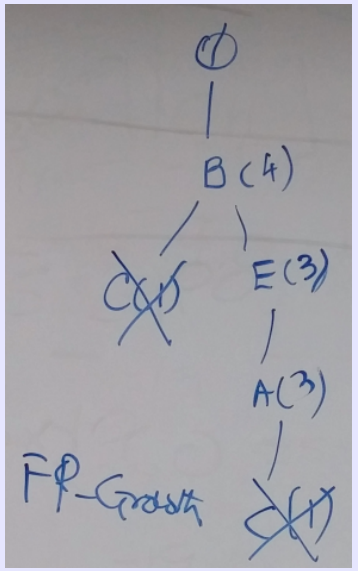
# Clustering: K-Means

- Complexity
  - Space: *O((m+K)n)*, *m* = number of observations, and *n* = number of attributes, *K* = number of clusters.
  - Time: *O(I*K*m*n)*, *I* = number of iterations required to converge.
- Additional issues:
  - Empty clusters.
  - Outliers may lead to higher SSE and non-representative cluster centroids.





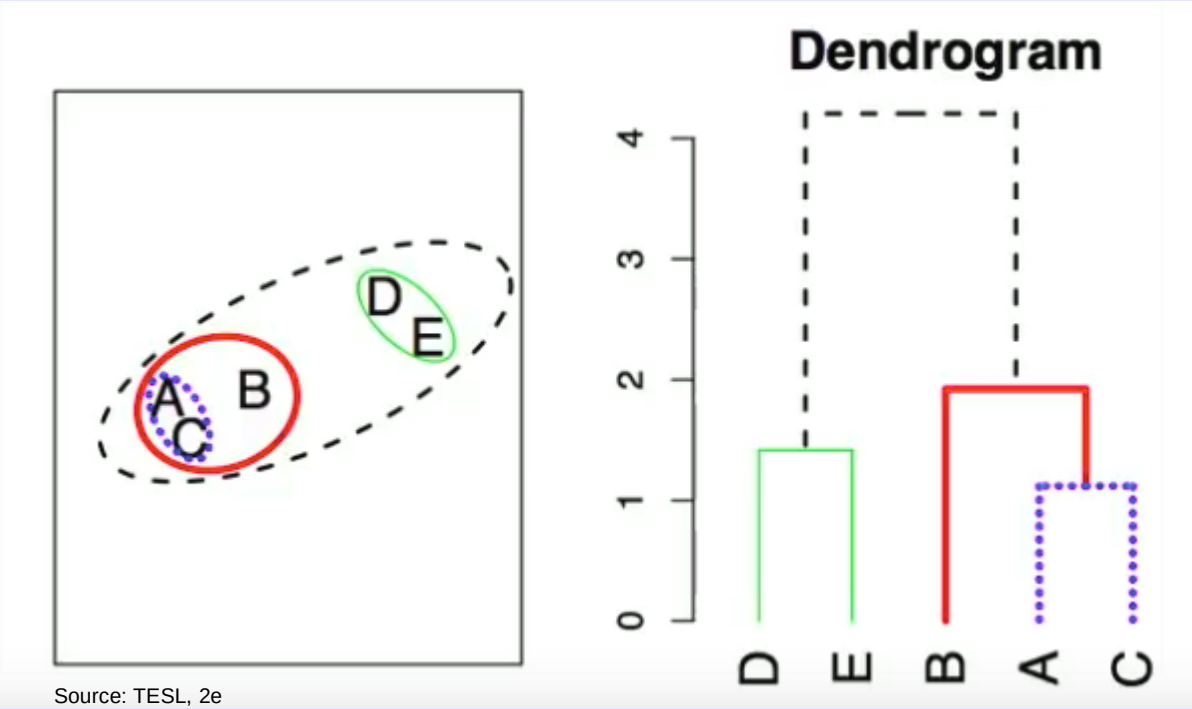

CS 422: Data Mining
Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

**Clustering II**

# Hierarchical clustering

- Inspired by the area of taxonomy, where hierarchical structures are common and elements under the same hierarchy automatically constitute a cluster.
- Unlike K-means, does not require us to choose *k* a-priori. Both an advantage and disadvantage.
- Two approaches:
  - Agglomerative (bottom-up): Each point starts off as an individual cluster, and at each step, merge closest pairs of clusters. (Need cluster proximity metric.)
  - Divisive: All points in one cluster, at each step, split a cluster until singleton clusters of individual points remain. (Which cluster to split and how to do the splitting.)

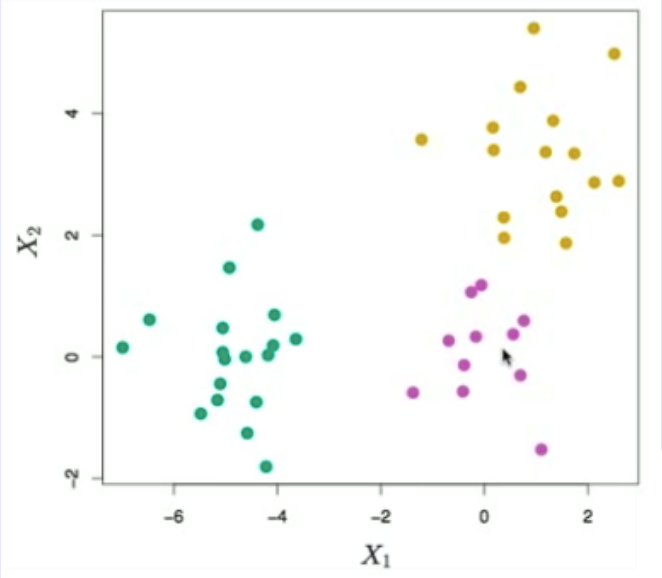
# Hierarchical clustering



Source: TESL, 2e

- Depicted as tree-like structure called Dendrograms.
- Y-axis labeled with the proximity between clusters.
- Look for closest cluster (in terms, say, squared distance), and join them. And continue until one cluster left.

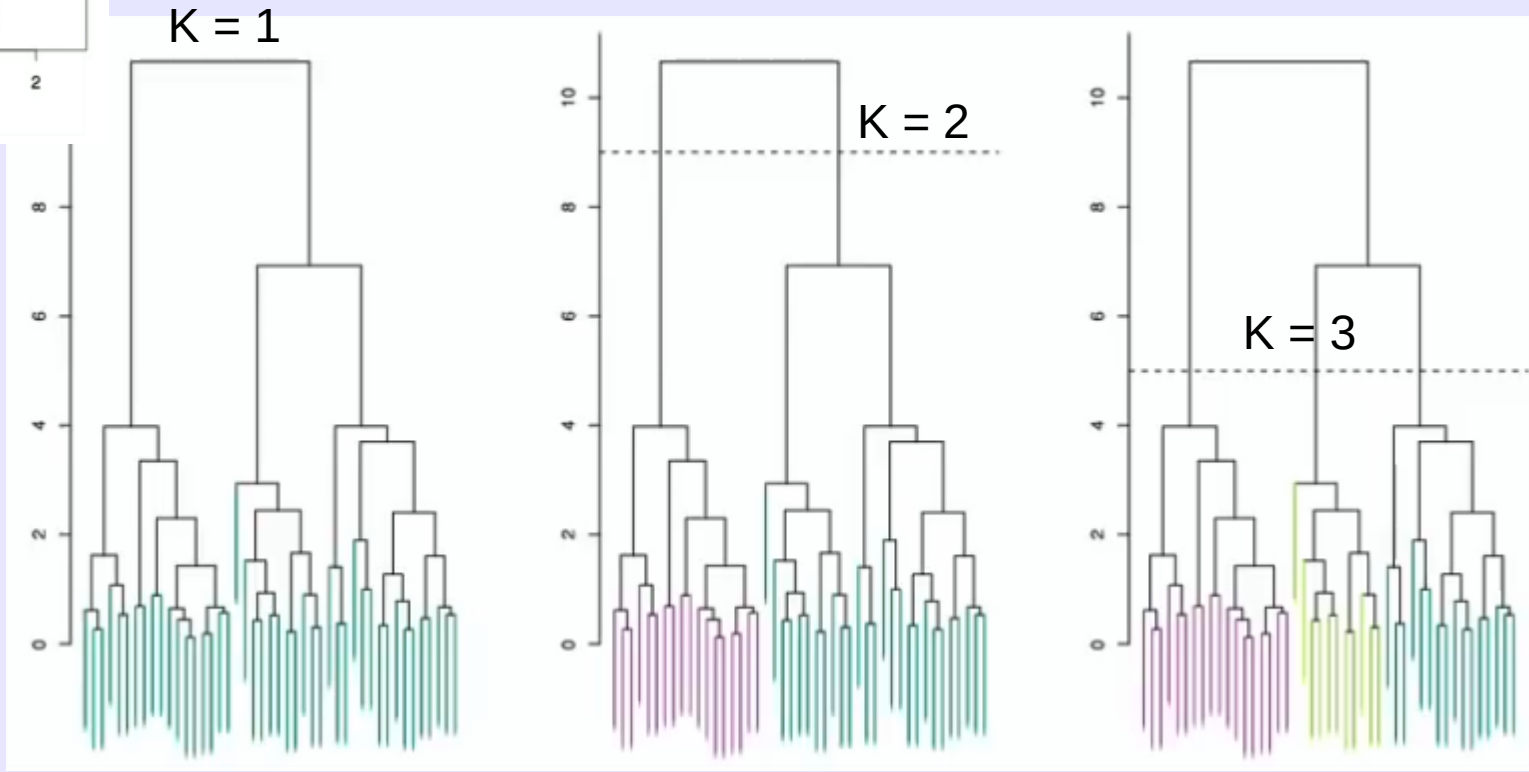
# Hierarchical clustering




Source: TESL, 2e

45 data points in 2-D. 3 distinct classes, shown in separate colors. However, we treat the class labels as unknown and seek to cluster the observations to discover classes from the data.

Note: We can play around with K to get required clustering. K can vary from 1 to 45, in which case each point is a cluster.



# Hierarchical clustering

**Algorithm 8.3** Basic agglomerative hierarchical clustering algorithm.

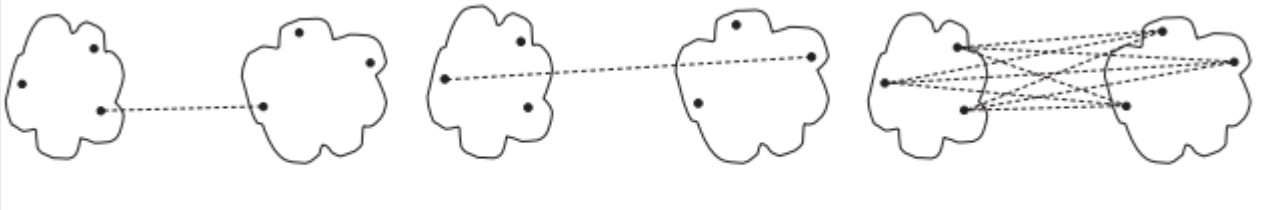
1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

# Hierarchical clustering

**Algorithm 8.3** Basic agglomerative hierarchical clustering algorithm.

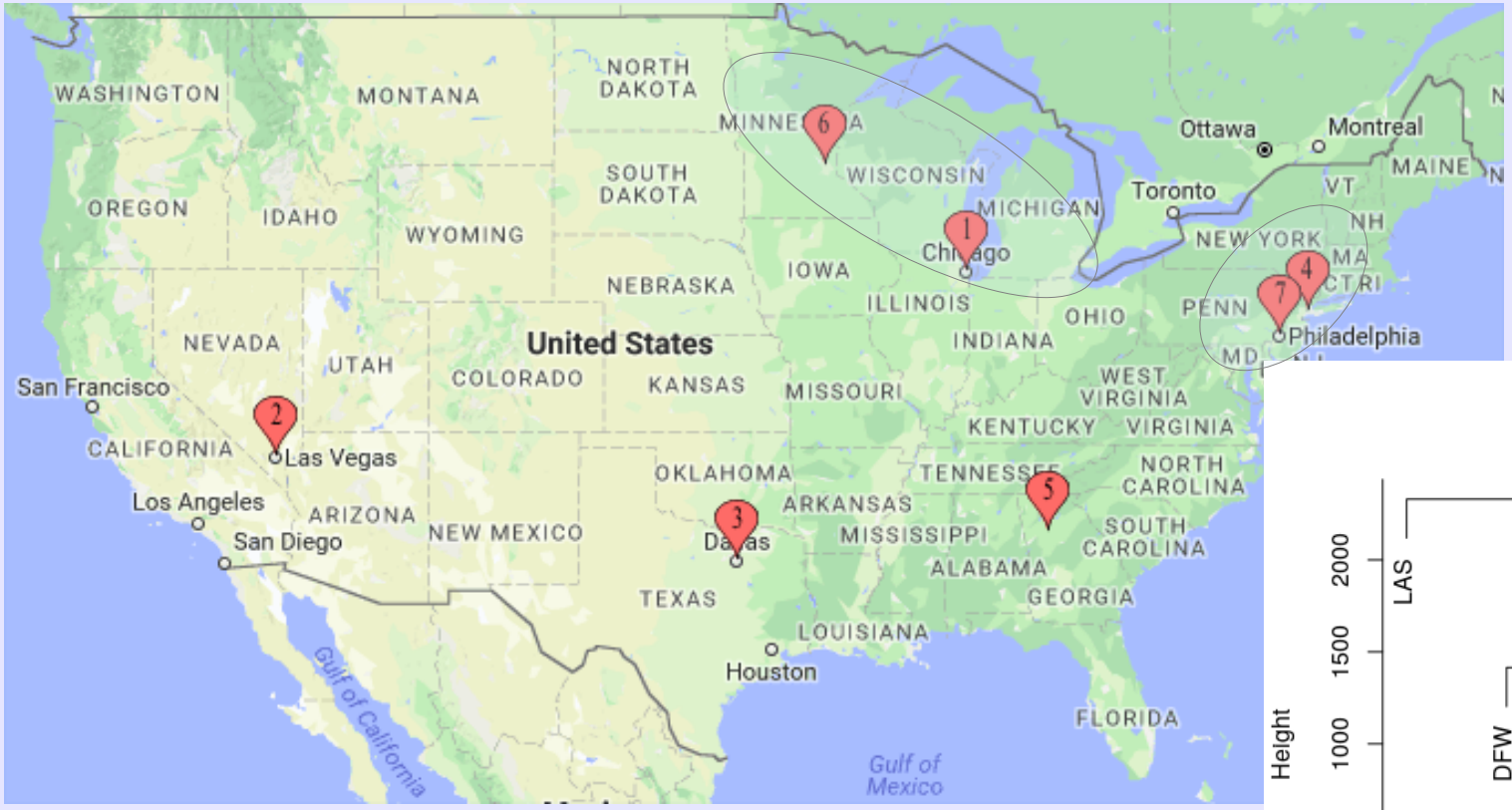
1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

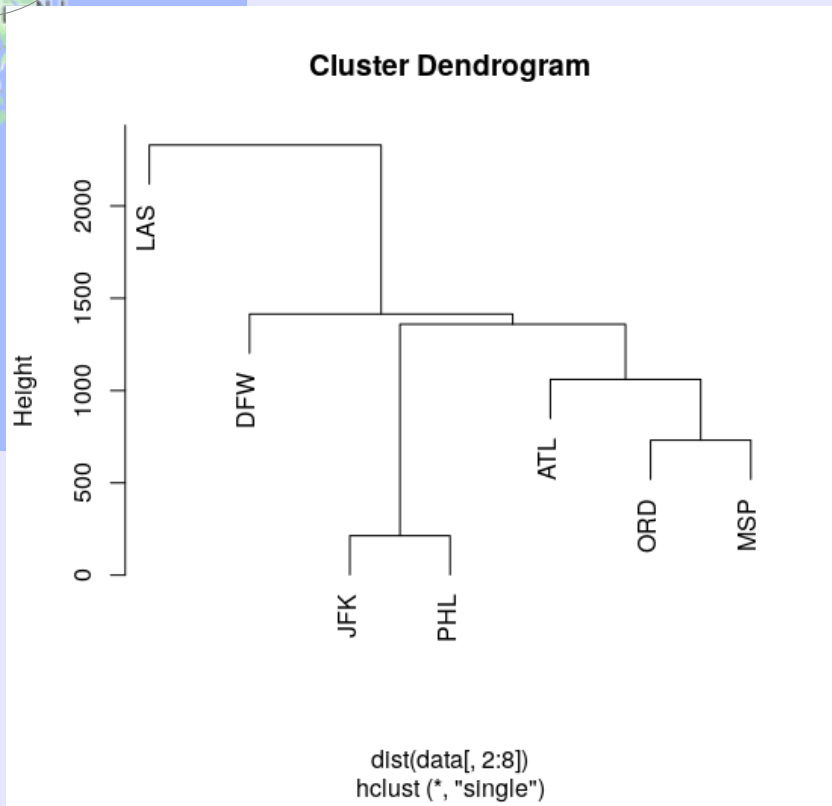
How should this *merge* be done? In other words, how do we define proximity between clusters so that the two closest ones are merged?



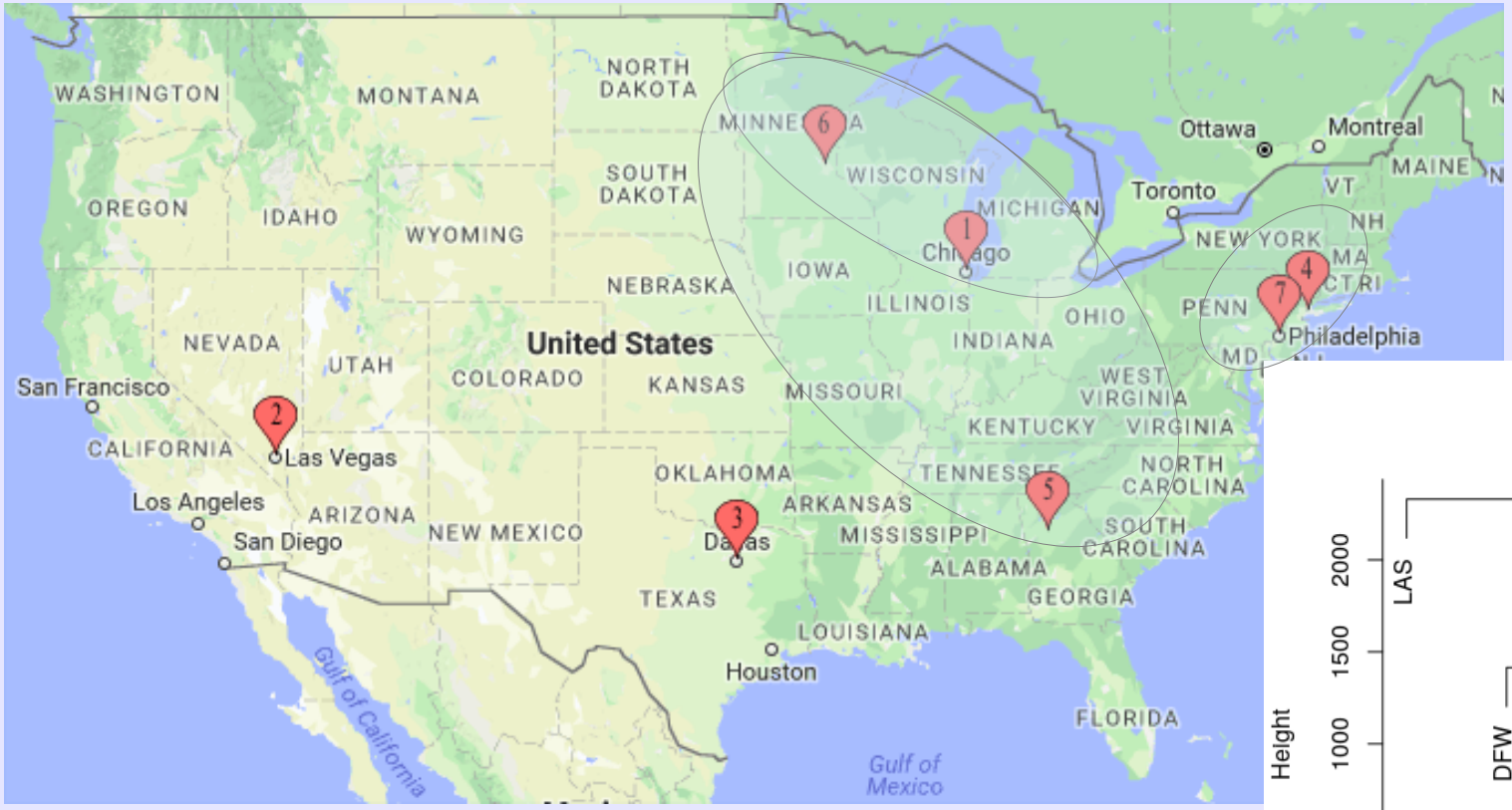
(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

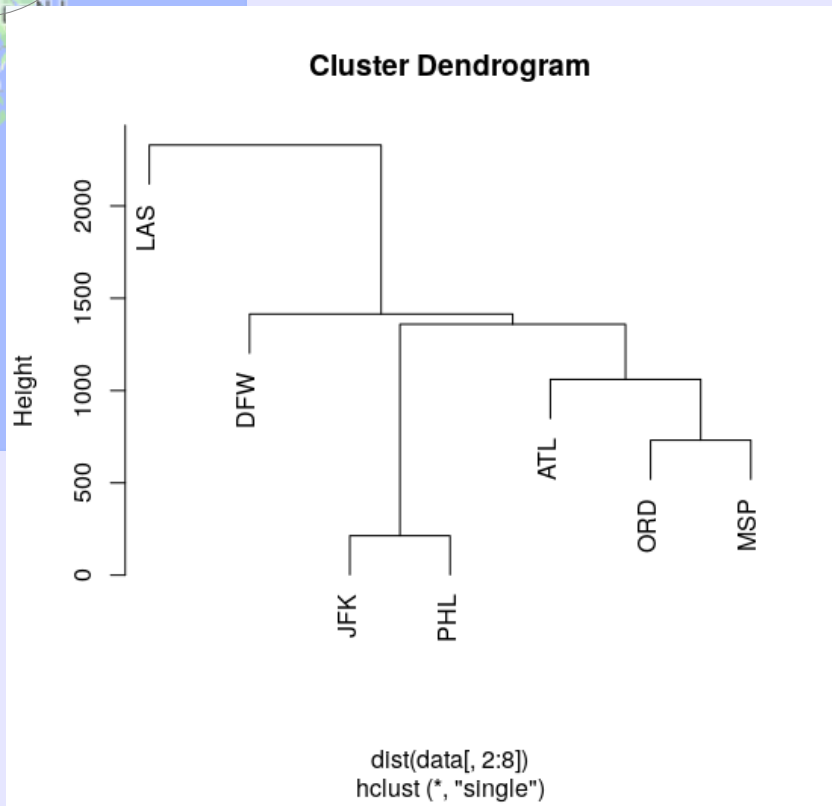
# Hierarchical clustering



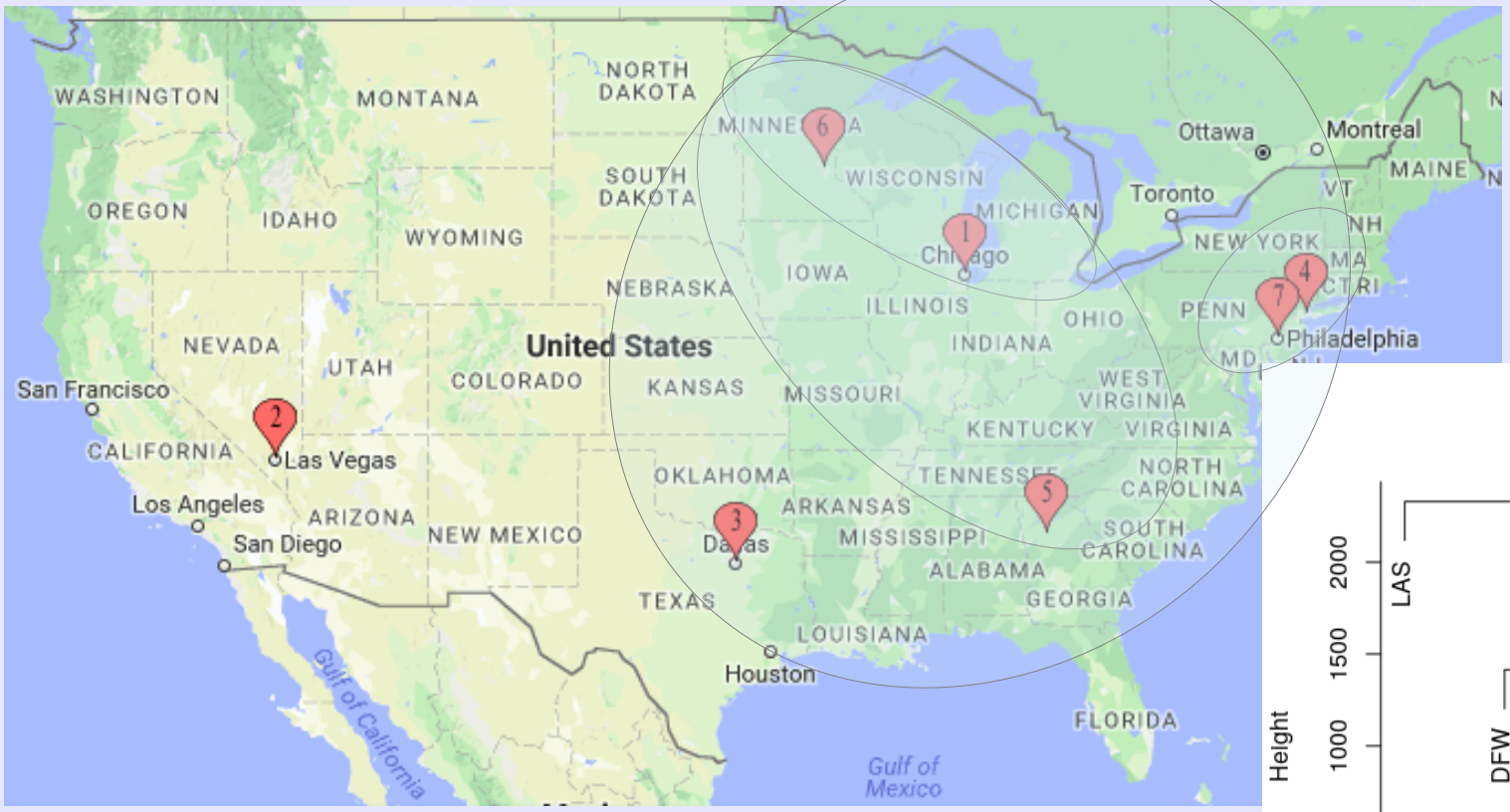



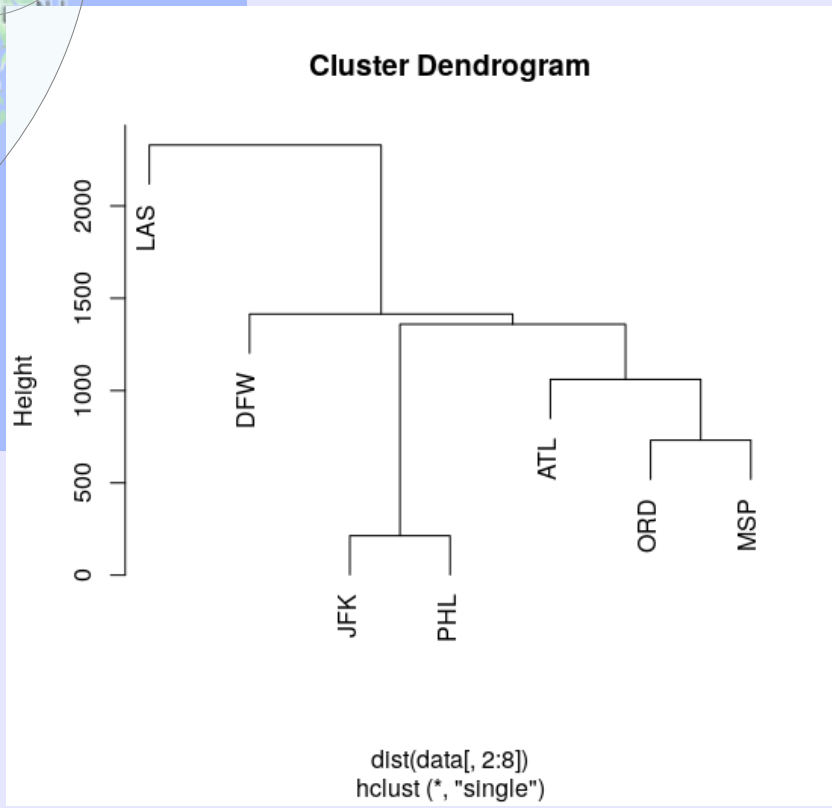
# Hierarchical clustering



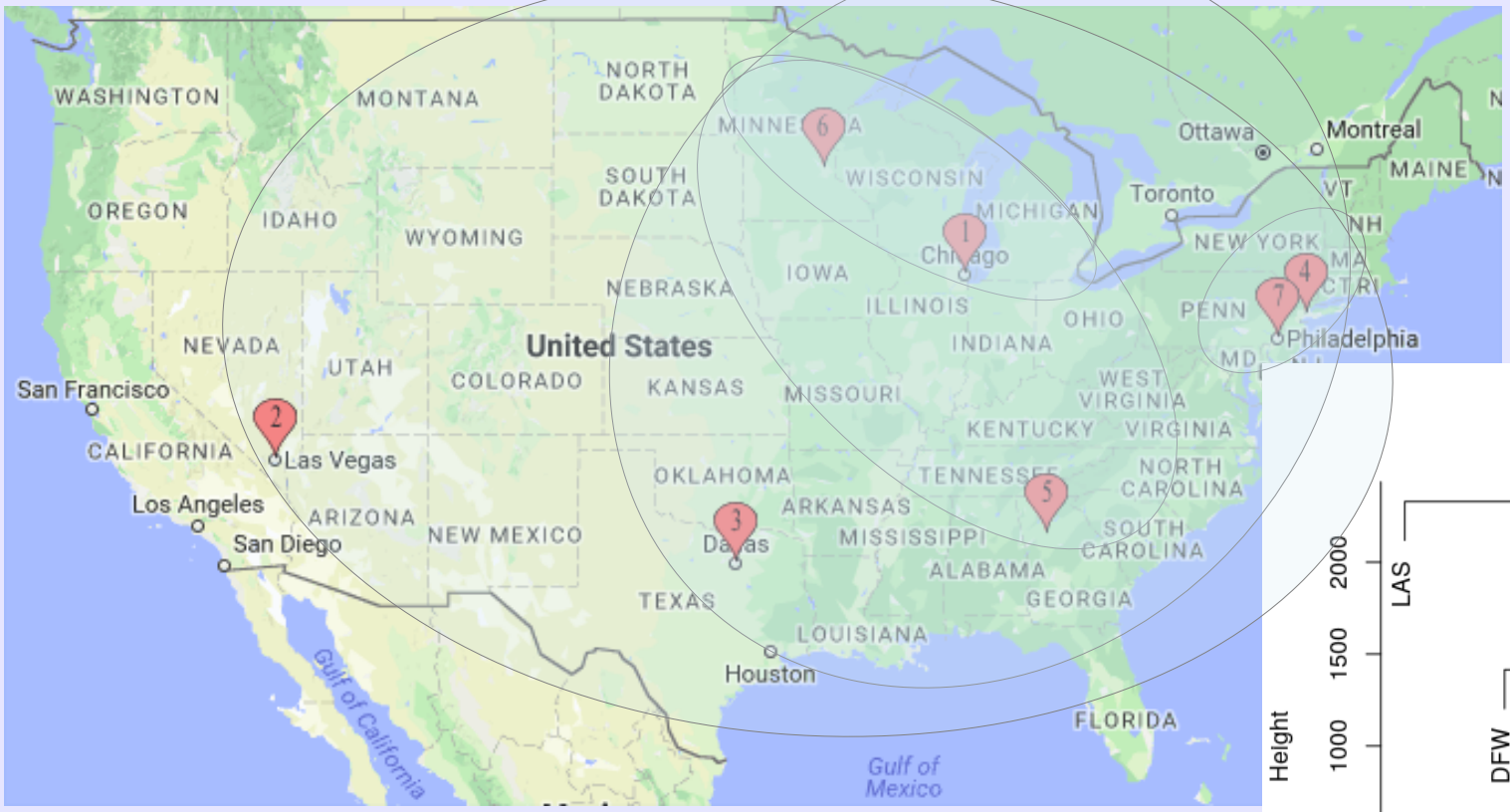



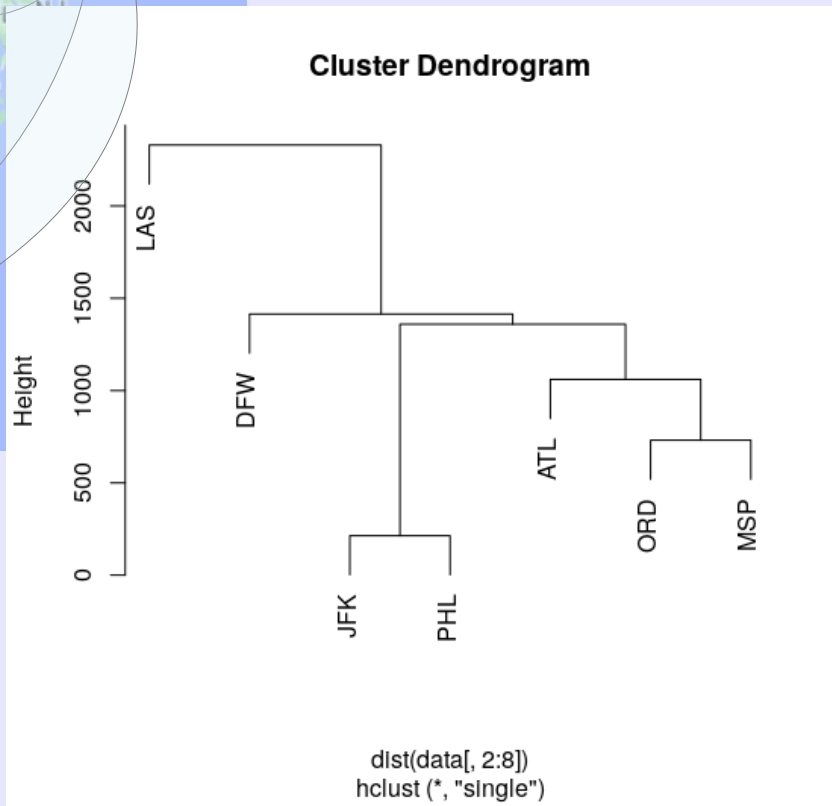
# Hierarchical clustering



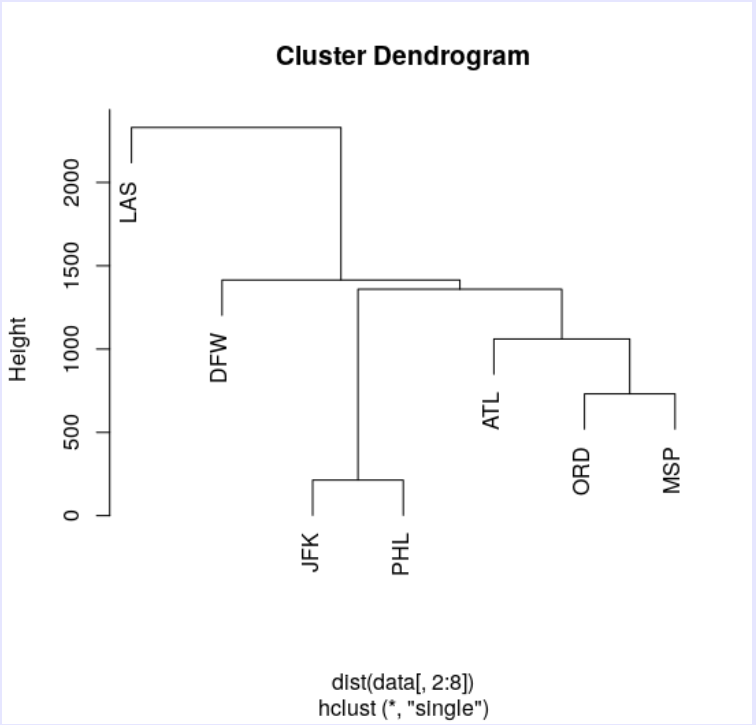
# Hierarchical clustering



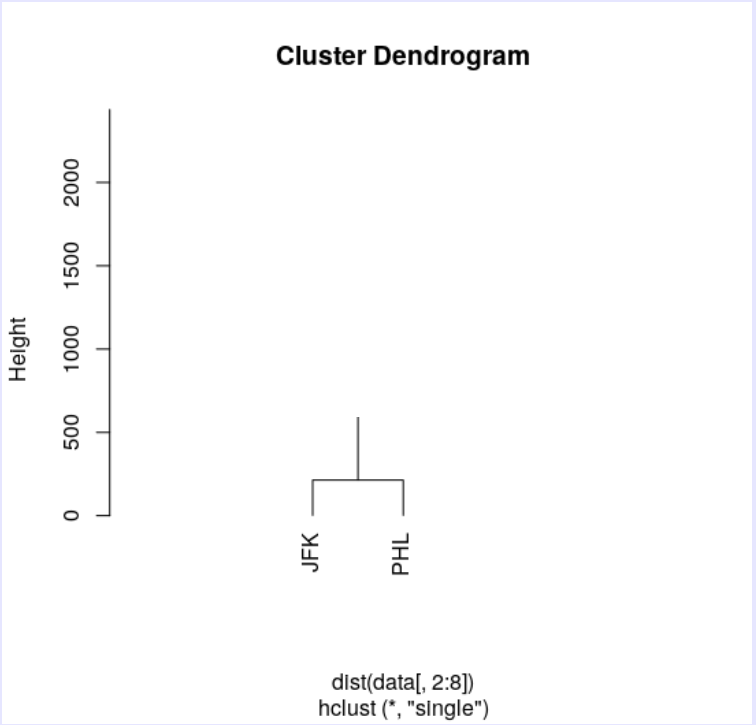

Code: hclust.r

# Hierarchical clustering: Updating the proximity matrix


| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | | | | | | |
| 2 | 3367 | 0 | | | | | |
| 3 | 1610 | 2330 | 0 | | | | |
| 4 | 1678 | 4016 | 2601 | 0 | | | |
| 5 | 1061 | 3408 | 1543 | 1482 | 0 | | |
| 6 | 731 | 2897 | 1414 | 2063 | 1450 | 0 | |
| 7 | 1560 | 4008 | 2518 | 214 | 1359 | 1980 | 0 |

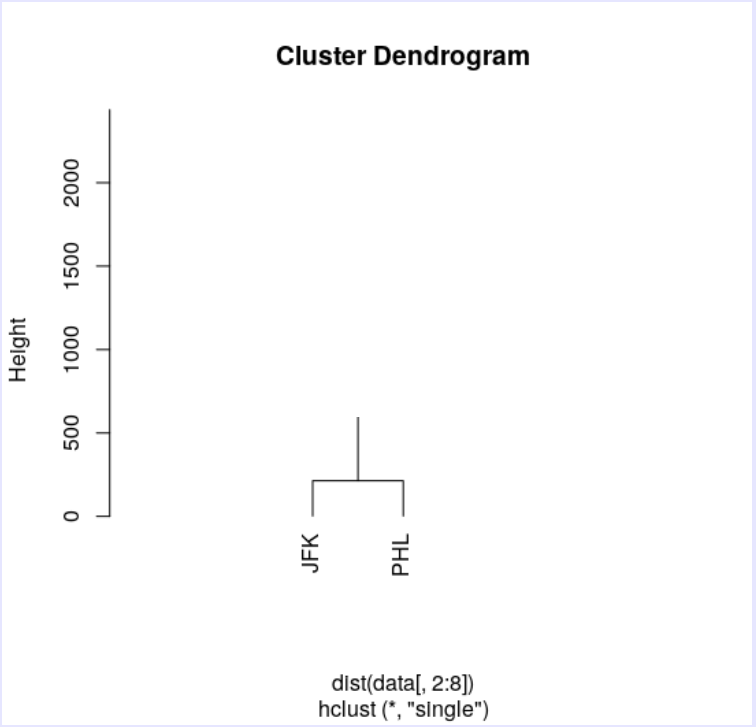
# Hierarchical clustering: Updating the proximity matrix


| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | | | | | | |
| 2 | 3367 | 0 | | | | | |
| 3 | 1610 | 2330 | 0 | | | | |
| 4 | 1678 | 4016 | 2601 | 0 | | | |
| 5 | 1061 | 3408 | 1543 | 1482 | 0 | | |
| 6 | 731 | 2897 | 1414 | 2063 | 1450 | 0 | |
| 7 | 1560 | 4008 | 2518 | 214 | 1359 | 1980 | 0 |

Merge closest two clusters.
Merging strategy: MIN (single link)

# Hierarchical clustering: Updating the proximity matrix


| | 1 | 2 | 3 | 4,7 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | | | | | |
| 2 | 3367 | 0 | | | | |
| 3 | 1610 | 2330 | 0 | | | |
| 4,7 | 1560 | 4008 | 2518 | 0 | | |
| 5 | 1061 | 3408 | 1543 | 1359 | 0 | |
| 6 | 731 | 2897 | 1414 | 1980 | 1450 | 0 |

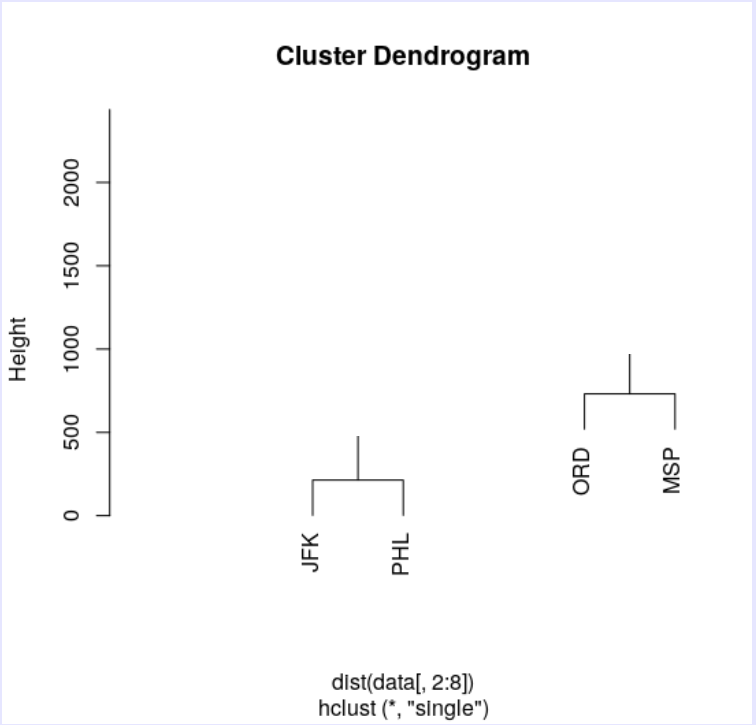
D({4,7}→{1}) = min(D({4}→{1}), D({7}→{1}))
= min(1678, 1560) = 1560

…

D({4,7}→{5}) = min(D({4}→{5}), D({7}→{5}))
= min(1482, 1359) = 1359

D({4,7}→{6}) = min( D({4}→{6}), D({7}→{6}))
= min(2063, 1980) = 1980

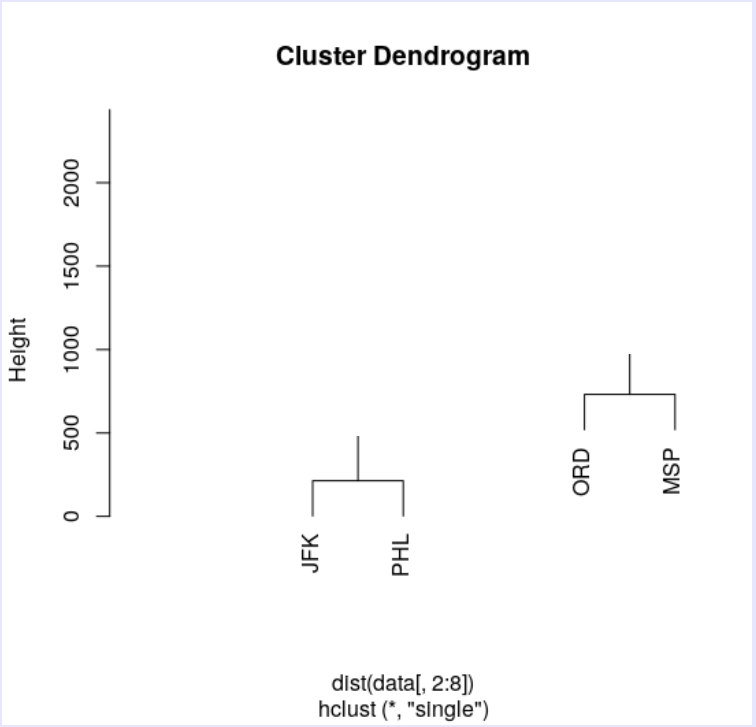
# Hierarchical clustering: Updating the proximity matrix


| | 1 | 2 | 3 | 4,7 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | | | | | |
| 2 | 3367 | 0 | | | | |
| 3 | 1610 | 2330 | 0 | | | |
| 4,7 | 1560 | 4008 | 2518 | 0 | | |
| 5 | 1061 | 3408 | 1543 | 1359 | 0 | |
| 6 | 731 | 2897 | 1414 | 1980 | 1450 | 0 |

Merge closest two clusters.

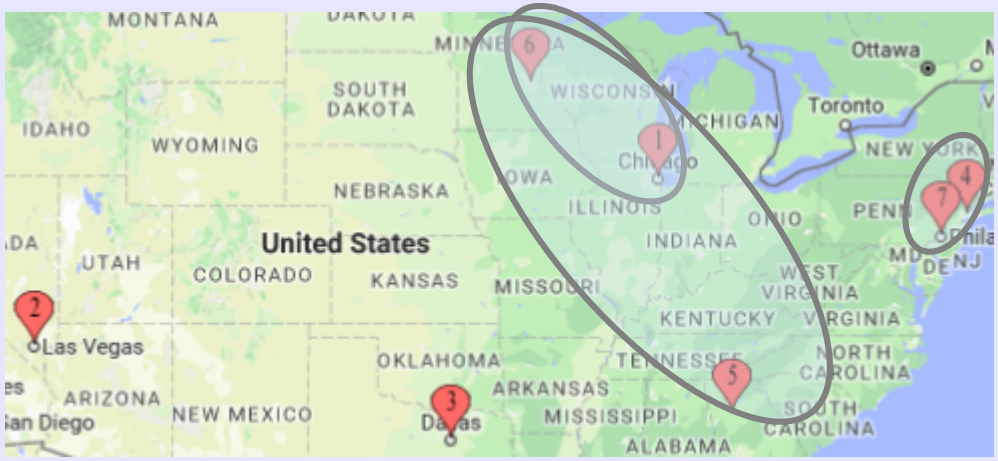
# Hierarchical clustering: Updating the proximity matrix

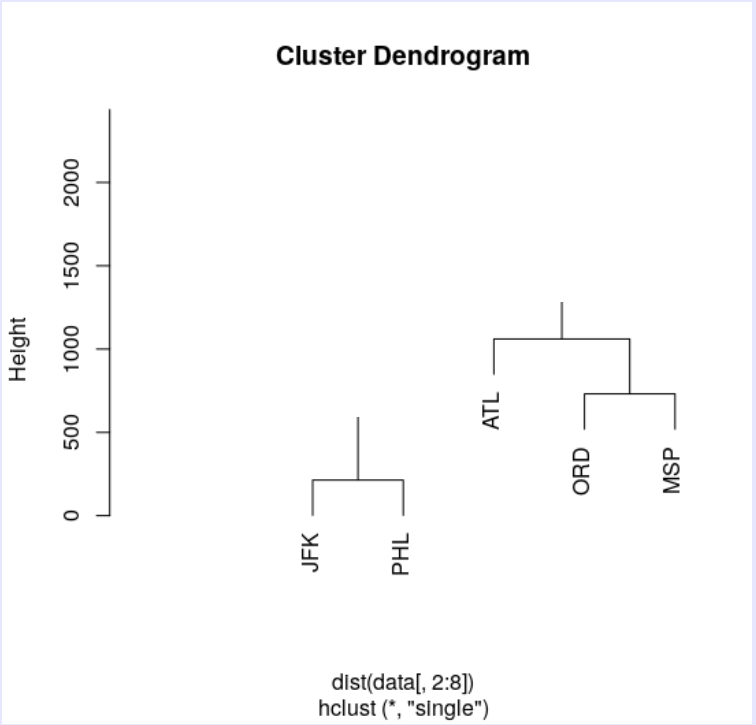




| | 1,6 | 2 | 3 | 4,7 | 5 |
|---|---|---|---|---|---|
| 1,6 | 0 | | | | |
| 2 | 2897 | 0 | | | |
| 3 | 1414 | 2330 | 0 | | |
| 4,7 | 1560 | 4008 | 2518 | 0 | |
| 5 | 1061 | 3408 | 1543 | 1359 | 0 |

D({1,6}→{2}) = min(D({1}→{2}), D({6}→{2}))
= min(3367, 2897) = 2897
…
D({1,6}→{5}) = min(D({1}→{5}), D({6}→{5}))
= min(1061, 1450) = 1061
…
D({1,6}→{4,7}) = min(D({1}→{4}), D({1}→{7}),
D({6}→{4}, D({6}→{7}))
= min(1678, 1560, 2063, 1980) = 1560
...

# Hierarchical clustering: Updating the proximity matrix

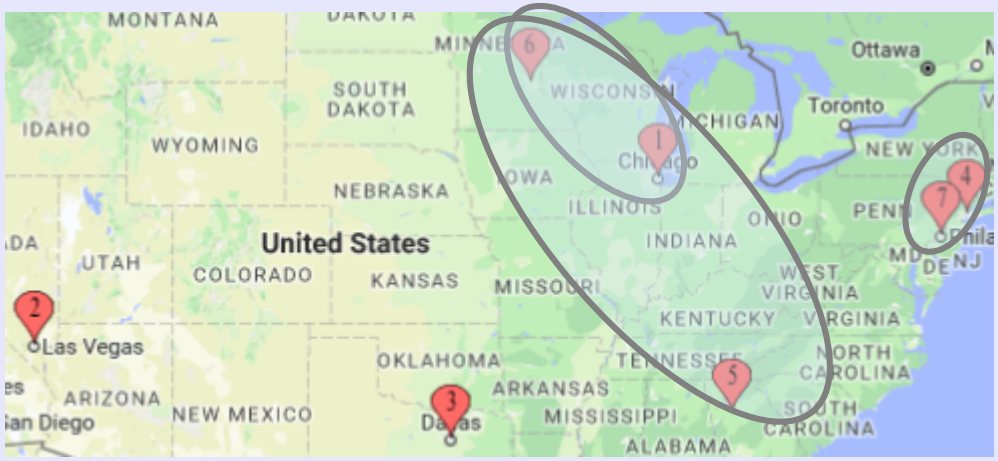


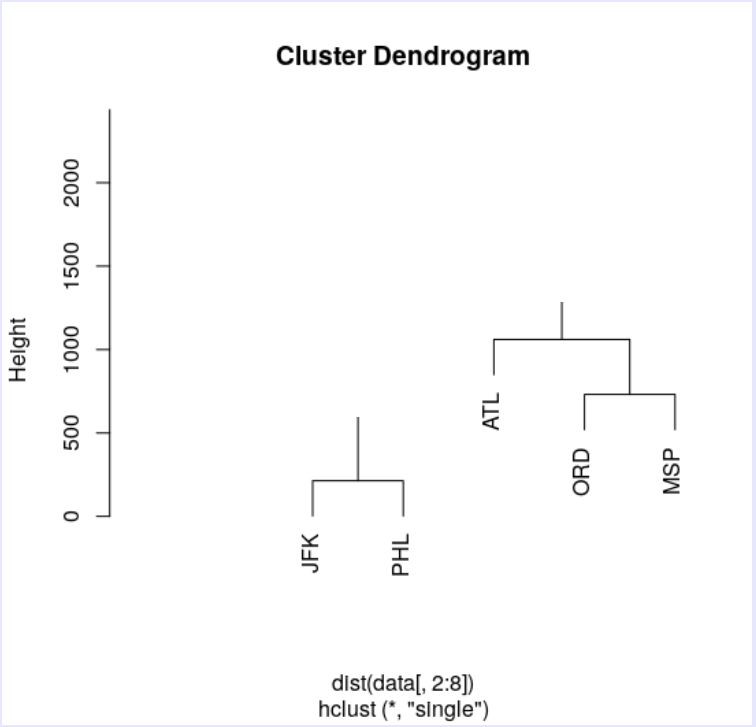




| | 1,6 | 2 | 3 | 4,7 | 5 |
|---|---|---|---|---|---|
| 1,6 | 0 | | | | |
| 2 | 2897 | 0 | | | |
| 3 | 1414 | 2330 | 0 | | |
| 4,7 | 1560 | 4008 | 2518 | 0 | |
| 5 | 1061 | 3408 | 1543 | 1359 | 0 |

Merge closest two clusters.

# Hierarchical clustering: Updating the proximity matrix

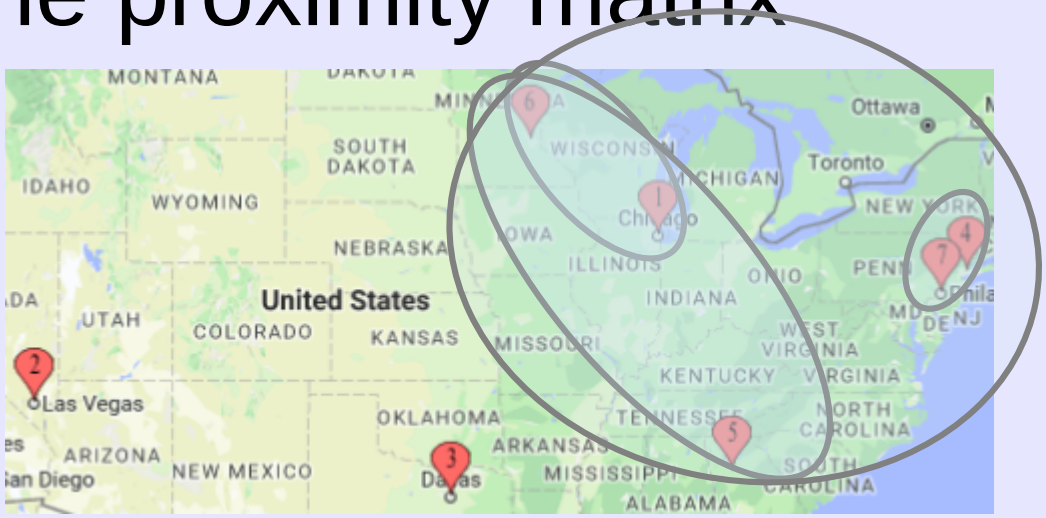


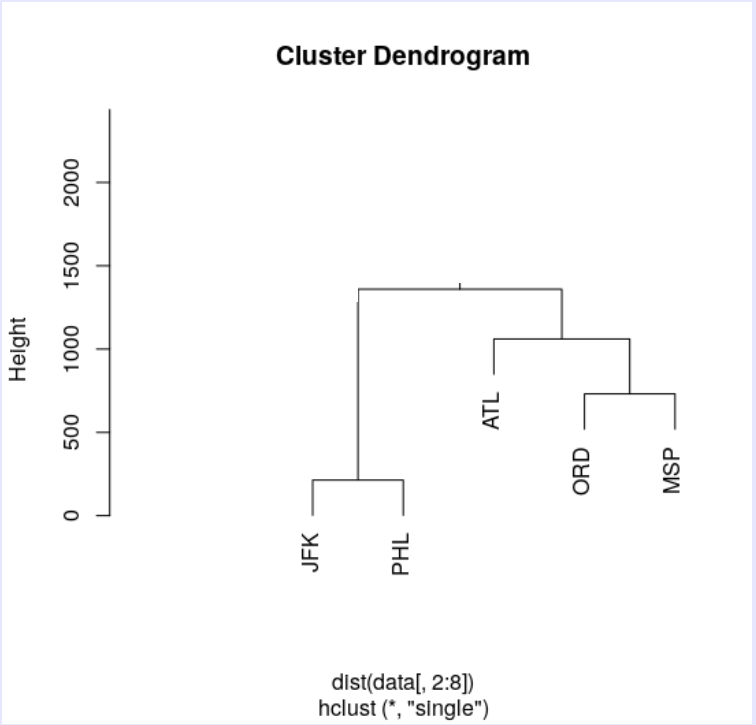




| | 1,6,5 | 2 | 3 | 4,7 |
|---|---|---|---|---|
| 1,6,5 | 0 | | | |
| 2 | 2897 | 0 | | |
| 3 | 1414 | 2330 | 0 | |
| 4,7 | 1359 | 4008 | 2518 | 0 |

D({1,6,5}→{2}) = min(D({1}→{2}), D({6}→{2}), D({5}→{2}))
= min(3367, 2897, 3408) = 2897
…
D({1,6,5}→{4,7}) = min(D({1}→{4}), D({6}→{4}), D({5}→{4}),
D({1}→{7}), D({6}→{7}), D({5}→{7}))
= min(1678, 2063, 1482, 1560, 1980, 1359) = 1359
…

# Hierarchical clustering: Updating the proximity matrix

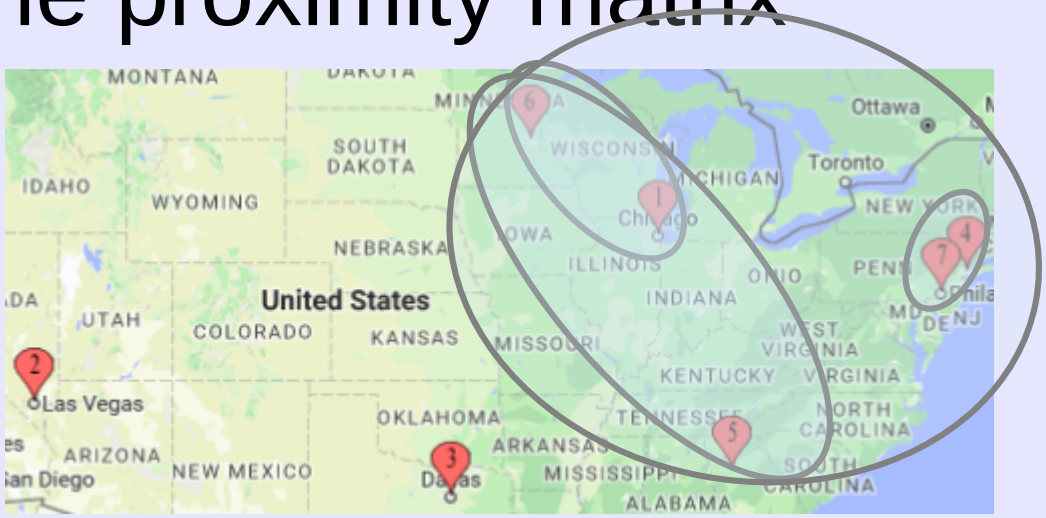


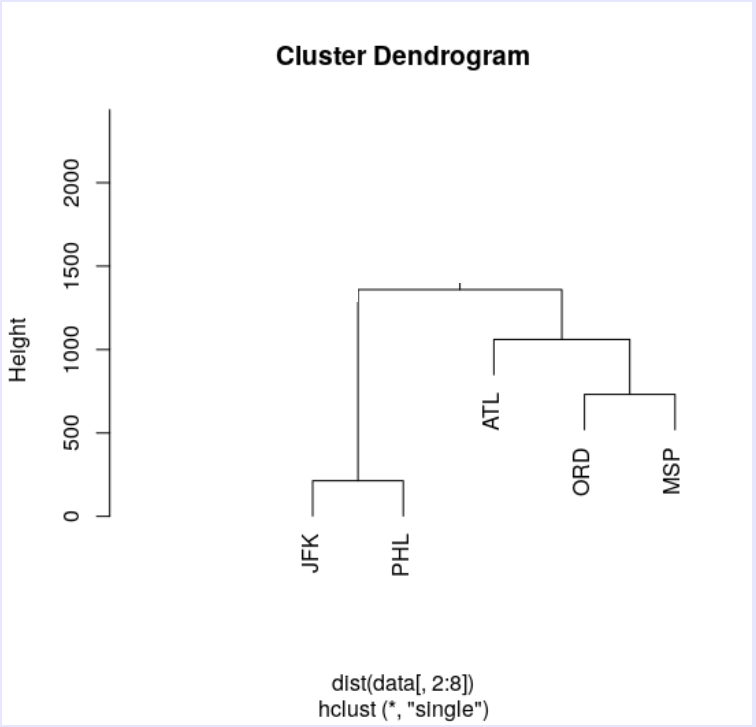




| | 1,6,5 | 2 | 3 | 4,7 |
|---|---|---|---|---|
| 1,6,5 | 0 | | | |
| 2 | 2897 | 0 | | |
| 3 | 1414 | 2330 | 0 | |
| 4,7 | 1359 | 4008 | 2518 | 0 |

Merge closest two clusters.

# Hierarchical clustering: Updating the proximity matrix



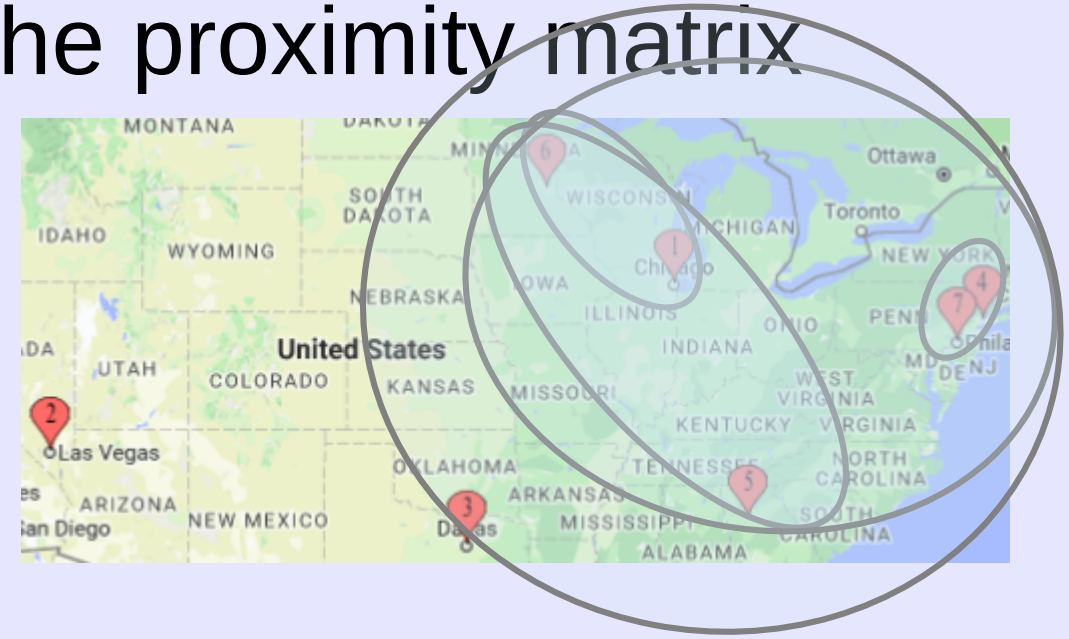



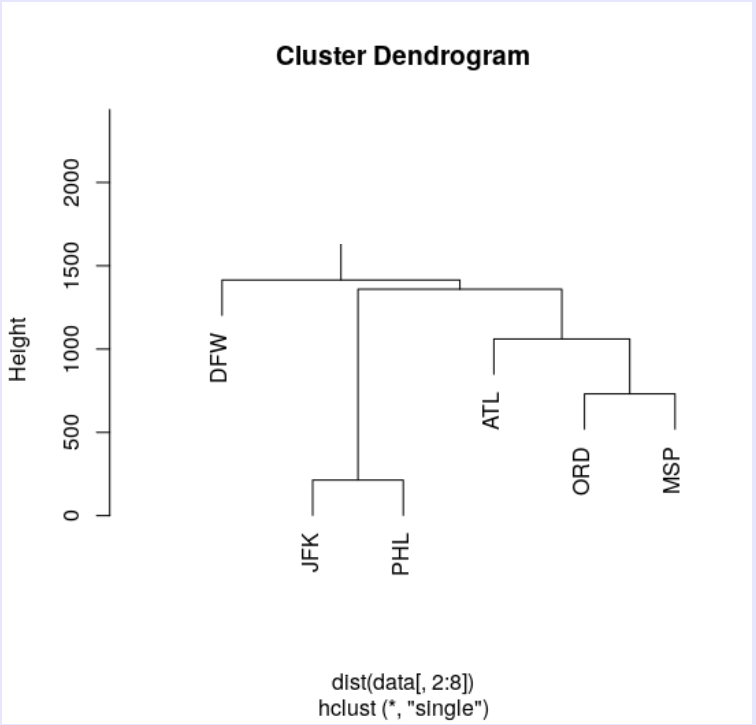

| | 1,6,5,4,7 | 2 | 3 |
|---|---|---|---|
| 1,6,5,4,7 | 0 | | |
| 2 | 2897 | 0 | |
| 3 | 1414 | 2330 | 0 |

D({1,6,5,4,7}→{2}) = min(D({1}→{2}), D({6}→{2}), D({5}→{2}), D({4}→{2}), D({7}→{2}))
= min(3367, 2897, 3408, 4016, 4008) = 2897

D({1,6,5,4,7}→{3}) = min(D({1}→{3}), D({6}→{3}), D({5}→{3}), D({4}→{3}), D({7}→{3}))
= min(1610, 1414, 1543, 2601, 2518) = 1414

# Hierarchical clustering: Updating the proximity matrix



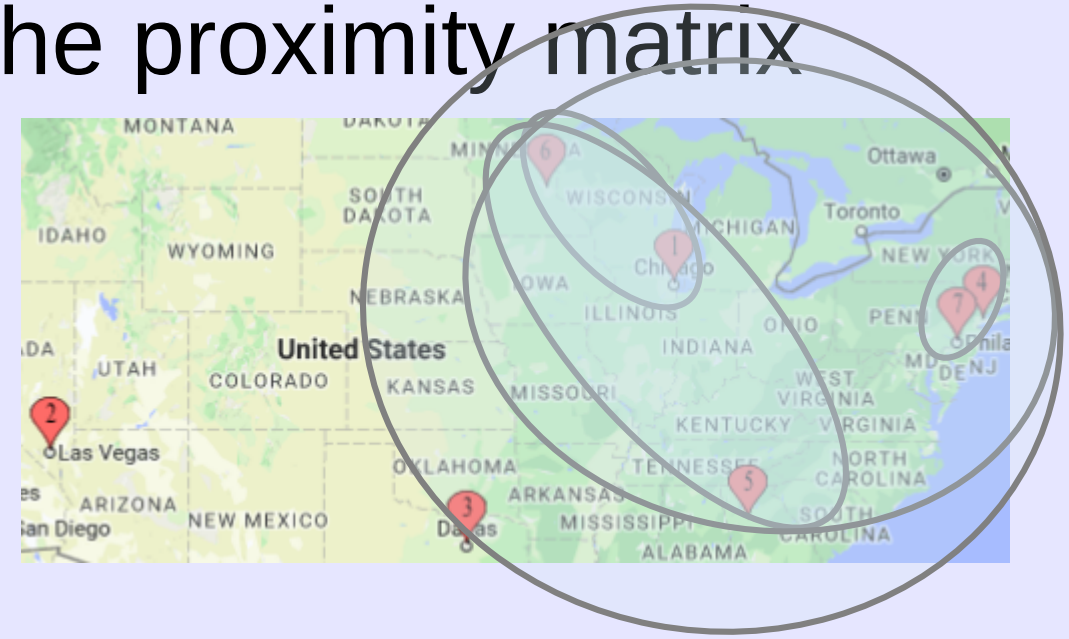



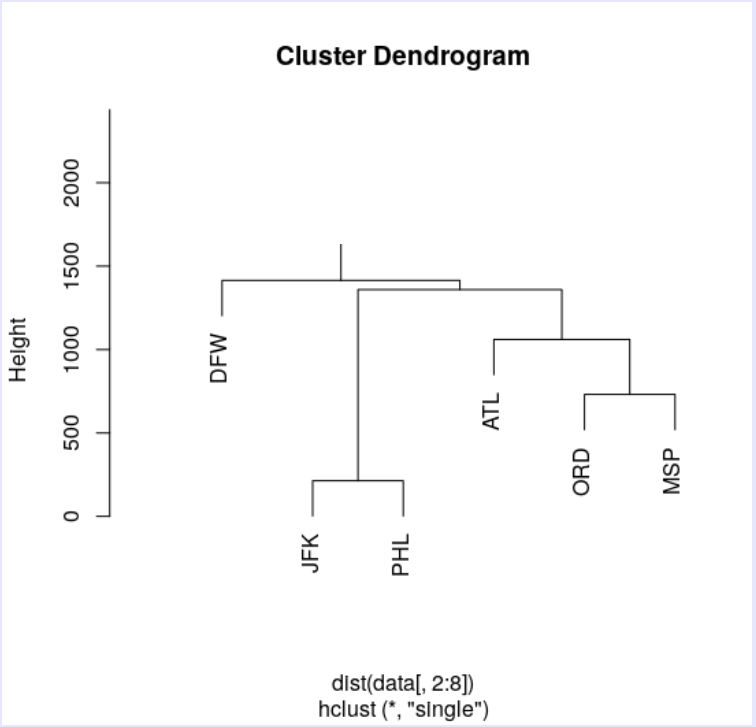

| | 1,6,5,4,7 | 2 | 3 |
|---|---|---|---|
| 1,6,5,4,7 | 0 | | |
| 2 | 2897 | 0 | |
| 3 | 1414 | 2330 | 0 |

D({1,6,5,4,7}→{2}) = min(D({1}→{2}), D({6}→{2}), D({5}→{2}), D({4}→{2}), D({7}→{2}))
= min(3367, 2897, 3408, 4016, 4008) = 2897

D({1,6,5,4,7}→{3}) = min(D({1}→{3}), D({6}→{3}), D({5}→{3}), D({4}→{3}), D({7}→{3}))
= min(1610, 1414, 1543, 2601, 2518) = 1414

# Hierarchical clustering: Updating the proximity matrix

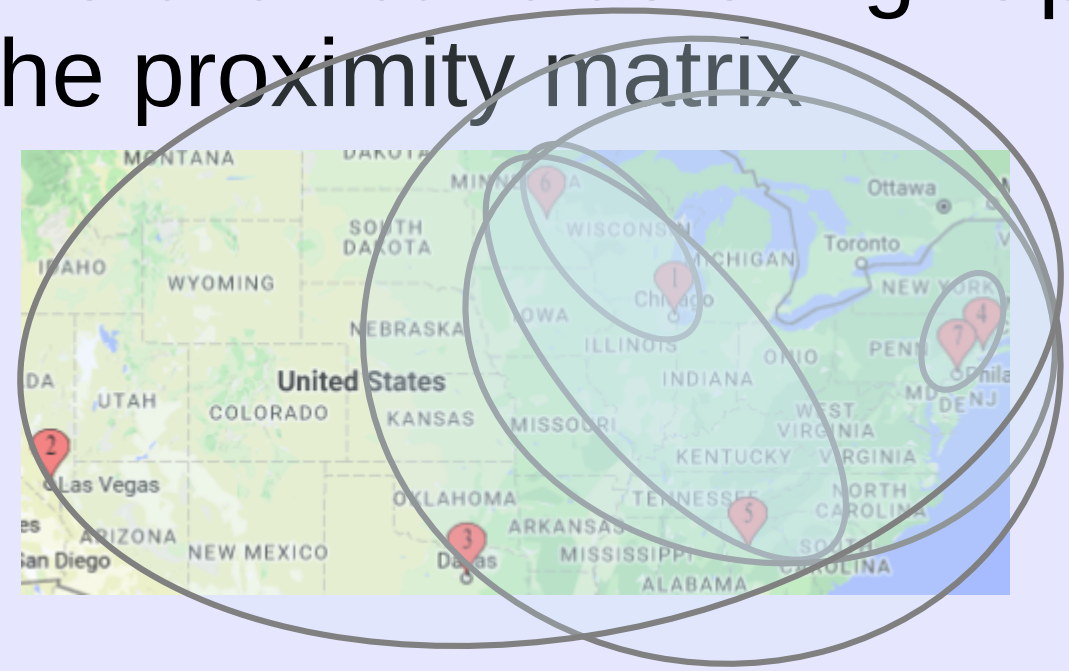


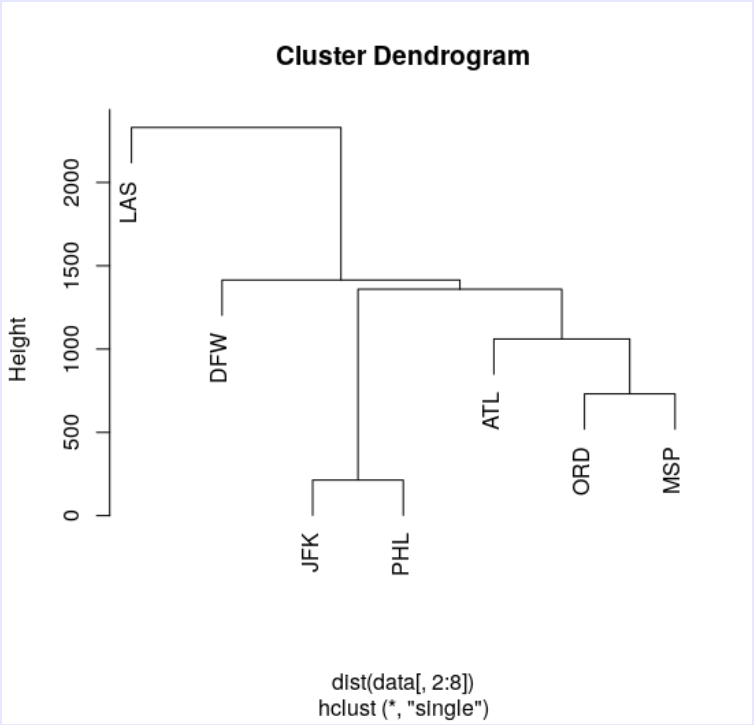




| | 1,6,5,4,7,3 | 2 |
|---|---|---|
| 1,6,5,4,7,3 | 0 | |
| 2 | 2897 | 0 |

Merge closest two clusters.

# Hierarchical clustering: Updating the proximity matrix




| | 1,6,5,4,7,3,2 |
|---|---|
| 1,6,5,4,7,3,2 | 0 |

Merge closest two clusters...until only one cluster remains.

# Hierarchical clustering: Updating the proximity matrix

- For complete linkage and group average, proceed as before except update the proximity matrix using **MAX** and **AVG**, respectively.

# Hierarchical clustering: Complexity

- Space complexity:
  - *m* is number of data points; storage required: $1/2m^2$ => $O(m^2)$.
- Time complexity:
  - *m* is the number of data points; $O(m^2)$ required for computing the proximity matrix.
  - If distances from each cluster to all other clusters are stored in a heap, overall time required for hierarchical clustering is $O(m^2 \log m)$.

# Hierarchical clustering: Practical issues

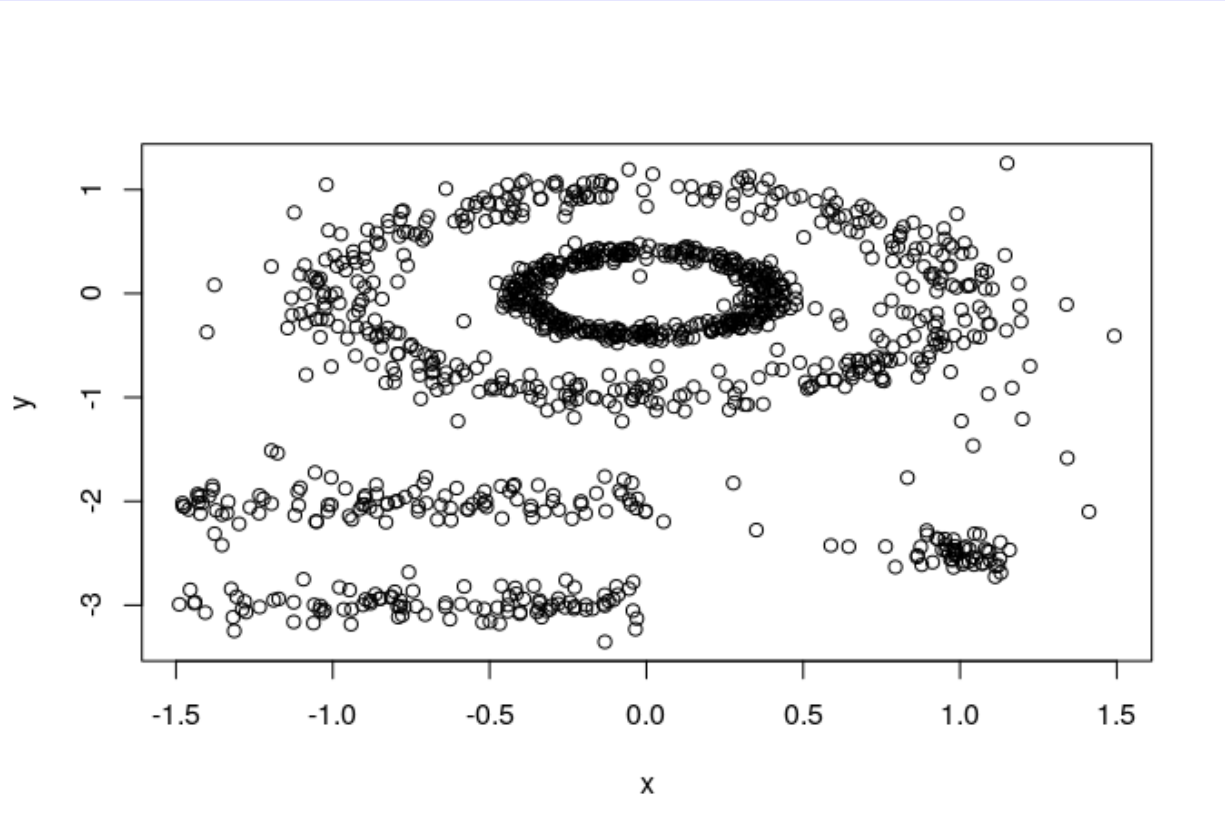
- Scaling matters if attributes are wide ranges.
- What dissimilarity measure should be used?
  - Minkowski with different values for R (Euclidean, ...)
  - Correlation-based
- What type of linkage to use?
- How many clusters to choose? Difficult problem as there is no agreed upon method. Sometimes the domain expert helps, other times study the data.
- Merging decisions are final (unlike k-means where observations may belong to different centroids over time).

# Hierarchical clustering: Using as a supervised learning method

- Even though clustering in general is considered an unsupervised learning method, it can be used in supervised learning mode.
- Cluster indicate a class label and each object in a certain cluster belongs to that class.
- Error measures for supervised clustering are the same as classification.
- Code: hclust-iris.r

# Density-based clustering: dbscan

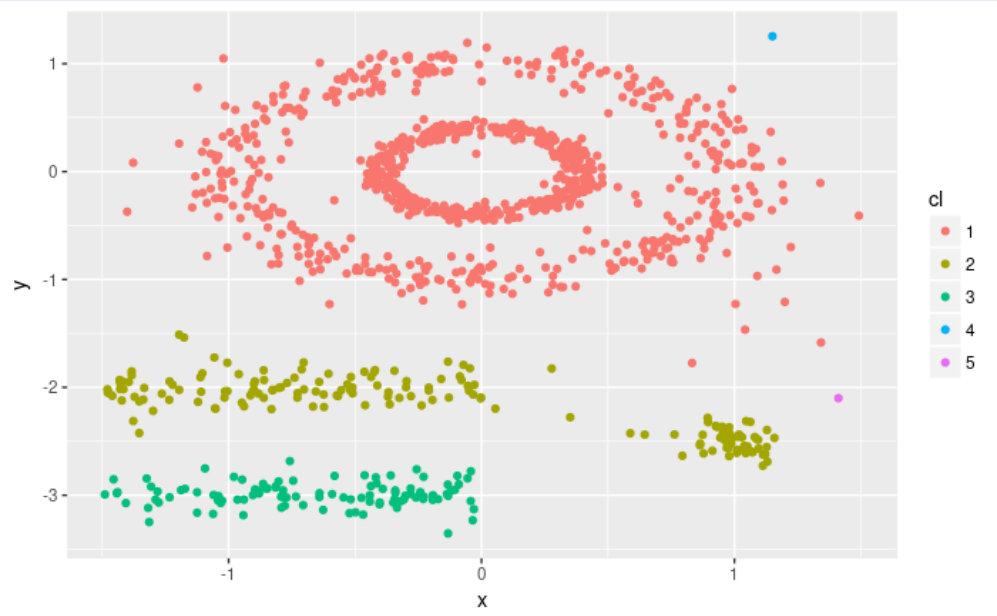
- K-means and hierarchical clustering do not gracefully handle non-globular clusters as shown on the left.

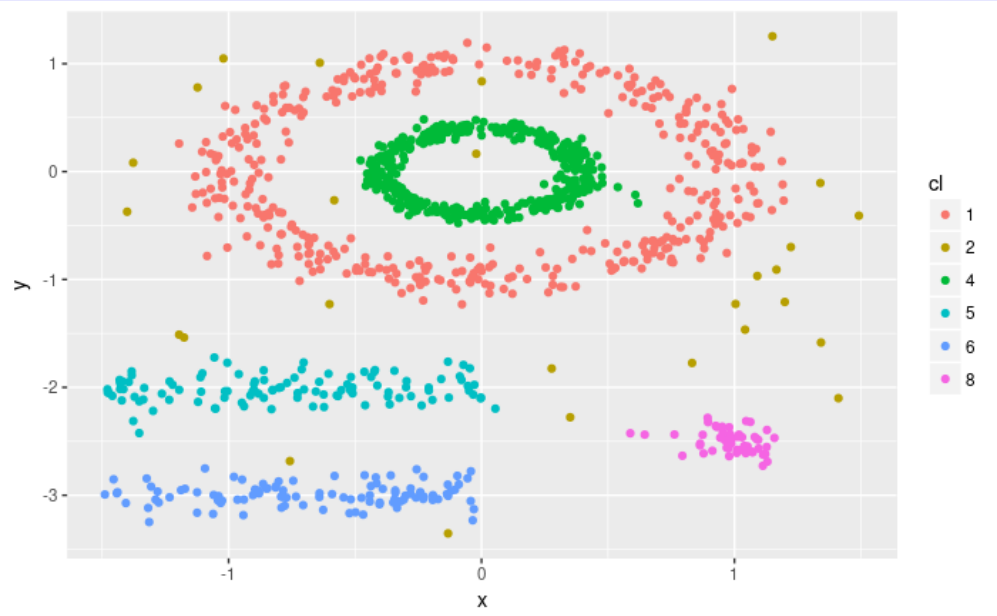



  - They will **find** clusters, but the resulting clusters may not be what we need.

# Density-based clustering: Miscellaneous

- Hierarchical clustering do on globular data:




- Observations:
  - Hierarchical clustering is not able to eliminate what would be called "noise" points in DBSCAN. So these become part of a cluster.
  - With 5 clusters hierarchical clustering is unable to discern the two nested clusters. It considers them as one.

Code:
hierarchical-clustering-globular-data.Rmd