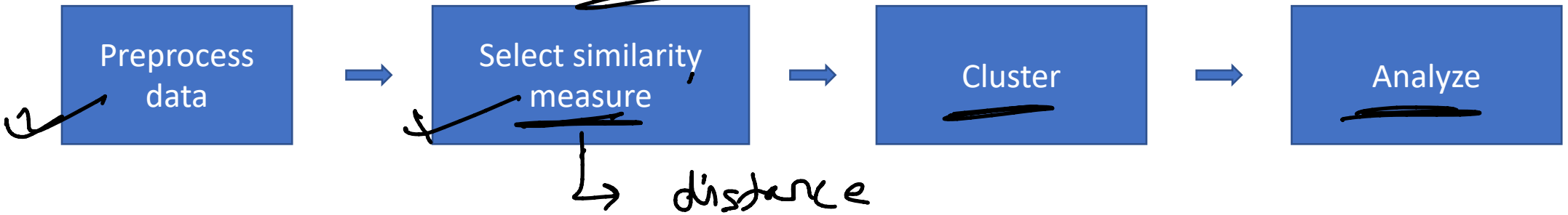
Clustering

Clustering

- A form of exploratory data analysis (EDA) where observations are divided into meaningful groups that share common characteristics (features)
- Grouping of objects based on the information found in the data describing the objects or their relationship
- The goal is that objects in one group should be similar to each other but different from objects in another group
- Deals with finding a structure in a collection of unlabeled data

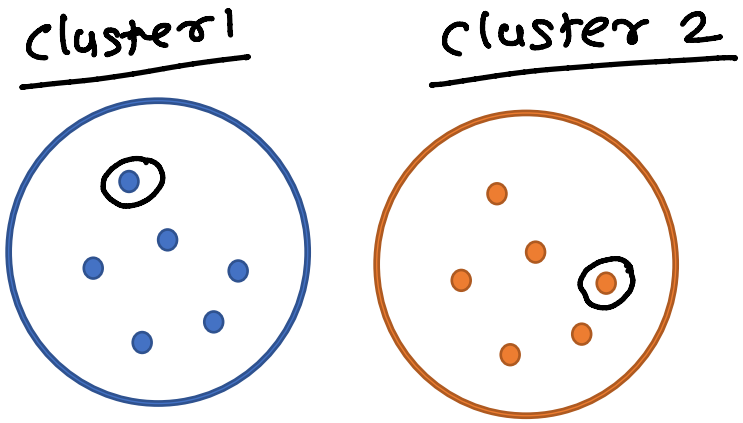


Clustering use cases

- Marketing
  - Discovering groups in customer databases like who makes long-distance calls or who are earning more or who are spending more
- Insurance
  - Identifying groups of insurance policy holder with high claim rate
- Land use
  - Identification of areas of similar land use in GIS database

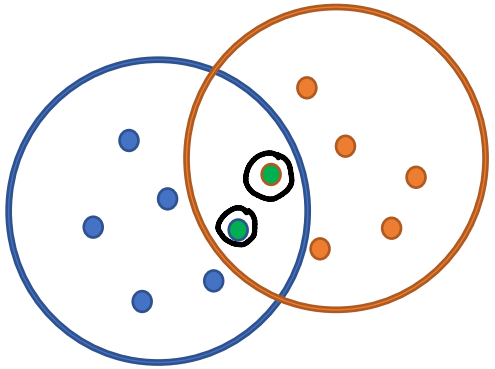
Clustering - Types

- Exclusive clustering
  - An item belongs exclusively to one cluster and not several
  - E.g. K-Means clustering



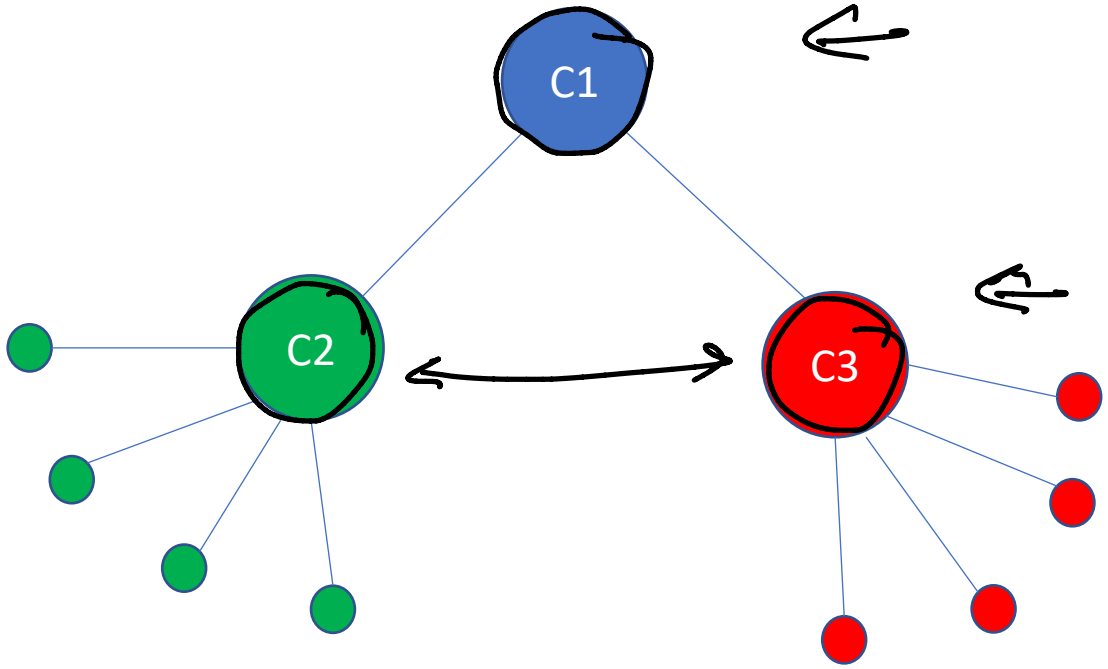
Clustering - Types

- Overlapping clustering
  - An Item can belong to multiple clusters
  - Its degree of association with each cluster is known
  - E.g. Fuzzy/C-means clustering



Clustering - Types

- Hierarchical clustering
  - When two clusters have a parent child relationship
  - It forms a tree like structure
  - E.g. Hierarchical clustering



K-Means Clustering

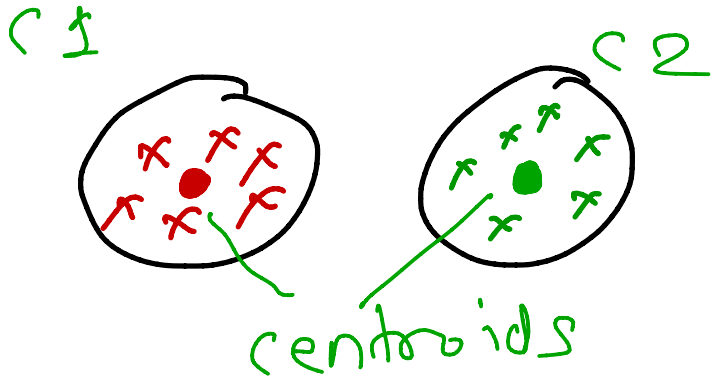
K-Means Clustering

- Used when you have unlabelled data (i.e., data without defined categories or groups)
- The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*

K = # clusters / groups

K-Means Clustering

- The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided
- Data points are clustered based on feature similarity (distance)
- The results of the *K*-means clustering algorithm are
  - The centroids of the *K* clusters, which can be used to label new data
  - Labels for the training data (each data point is assigned to a single cluster)

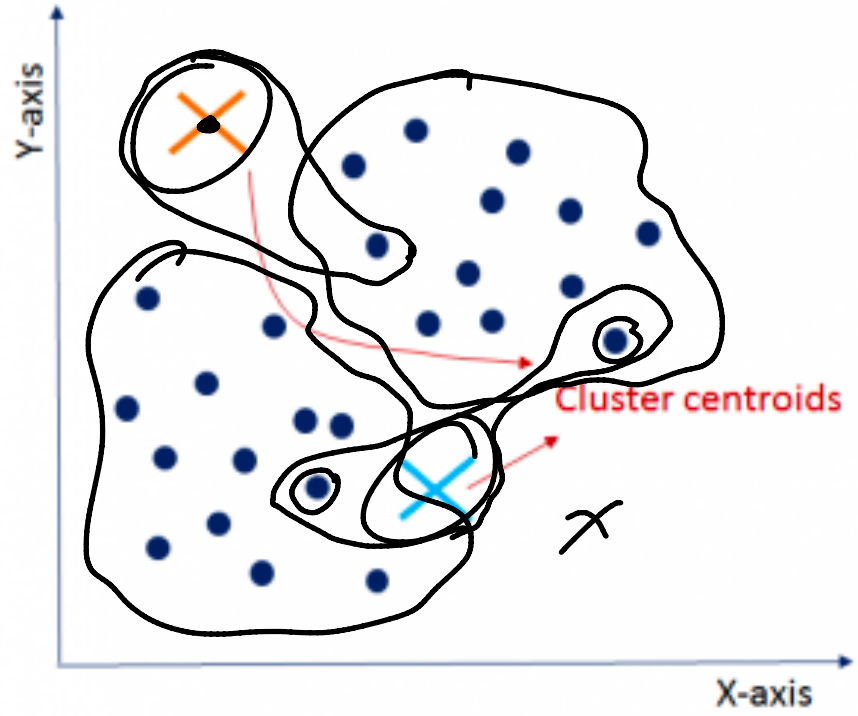


K-Means Clustering - Algorithm

K = 2

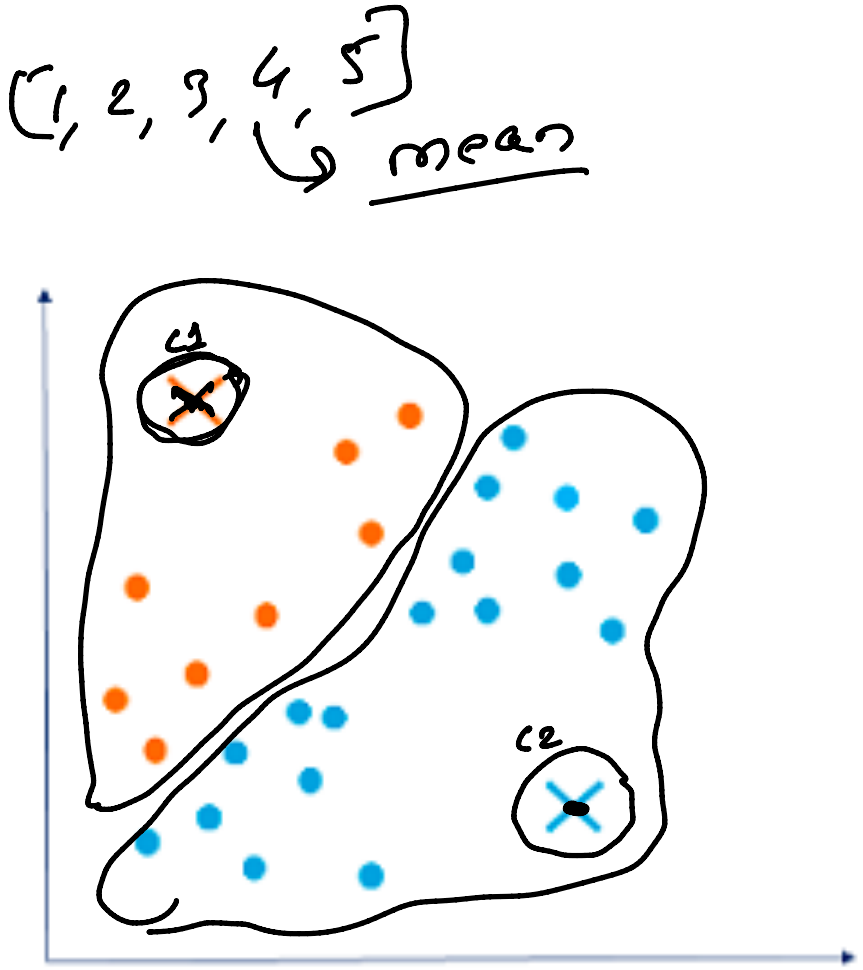
- **Initialization**
  - randomly initialise two points called the cluster centroids

unlabelled

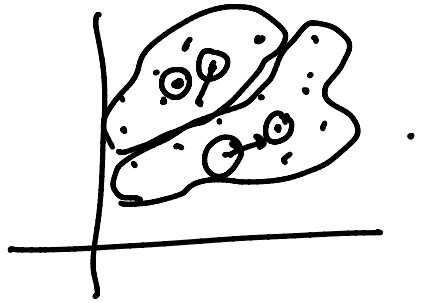


K-Means Clustering - Algorithm

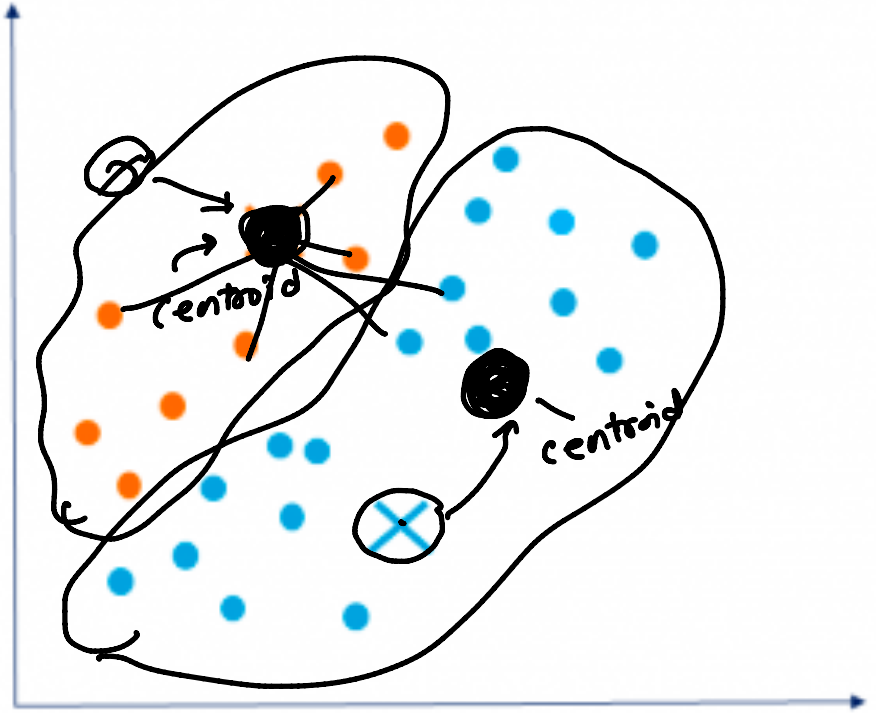
- **Cluster Assignment**
  - Compute the distance between both the points and centroids
  - Depending on the minimum distance from the centroid divide the points into two clusters



K-Means Clustering - Algorithm

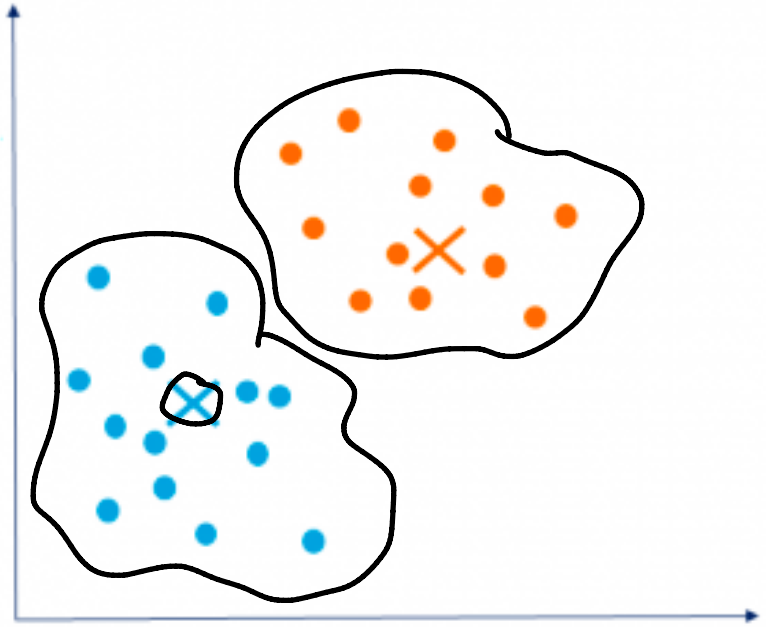


- **Move Centroid**
  - Consider the older centroids are data points
  - Take the older centroid and iteratively reposition them for optimization
- **Optimization**
  - Repeat the steps until the cluster centroids stop changing the positio



K-Means Clustering - Algorithm

- **Convergence**
  - Finally, k-means clustering algorithm converges and divides the data points into two clusters clearly visible in multiple clusters

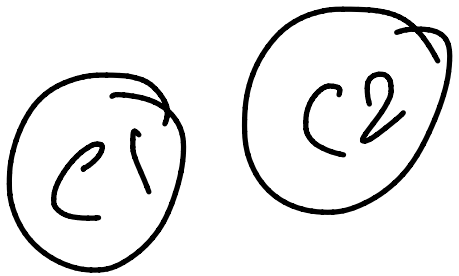


K-Means Clustering - Example

- Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

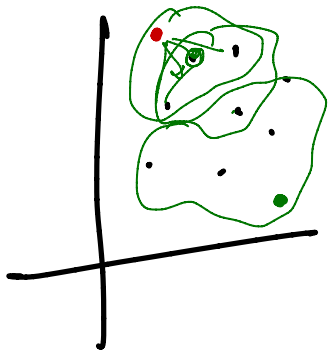
N = 19

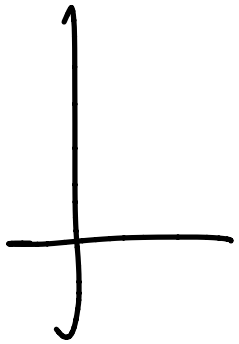


K-Means Clustering - Example

K = 2

- Initial clusters (random centroid or average)





k = 2

$c_1 = 16$

$c_2 = 22$

every Point

$$Distance\ 1 = |x_i - c_1|$$

$$Distance\ 2 = |x_i - c_2|$$

K-Means Clustering - Example

- **Iteration I**

Before:

$c_1 = 16$

$c_2 = 22$

After:

$c_1 = 15.33$

$c_2 = 36.25$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	
15	16	22	1	7	1	**15.33**
16	16	22	0	6	1	
19	16	22	9	3	2	
19	16	22	9	3	2	
20	16	22	16	2	2	
20	16	22	16	2	2	
21	16	22	25	1	2	
22	16	22	36	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	**36.25**
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

K-Means Clustering - Example

- **Iteration II**

Before:

$c_1 = 15.33$

$c_2 = 36.25$

After:

$c_1 = 18.56$

$c_2 = 45.9$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	18.56
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	45.9
43	15.33	36.25	27.67	6.75	2	
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

K-Means Clustering - Example

- **Iteration III**

Before:

$c_1 = 18.56$

$c_2 = 45.9$

After:

$c_1 = 19.50$

$c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	19.50
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	47.89
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

K-Means Clustering - Example

- **Iteration IV**

Before:

$c_1 = 19.50$

$c_2 = 47.89$

After:

$c_1 = 19.50$

$c_2 = 47.89$

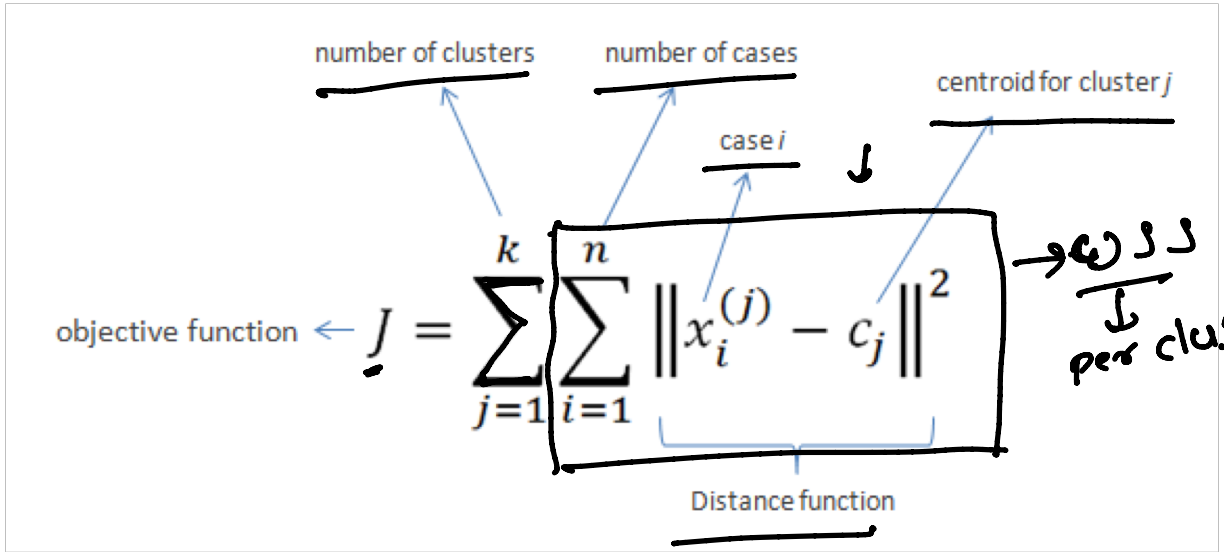
x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	19.50
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	47.89
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

K-Means Clustering

- How to find the optimum number of clusters?
  - Elbow Method
  - Purpose Method

Elbow Method

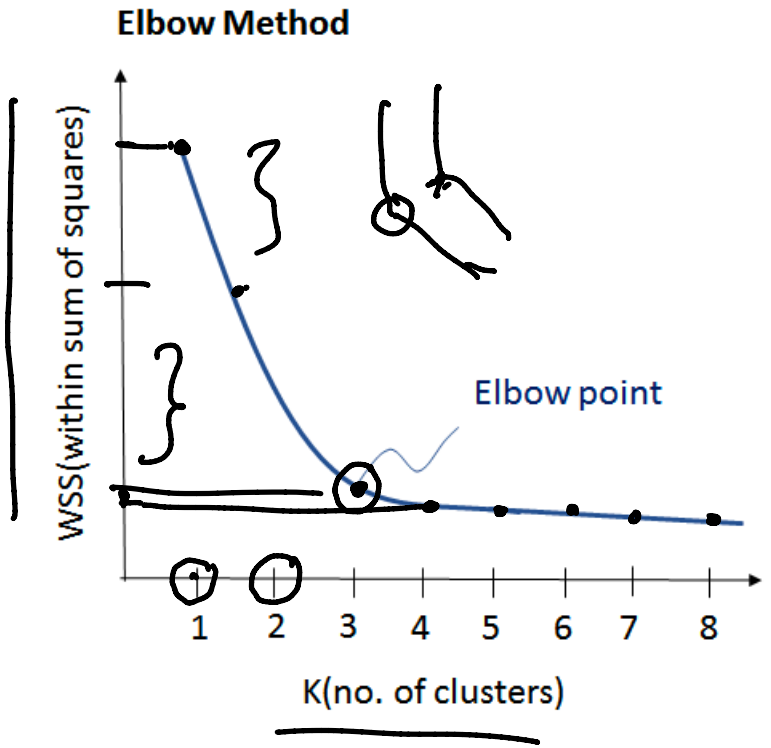
- Total within-cluster variation
  - Also known as Within Sum of Squares (WSS)
  - The sum of squared distances (Euclidean) between the items and the corresponding centroid



Elbow Method

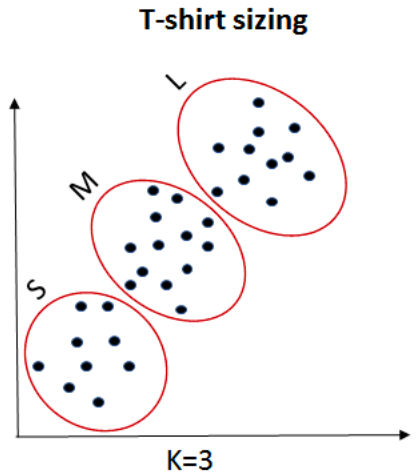
- Draw a curve between WSS (within sum of squares) and the number of clusters

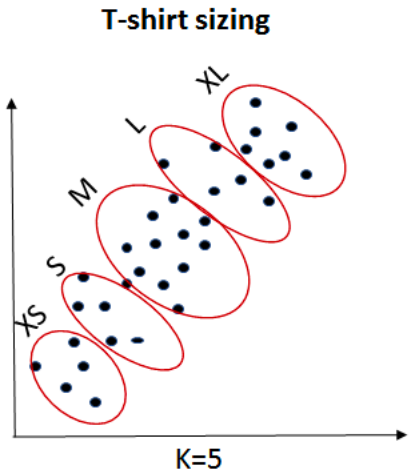
- It is called elbow method because the curve looks like a human arm and the elbow point gives us the optimum number of clusters



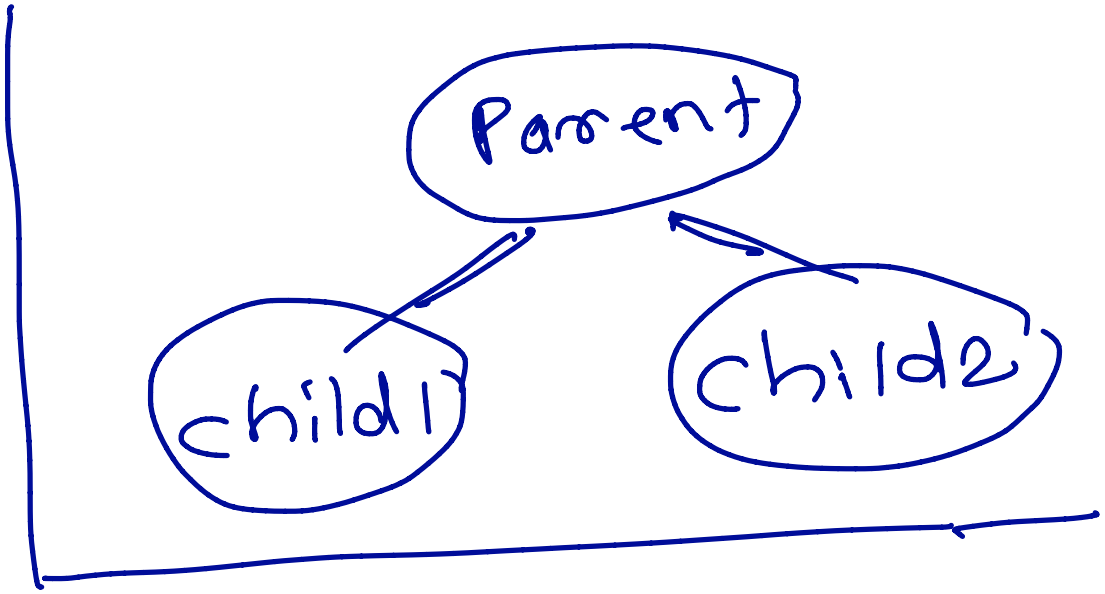
Purpose Method

- Get different clusters based on a variety of purposes
- Partition the data on different metrics and see how well it performs for that particular case





- K=3: If you want to provide only 3 sizes(S, M, L) so that prices are cheaper, you will divide the data set into 3 clusters.
- K=5: Now, if you want to provide more comfort and variety to your customers with more sizes (XS, S, M, L, XL), then you will divide the data set into 5 clusters.



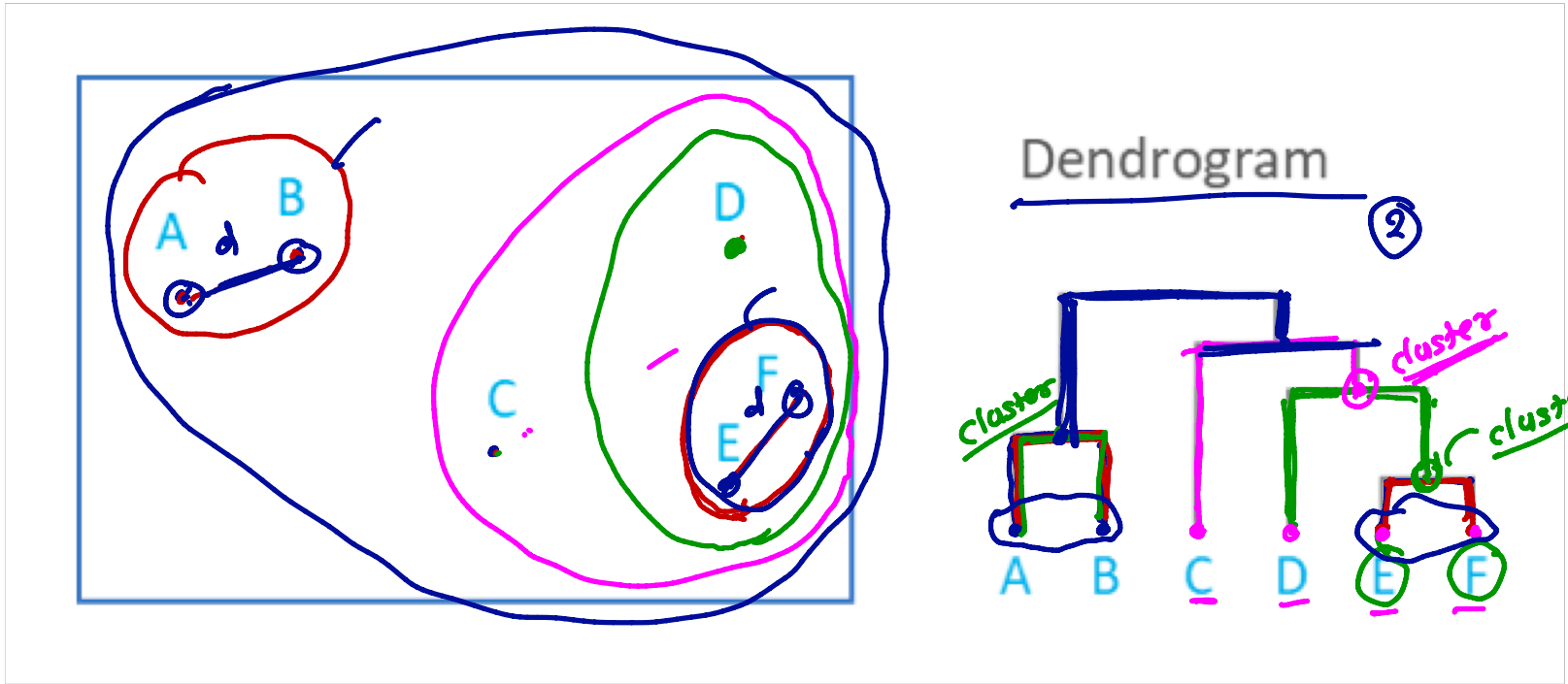
Hierarchical Clustering

Hierarchical Clustering (Hclust)

- Separating data into different groups based on some measure of similarity (distance)
- Types
  - Agglomerative — bottom to top
  - Divisive — top to bottom

Hierarchical Clustering

- Dendrogram
  - diagram that shows the hierarchical relationship between objects



Agglomerative Clustering

- Also called as bottom-top clustering as it uses bottom-up approach
- Each data point starts in its own cluster
- These clusters are then joined greedily by taking two most similar clusters together

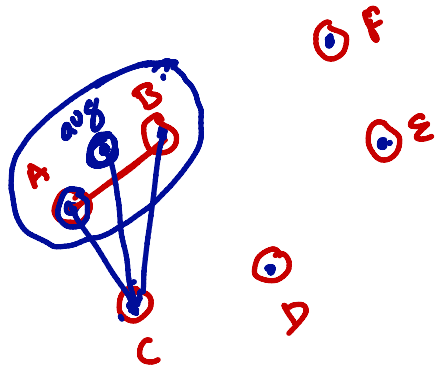
most similar = smaller distance
= closest

Agglomerative Clustering

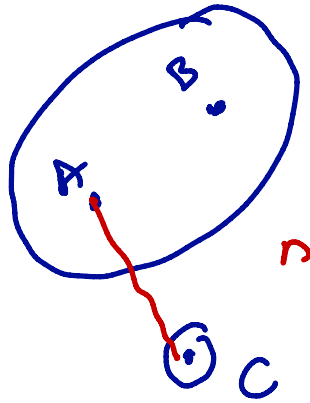
- Start by assigning each item to a cluster
  - if you have N items, you now have N clusters, each containing just one item
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less
- Compute distances (similarities) between the new cluster and each of the old clusters
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N

Agglomerative Clustering

- Step 3 can be done in different ways
  - Single-linkage
  - Complete-linkage
  - Average-linkage



Agglomerative Clustering

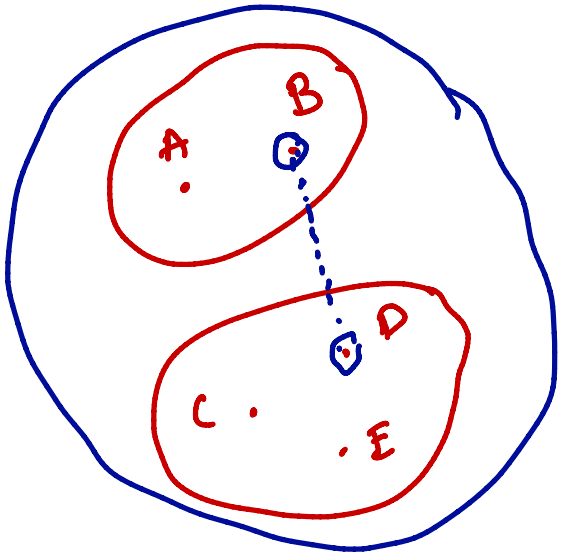


because A is nearest to C.

$d((A,B), C) = d(A, C)$

- **Single linkage**
  - Also known as nearest neighbour clustering
  - The distance between two groups is defined as the distance between their two closest members
  - It often yields clusters in which individuals are added sequentially to a single group

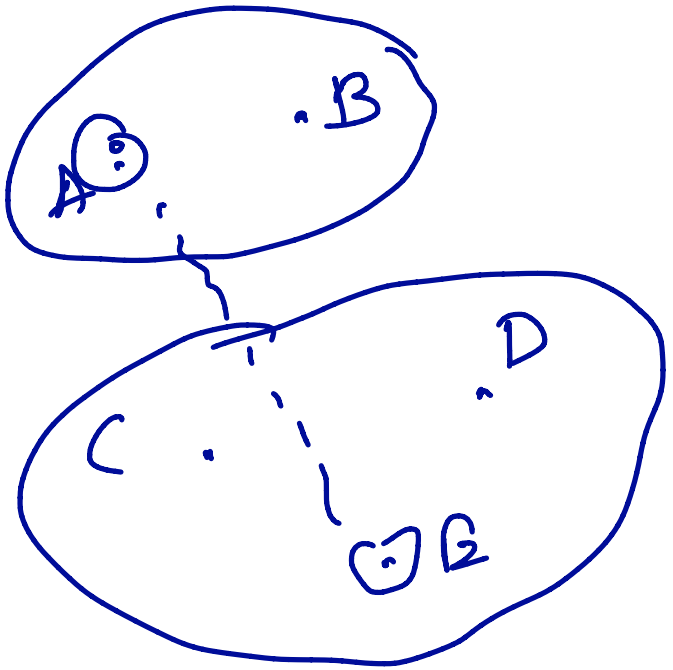
$d((A,B), (C,D,E)) = d(B, D)$



Agglomerative Clustering

- **Complete linkage** farthest
  - also known as ~~furthest~~ neighbour clustering
  - the distance between two groups as the distance between their two farthest-apart members

$$d((A,B),(C,D,E)) = d(A,E)$$

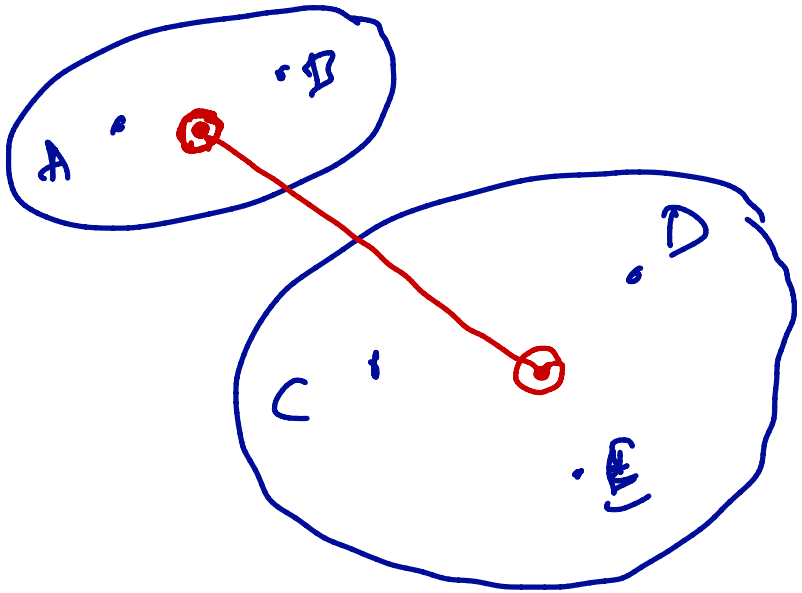


Agglomerative Clustering

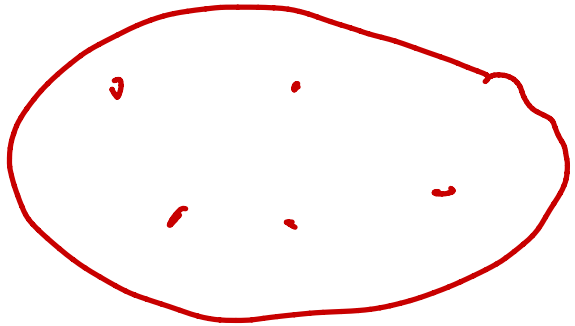
- **Average linkage**
  - referred to as the unweighted pair-group method
  - distance between two groups is defined as the average distance between each of their members

$$d((A,B),(C,D,E)) =$$

$$d(\text{mean}(A,B), \text{mean}(C,D,E))$$



Divisive Clustering



- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it's the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found

Divisive Clustering

- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it's the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found

Divisive Clustering

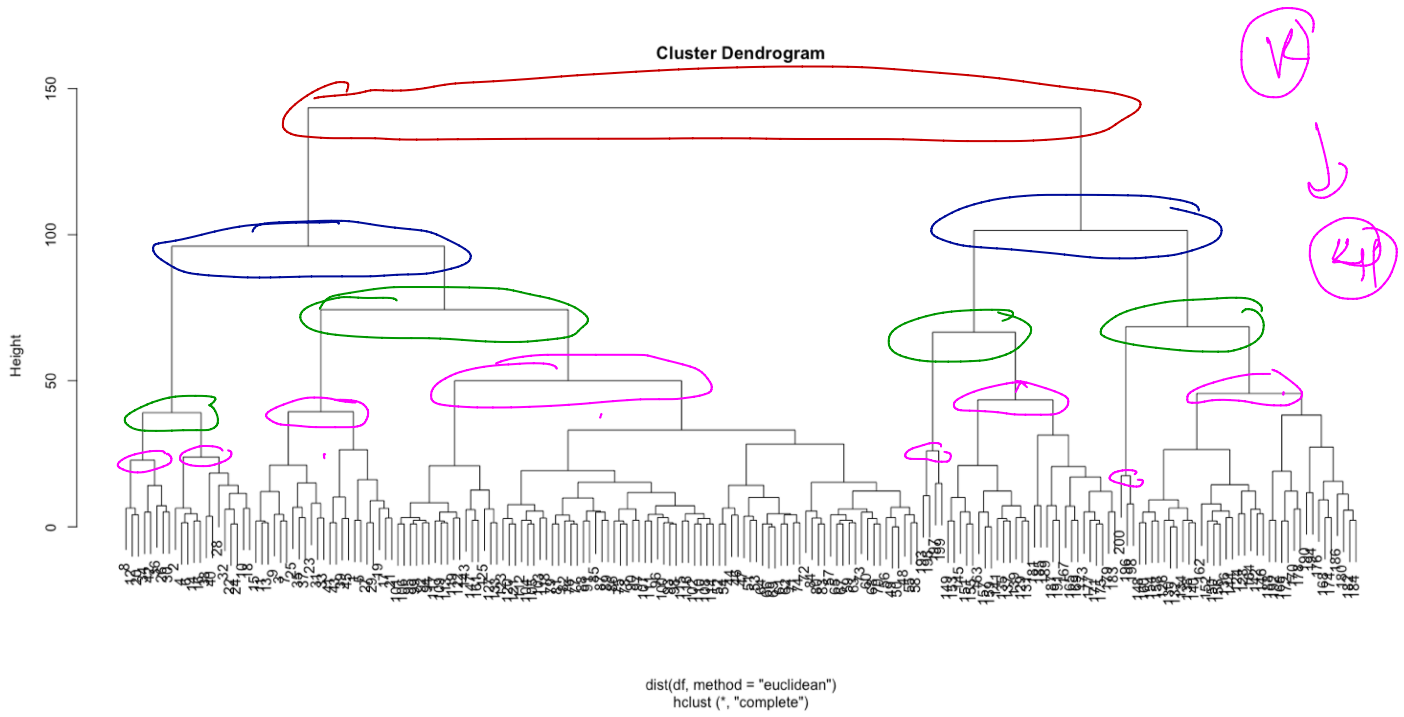
- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it’s the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found

Divisive Clustering

- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it’s the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found

Divisive Clustering

- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it’s the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found


Cluster Dendrogram
Height
0
50
100
150
dist(df, method = "euclidean")
hclust (*, "complete")