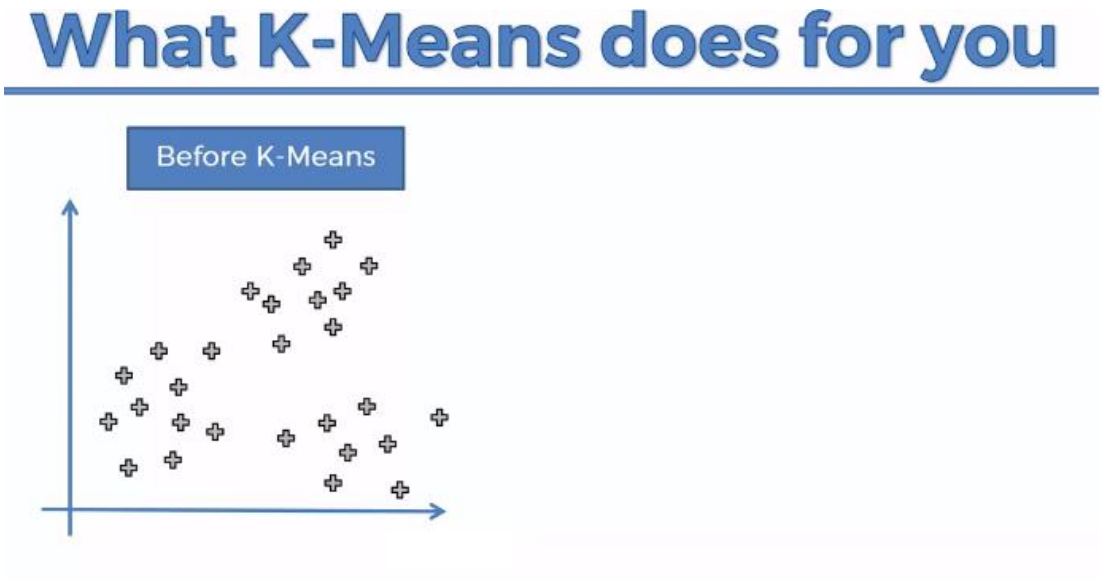
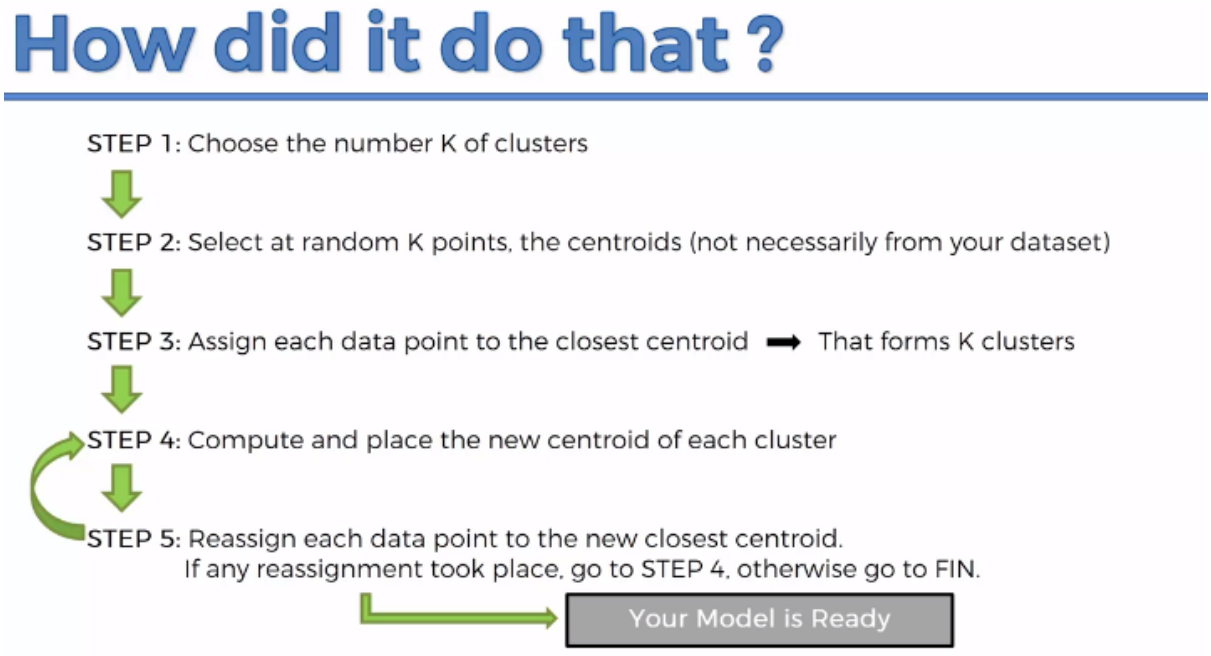
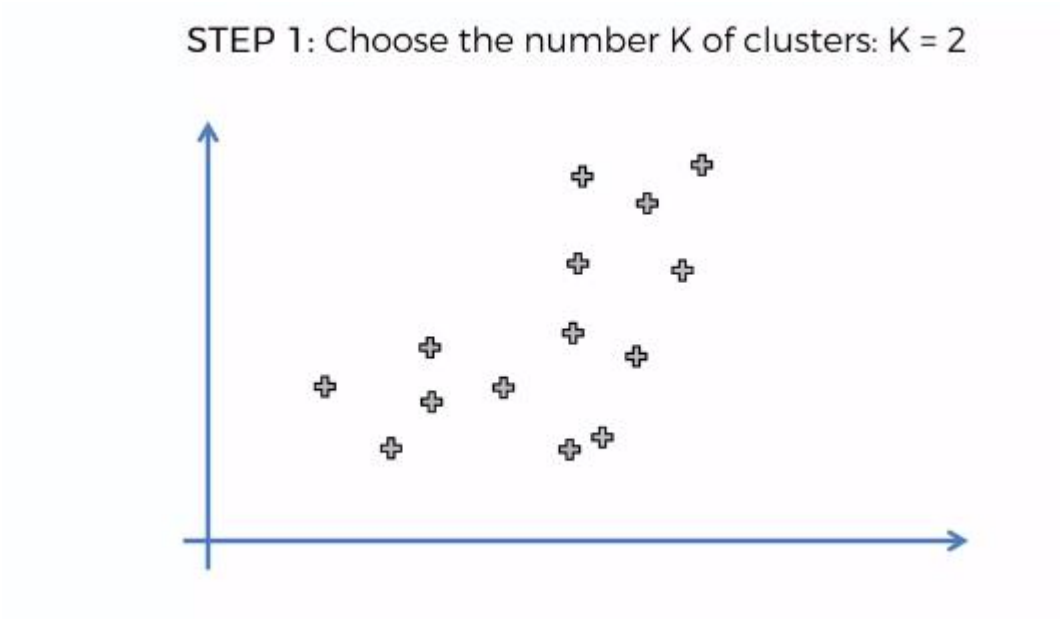
K Means clustering



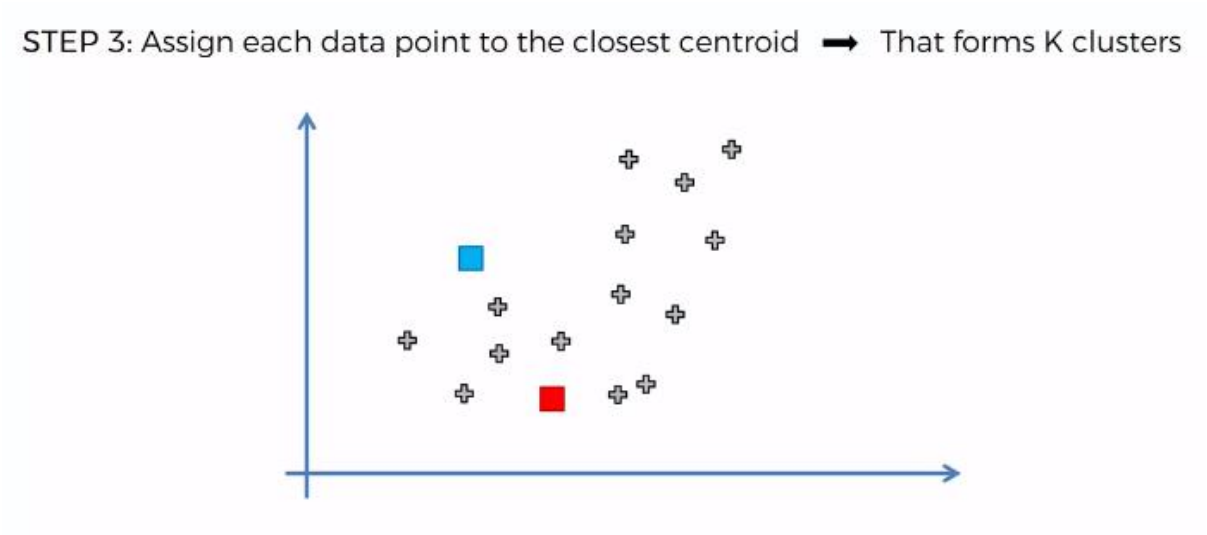
Step by step method



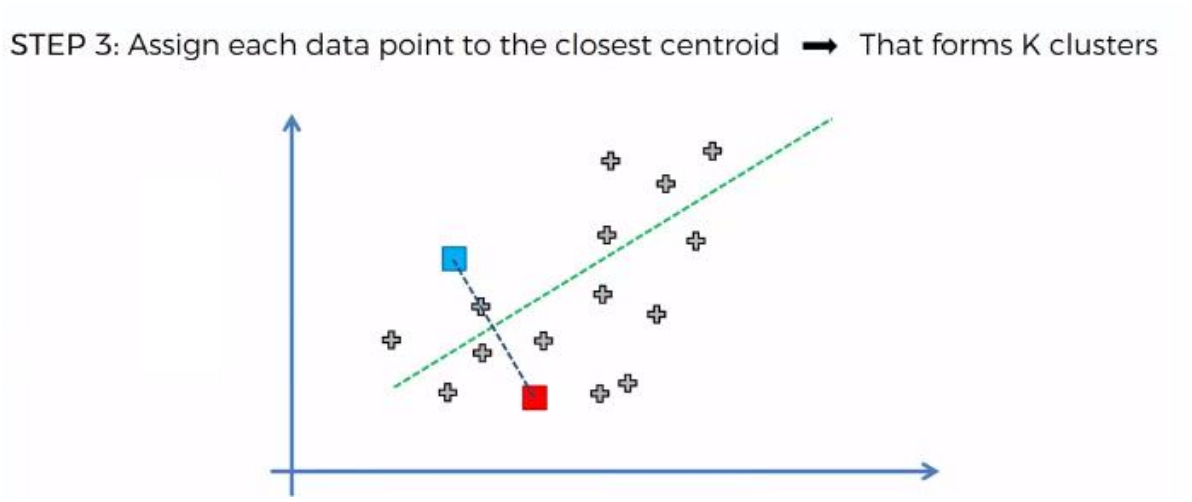


Lets consider we want to design 2 cluster

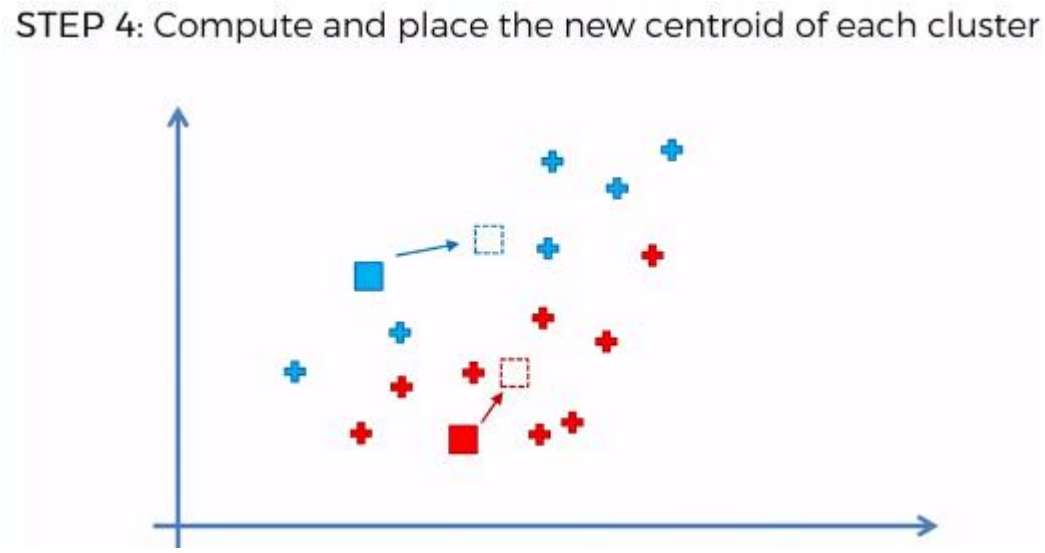
How to find this number we will discuss later



These centroids randomly selected. It may or may not be part of your data

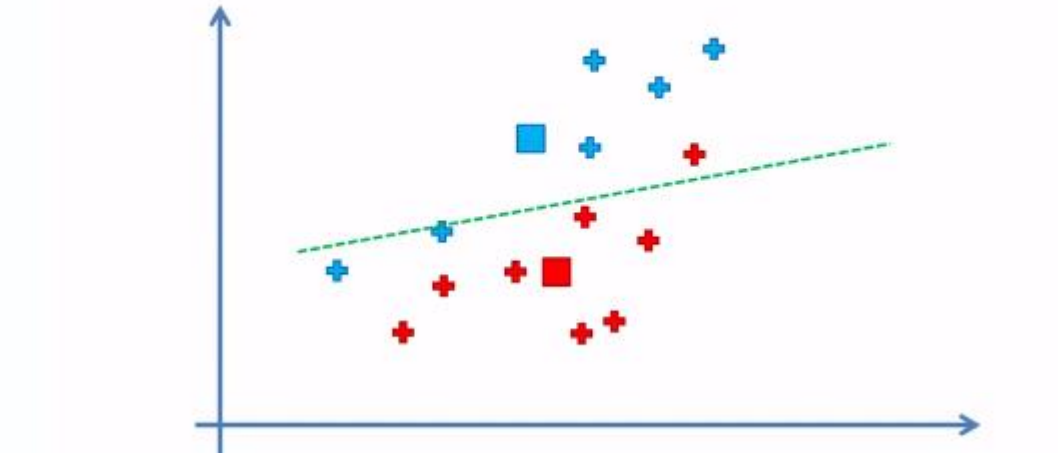


Join centroid and draw a perpendicular horizintal line. All the points on horintal line are at equidistance from centroid. This will help to find which pont is closer to which centroid



Re calculate centroid and again draw pependicular line and see if any changes are happening to cluster

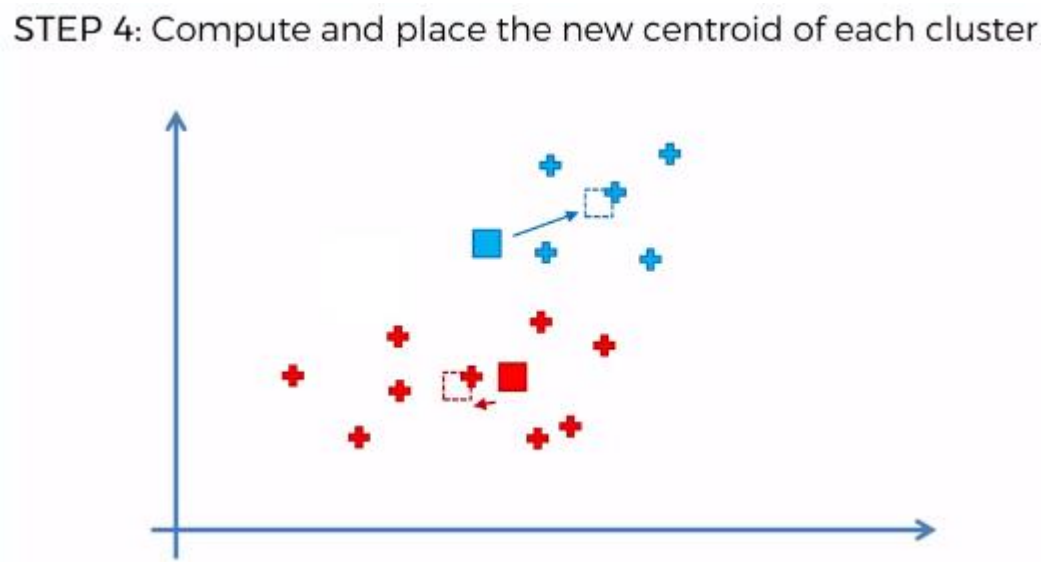
STEP 5: Reassign each data point to the new closest centroi
If any reassignment took place, go to STEP 4, otherwise go to F

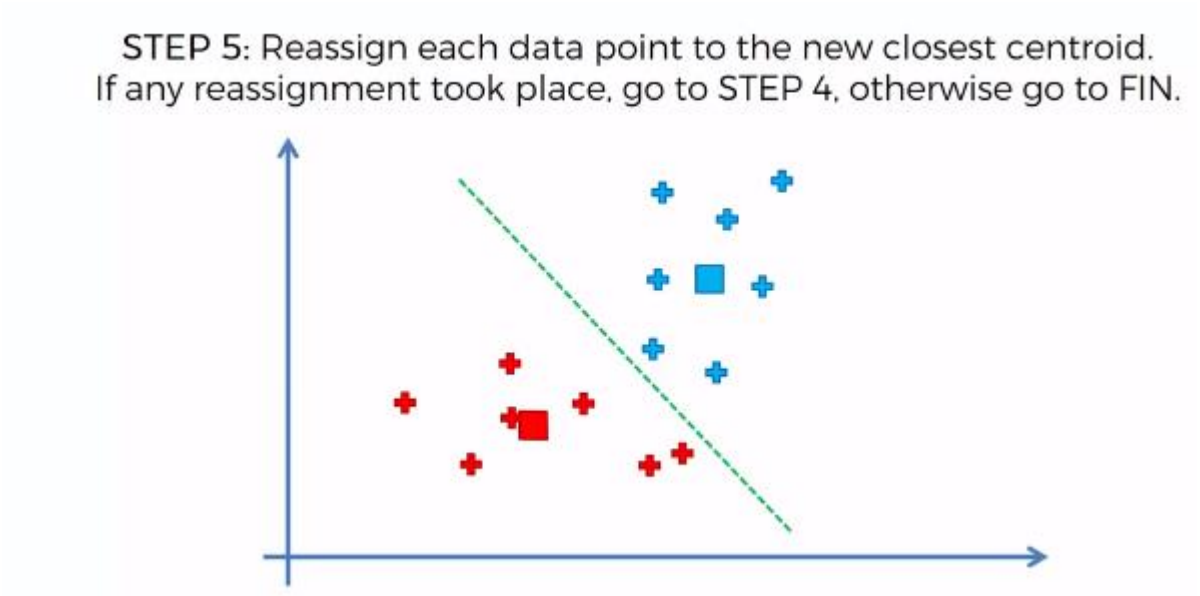


So changes in 3 points will happen

Continue this process till no change happens

STEP 4: Compute and place the new centroid of each cluster

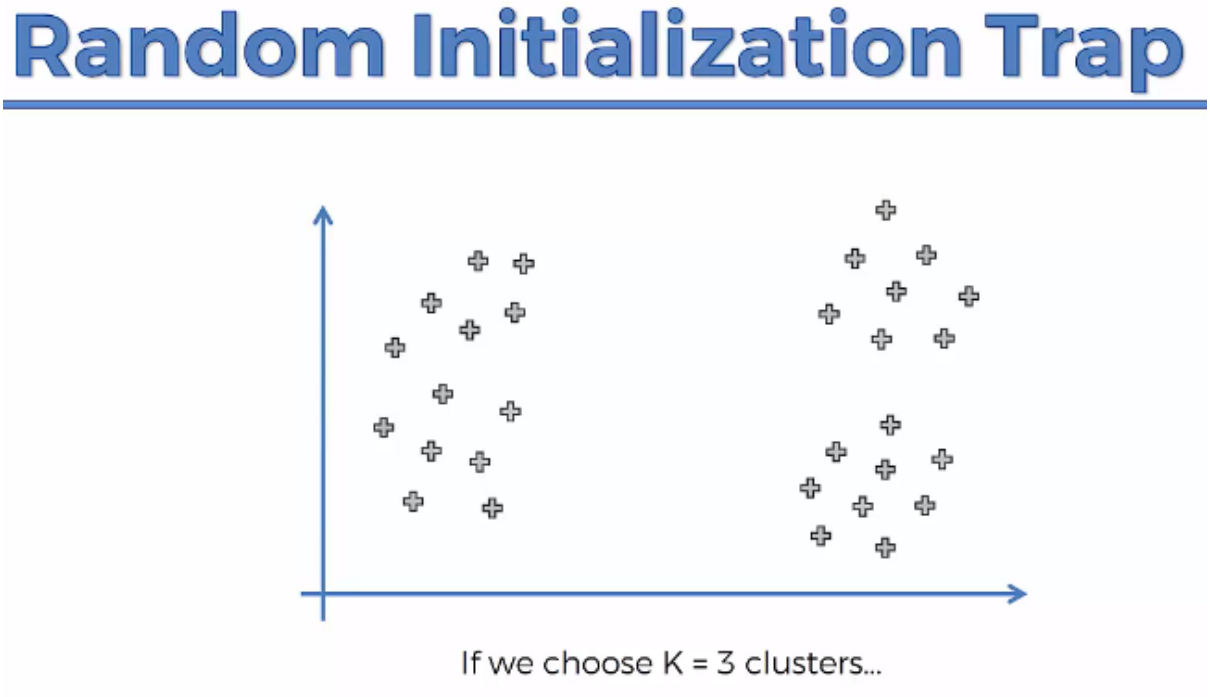




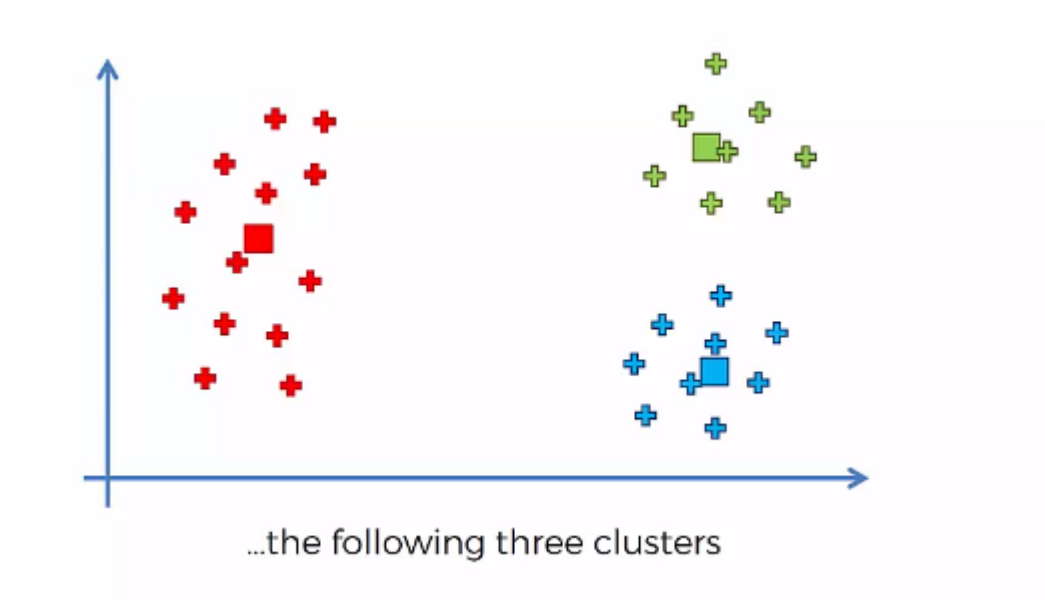
Cluster initialization trap

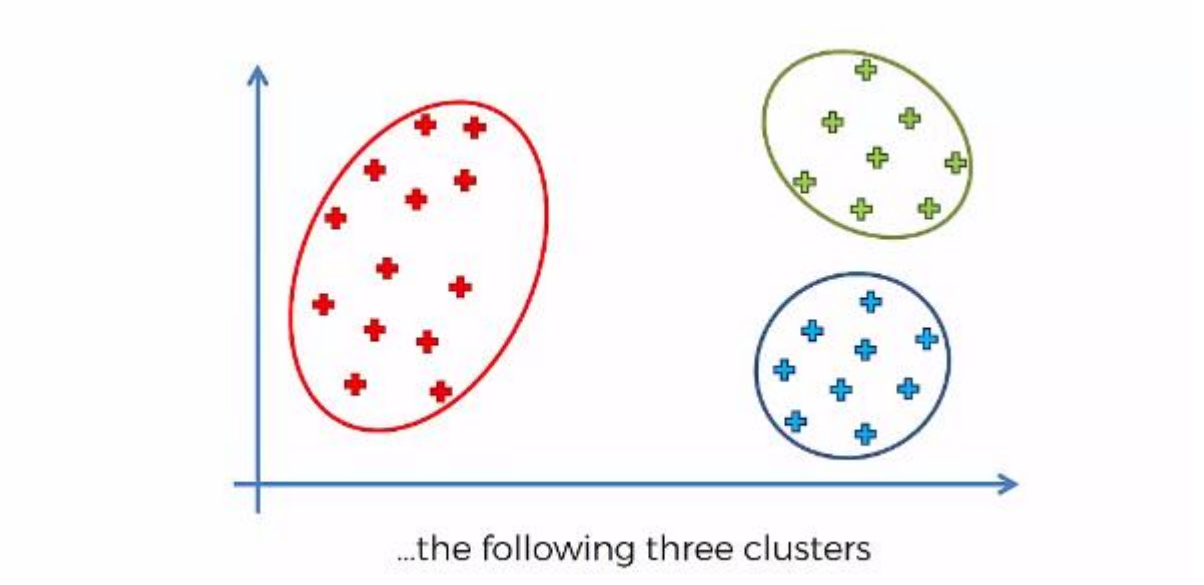
Initially we selecet centroid randomly. What if we started with wrong selection

Example

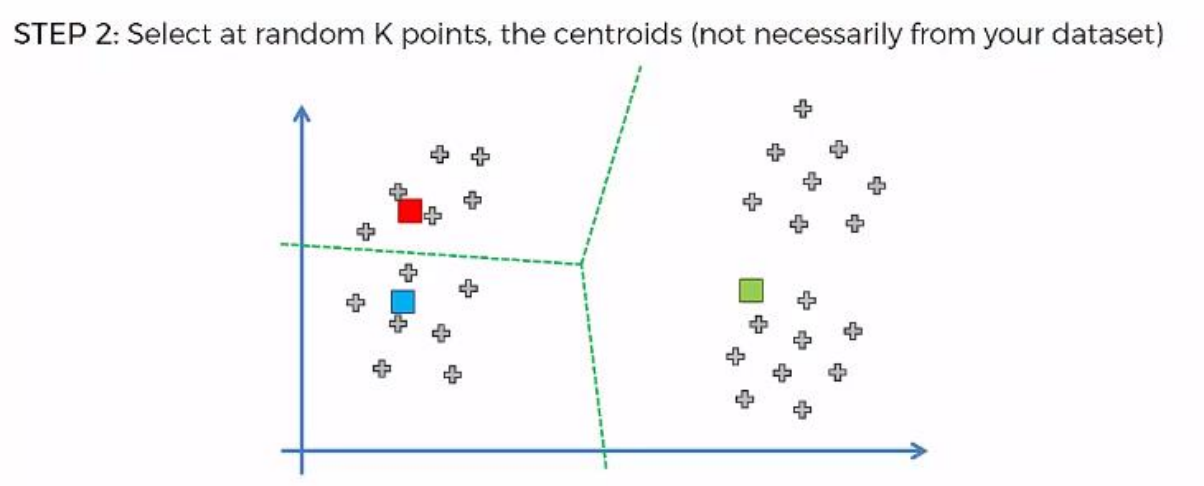


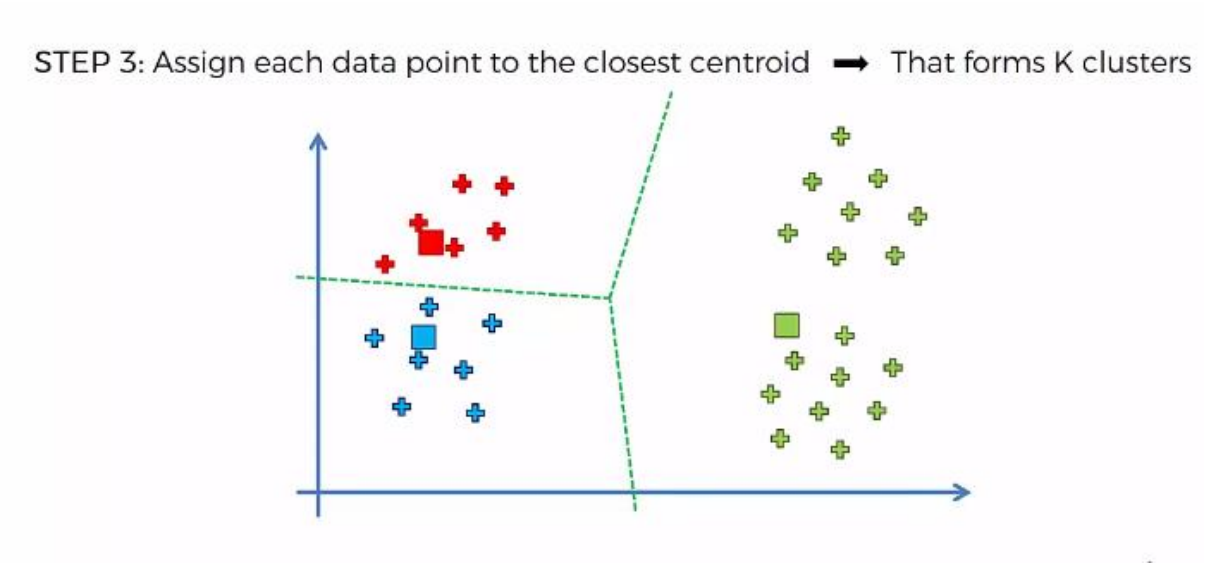
So obvious choice of centroid is as shown

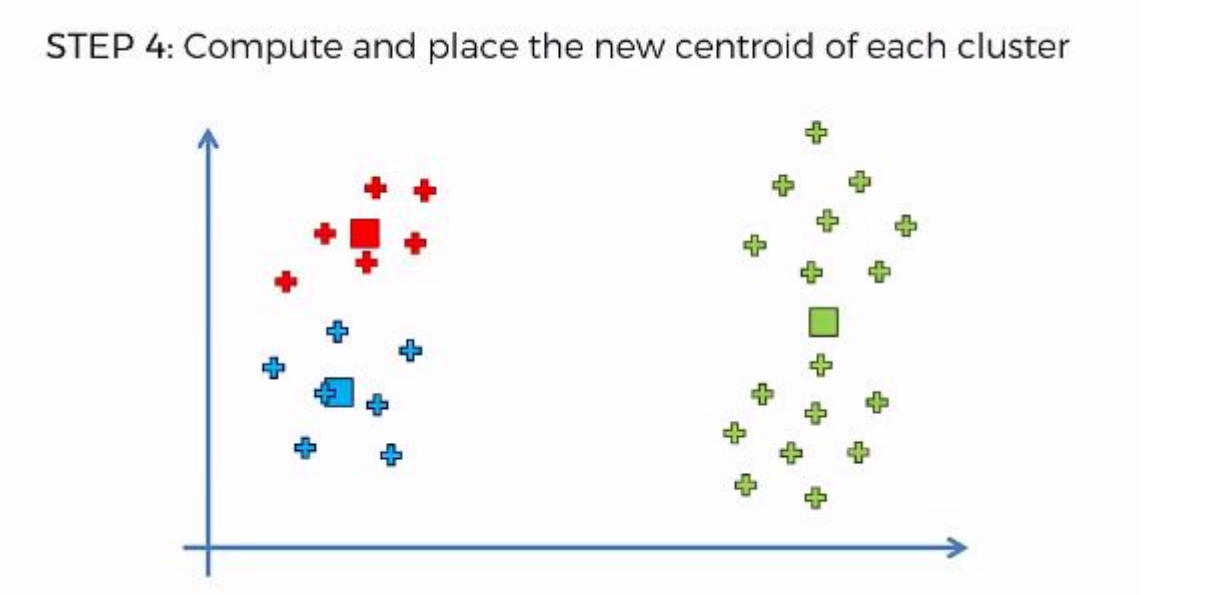




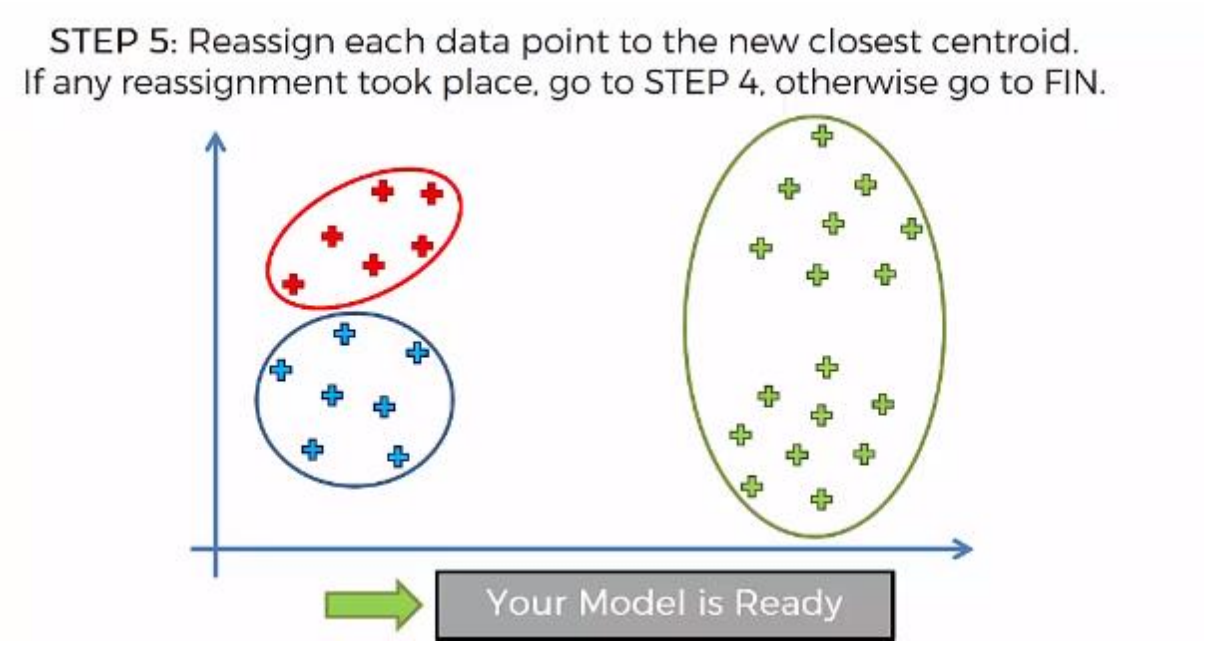
But if we select wrong select wrong centroid





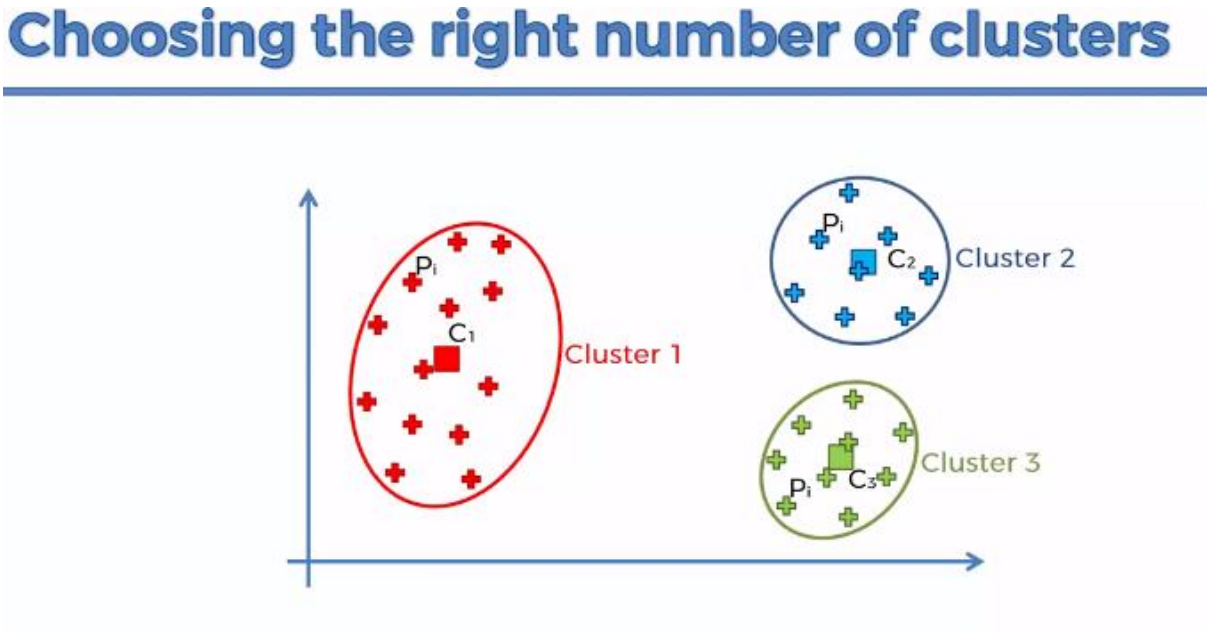


Then clusters will change may be like this



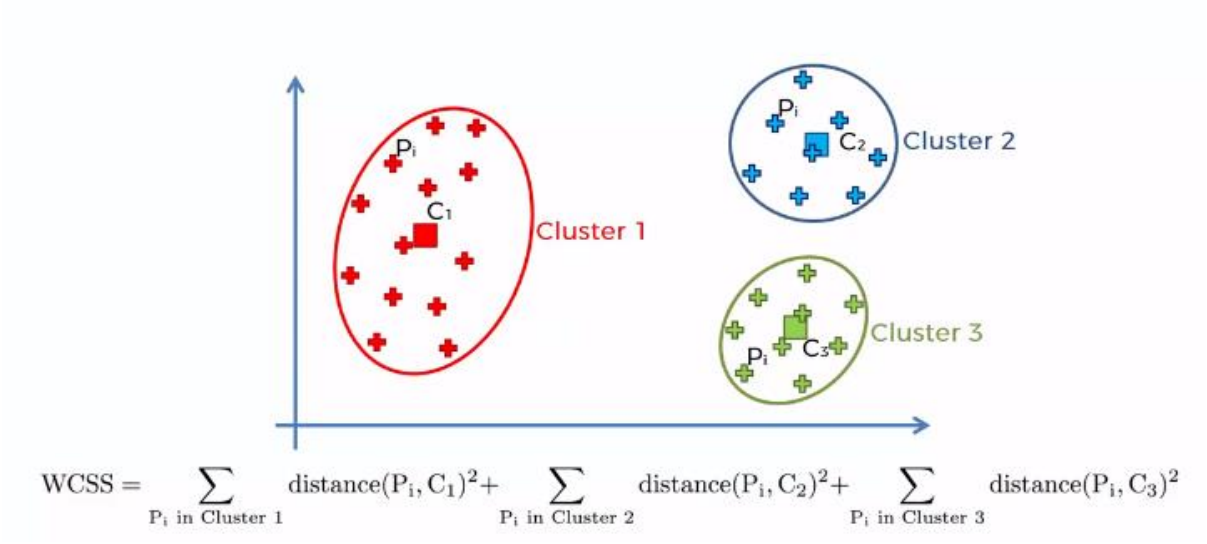
To avoid this internally it takes care. It uses k-means ++ algorithm and it is avoided.

How to find number of clusters for that it uses elbow model



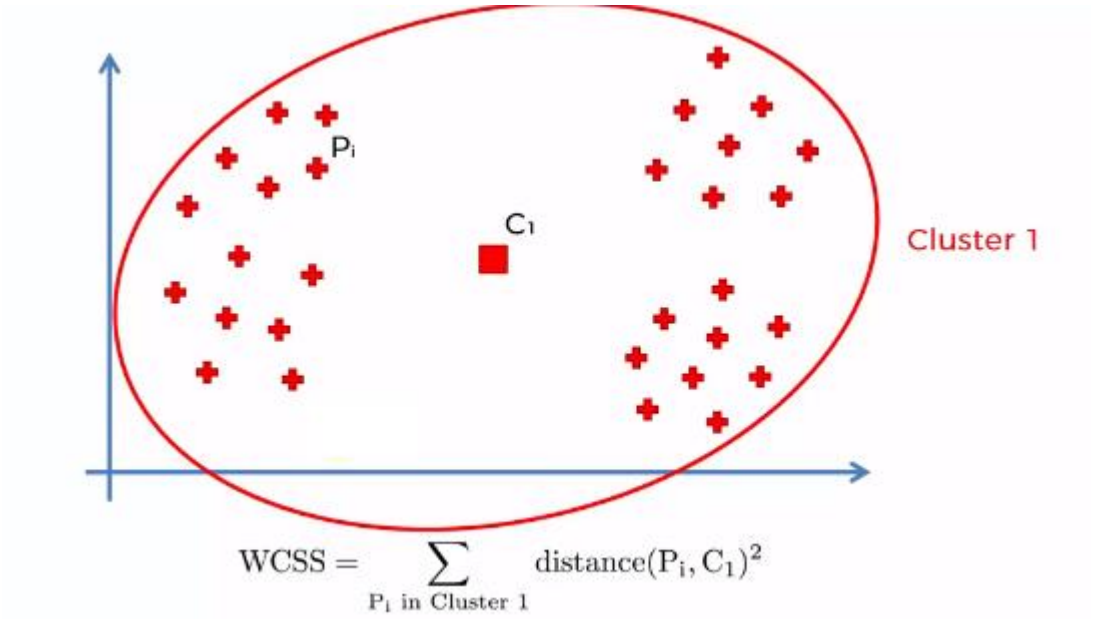
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

It is sum of distance of each point from centroid for each cluster and square of it



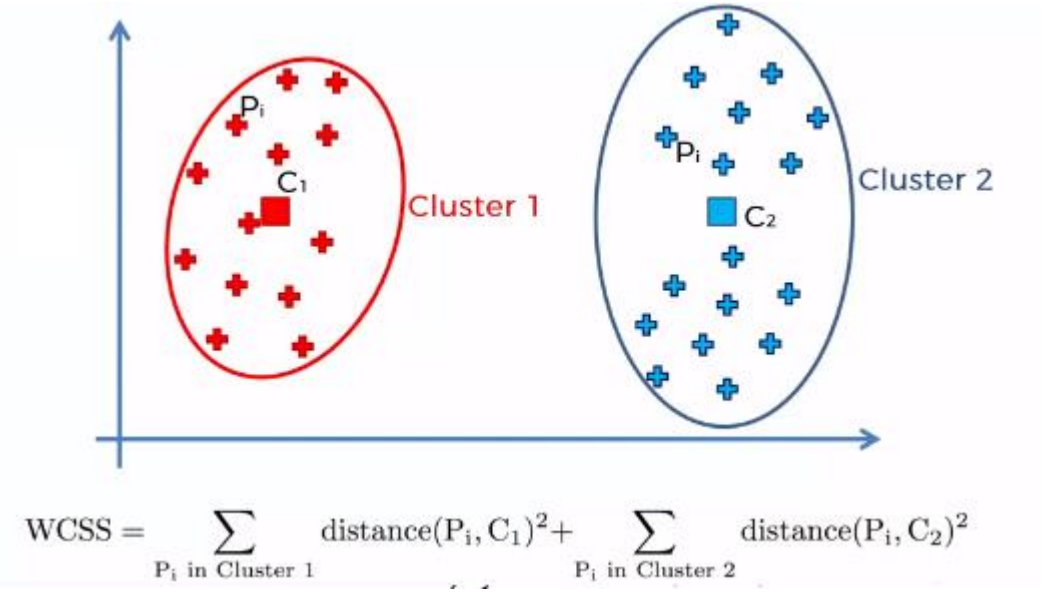
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Example : if one cluster is there



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$$

If 2 clusters are there



How many maximum clusters we can have?

There is no upper limit

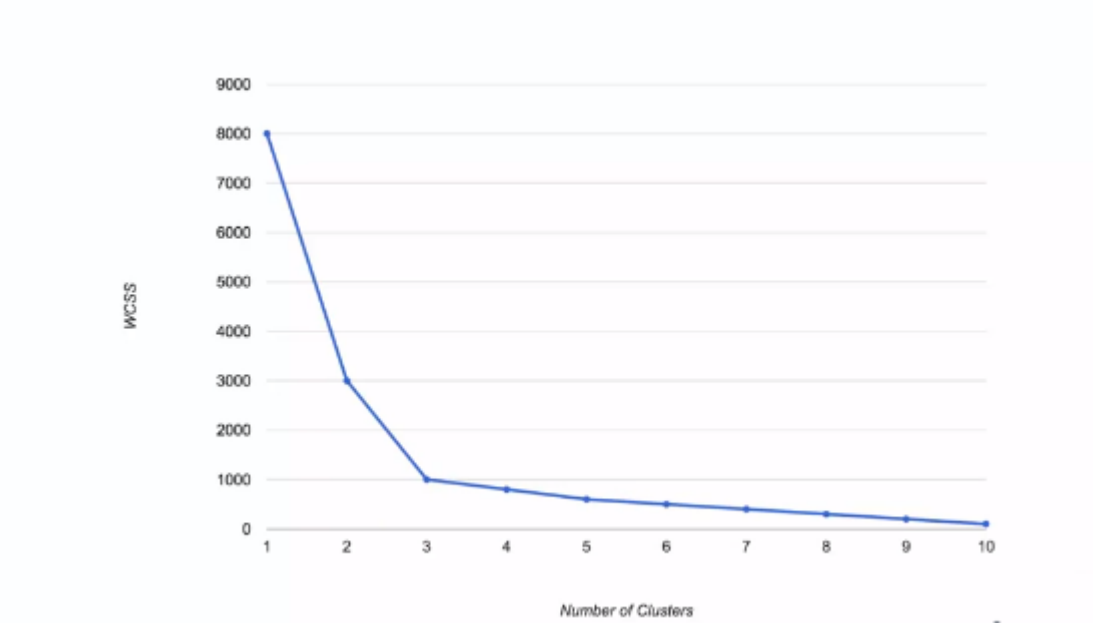
You can have maximum number of cluster = no of poits in the chart

If 50 points are there then you may have 50 clusters

If number of cluster increses WCSS value decreases.

So it will become better better. But then how many clusters?

Where to compromise WCSS value



Till number of clusters are 3 because till there substantial decrease in wcss value is there.

So we will consider it as 3.

This is called as elbow method as graph looks like elbow

Question : from mall.csv classify customers based on salary and spending score

In for loop we will try from cluster number 1 to 11

```
#%reset -f

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the mall dataset with pandas
dataset = pd.read_csv('mall.csv')
X = dataset.iloc[:, [3, 4]].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10)
```

N_cluster: number of clusters

Init :initialization method. Can be random,kmeans++ we will use kmeans++

Max-iter =300 maximum iterations when k_mean algo is running with different initial centroid till final clusters are formed.

N_iter maximum number of iterations when centroid will change and rearrangement of daa happen

To draw elbow graph

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the mall dataset with pandas
dataset = pd.read_csv('mall.csv')
X = dataset.iloc[:, [3, 4]].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Number of clustres are 5.

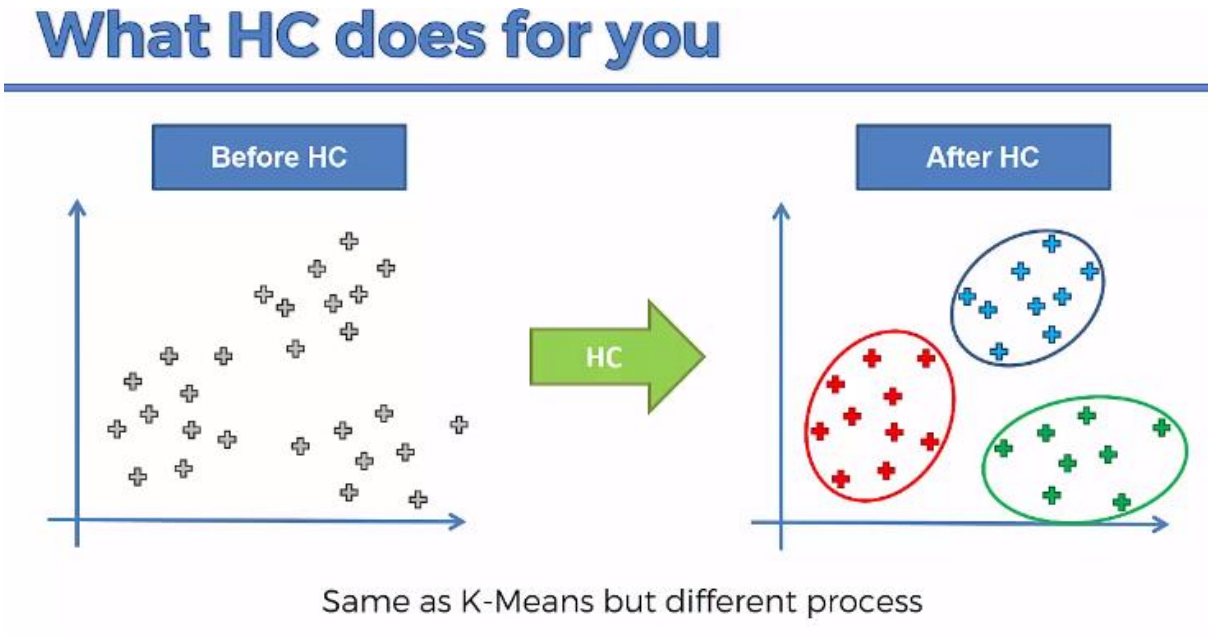
So now use kmean algorithm

```
# Importing the mall dataset with pandas
dataset = pd.read_csv('mall.csv')
X = dataset.iloc[:, [3, 4]].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Applying k-means to the mall dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters of clients')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Every scatter plot represent single cluster

Last scatter plot is for centroid so we hv changed size=300

Hirarchical clustering

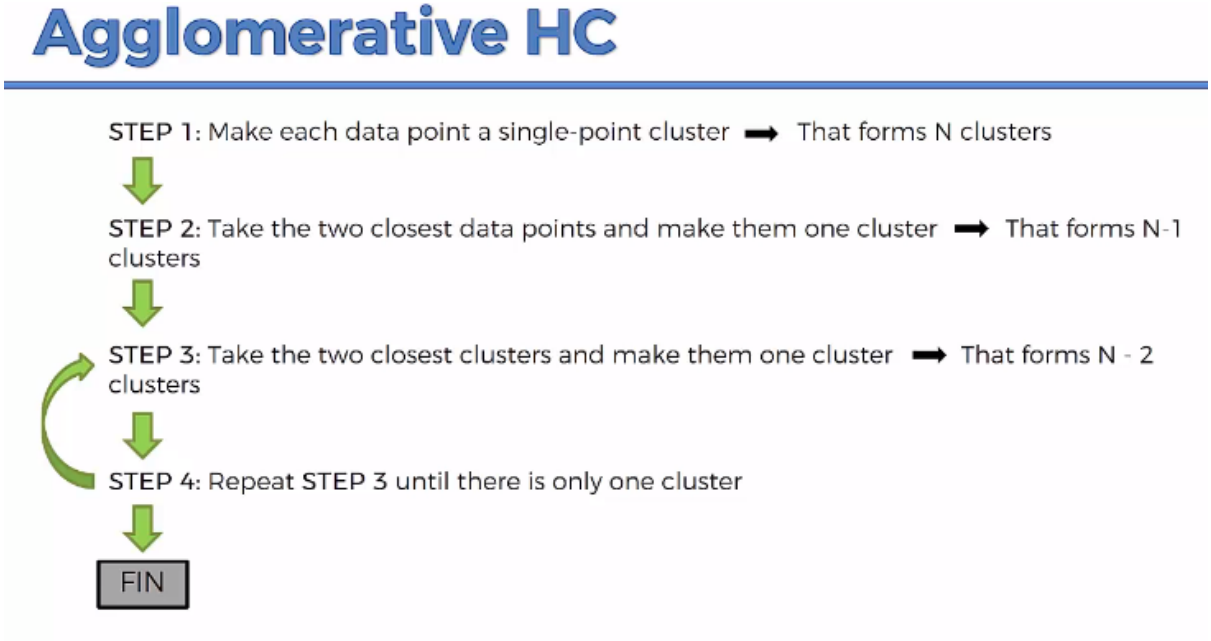


2 type

1. Agglomerative
2. Divisive

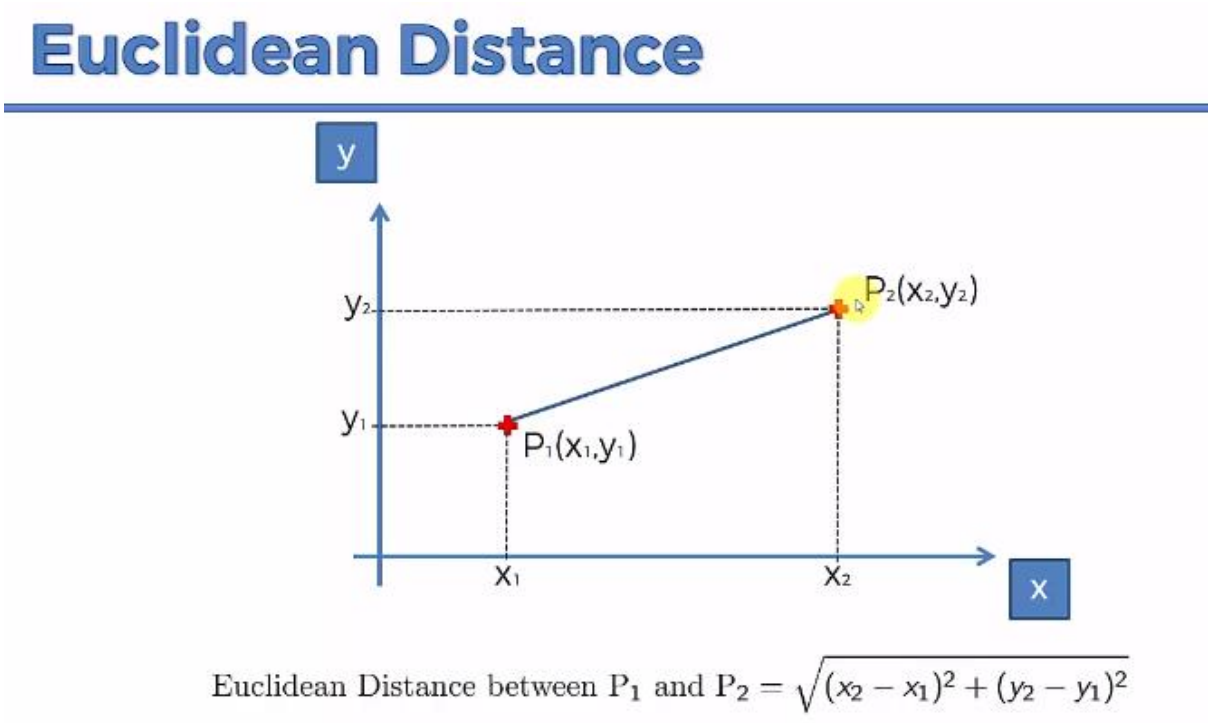
Agglomerative

step by step method



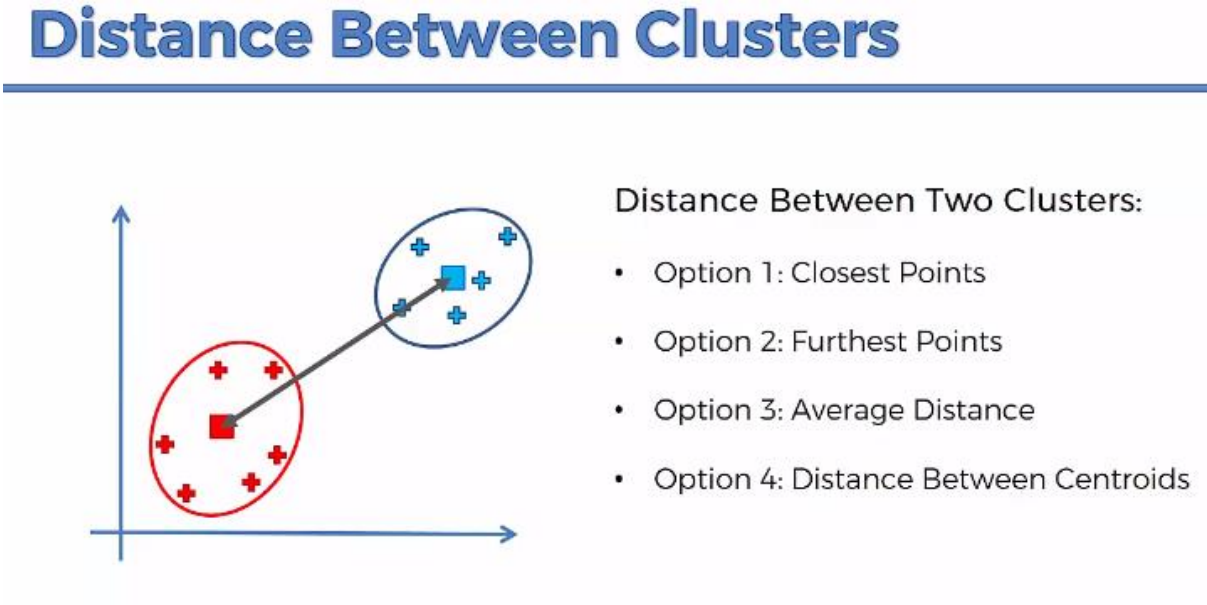
Closest clusters --- How to measure because this may affect cluster formation

We are using Euclidean Distance between P1 and P2

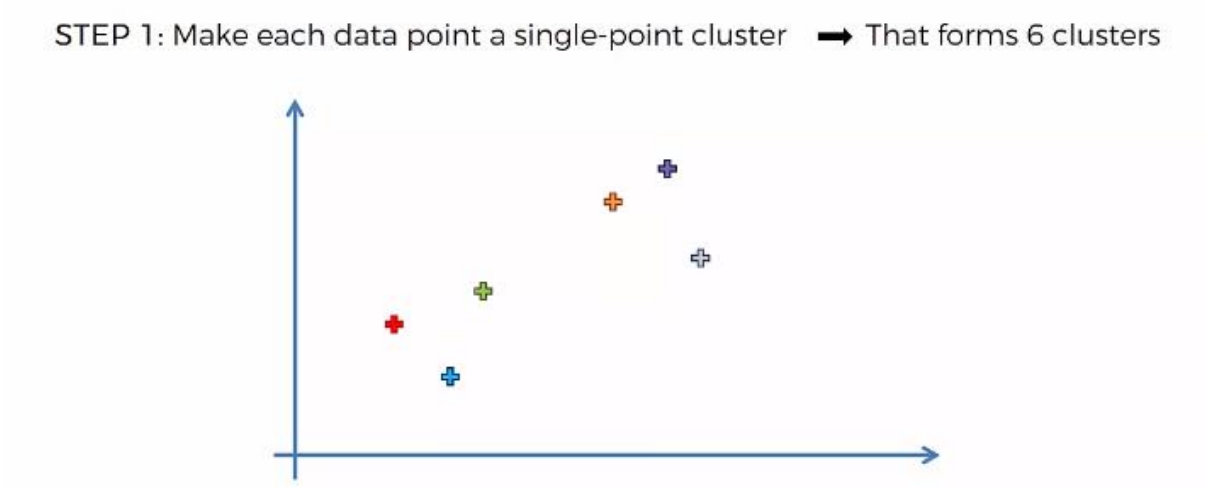


3. How to find distance between 2 cluster
    a. Distance between 2 closest point

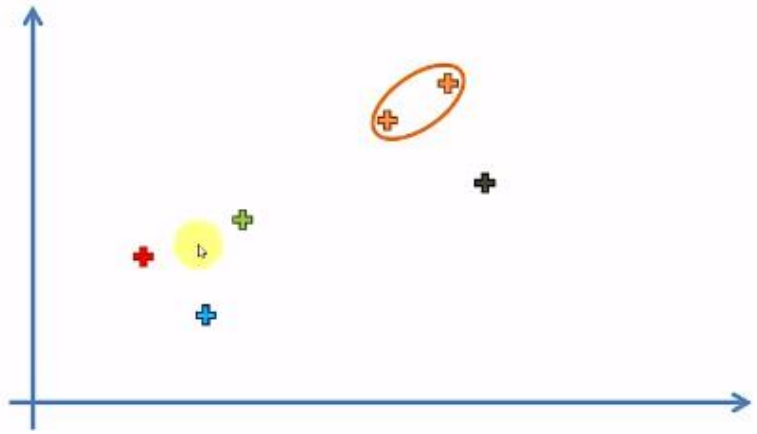
b. Distance between farthest point
c. Average distance between every point



You can select it based on your application

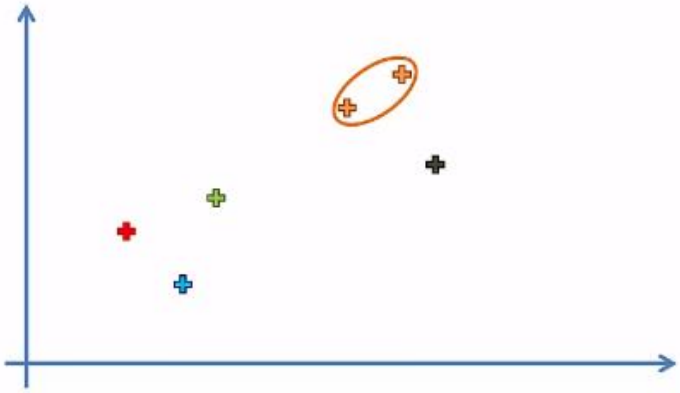


STEP 2: Take the two closest data points and make them one cluster
➞ That forms 5 clusters

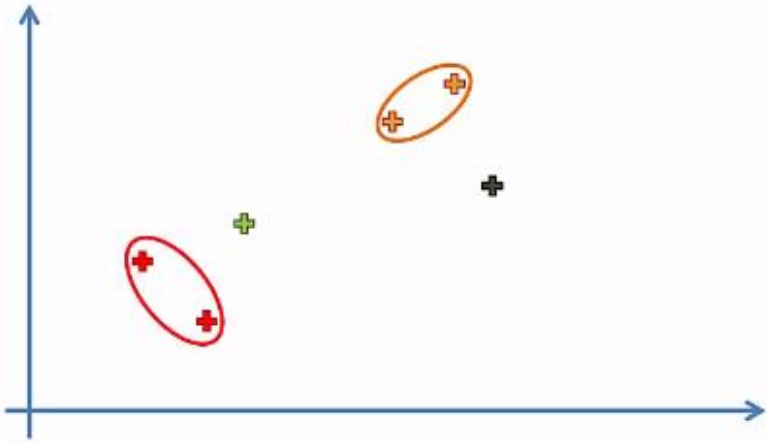


Agglomerative HC

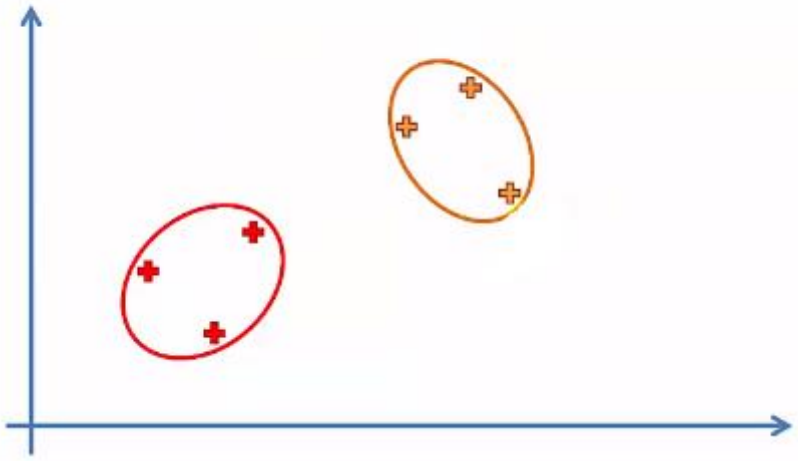
STEP 2: Take the two closest data points and make them one cluster
➞ That forms 5 clusters



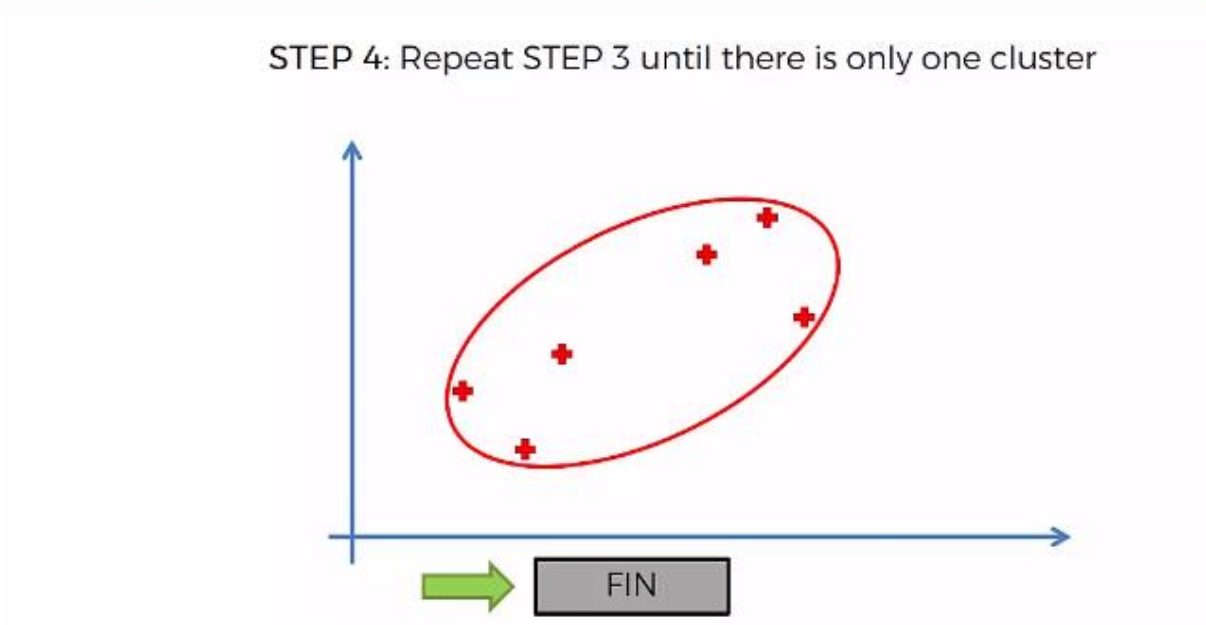
STEP 3: Take the two closest clusters and make them one cluster
➡ That forms 4 clusters



STEP 4: Repeat STEP 3 until there is only one cluster



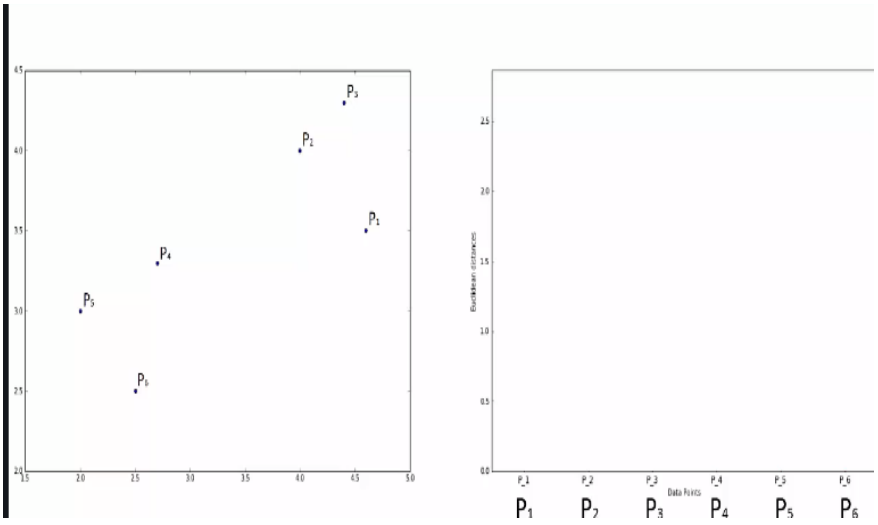
Agglomerative HC



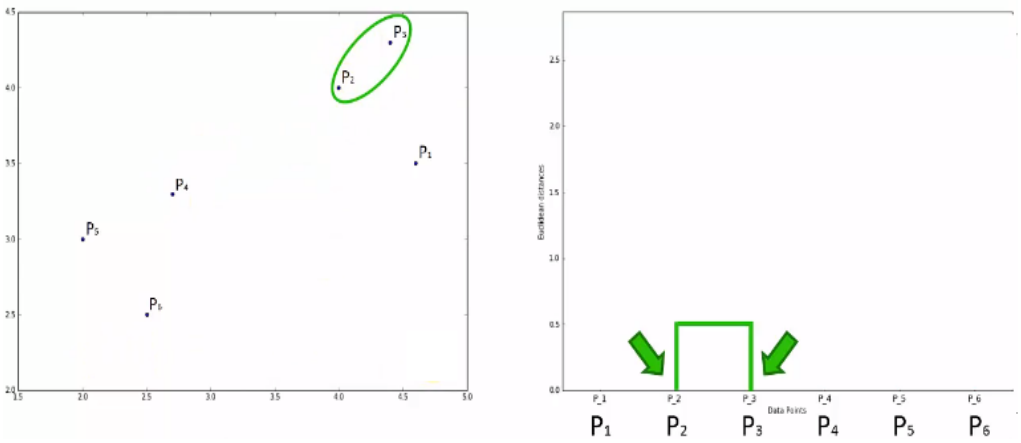
It maintains memory, how this process has happened.
This is stored in a graph called dendogram

Steps for dedogram

Create a graph and plot all points on x axis



Step 3:

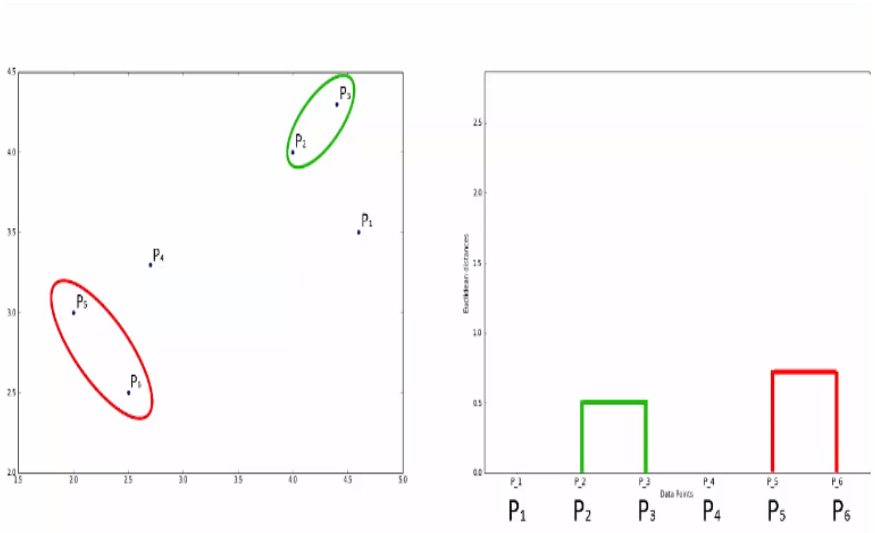


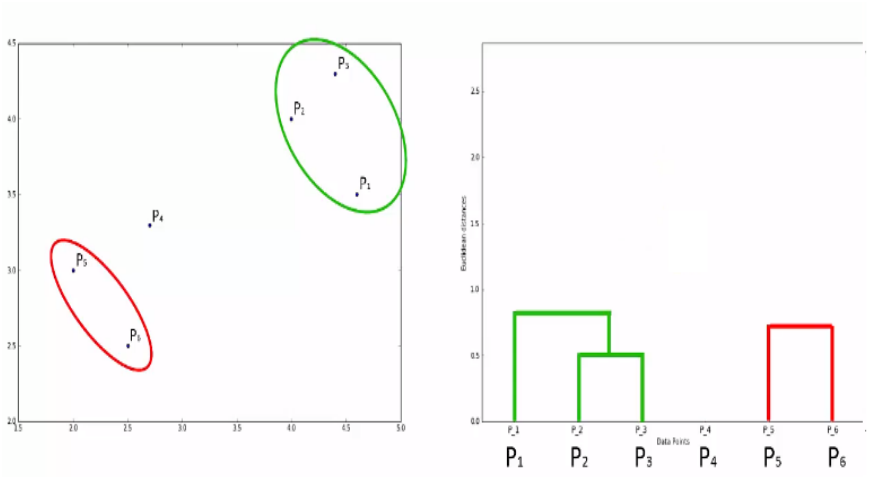
Since p1 and p2 are in same cluster we have drawn horizontal line. Vertical line represents distance between 2 points

Step 3

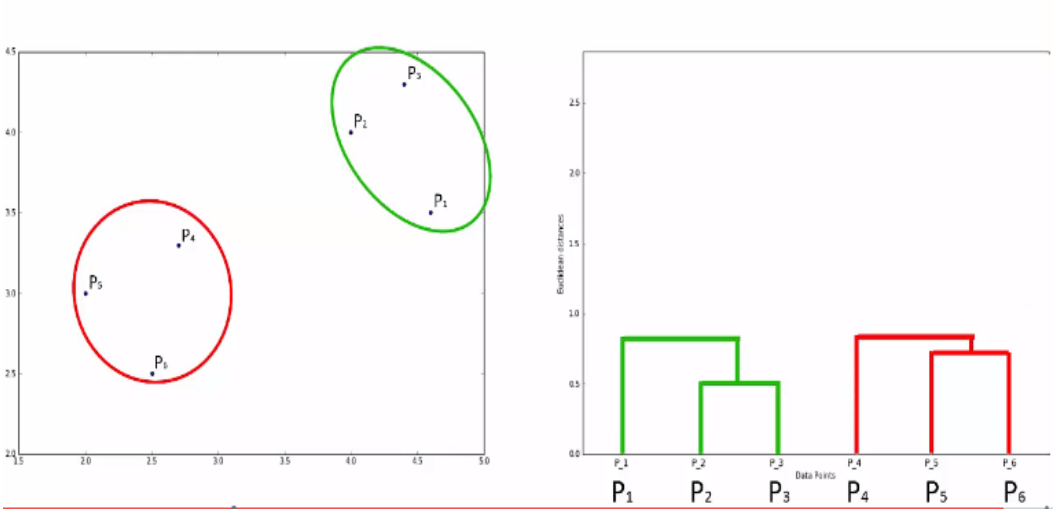
The next cluster is p5 p6 because these are closets points

while drawing horizontal line for p5 p6, draw it little higher than green horizontal line as y axis suggest Euclidian distance and distance is more than p2 and p3

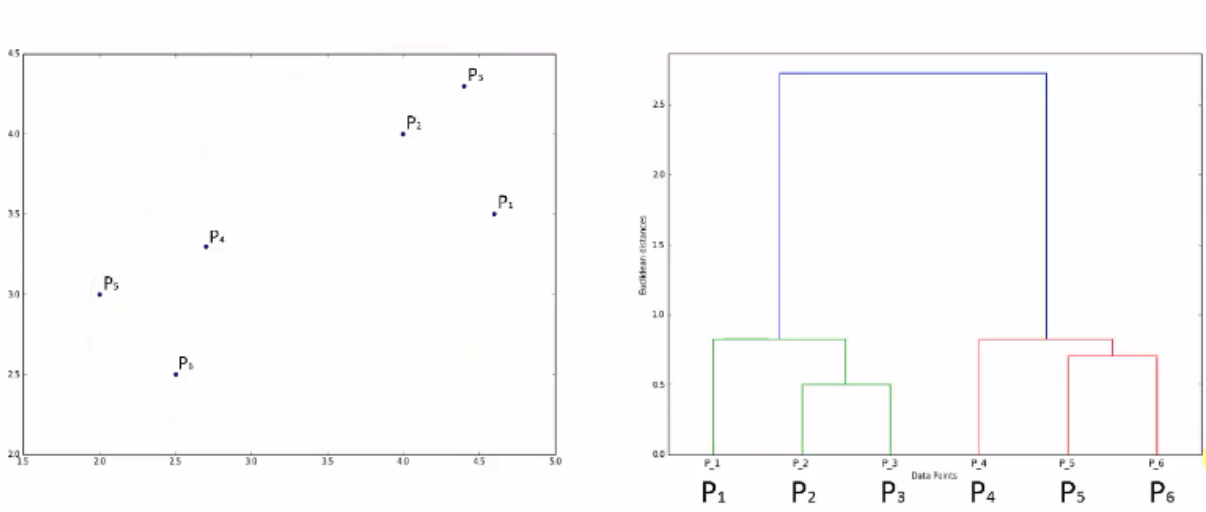




P5-p6 combined with p4p1 and p4 are equidistant from their cluster so nxe green and read are on the same height

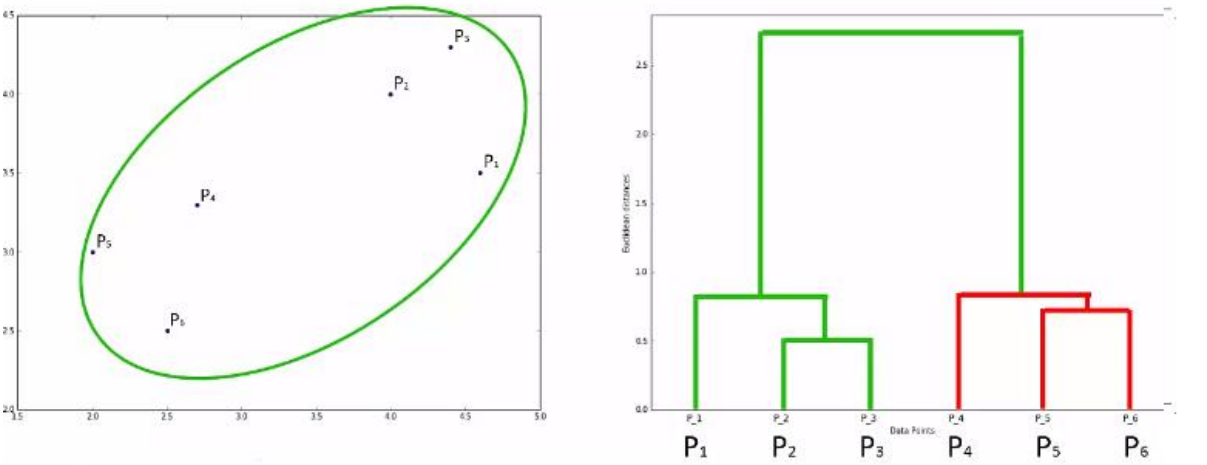


After combining all clusters together



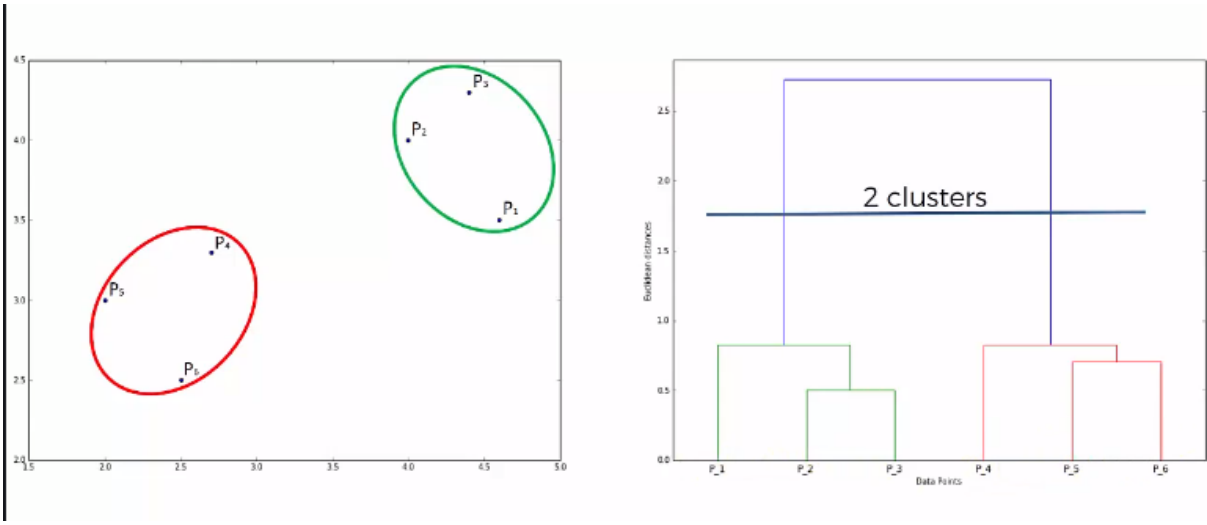
You may say that dendogram represents how the clustering has happened

And it also knows the order in which it has been formed



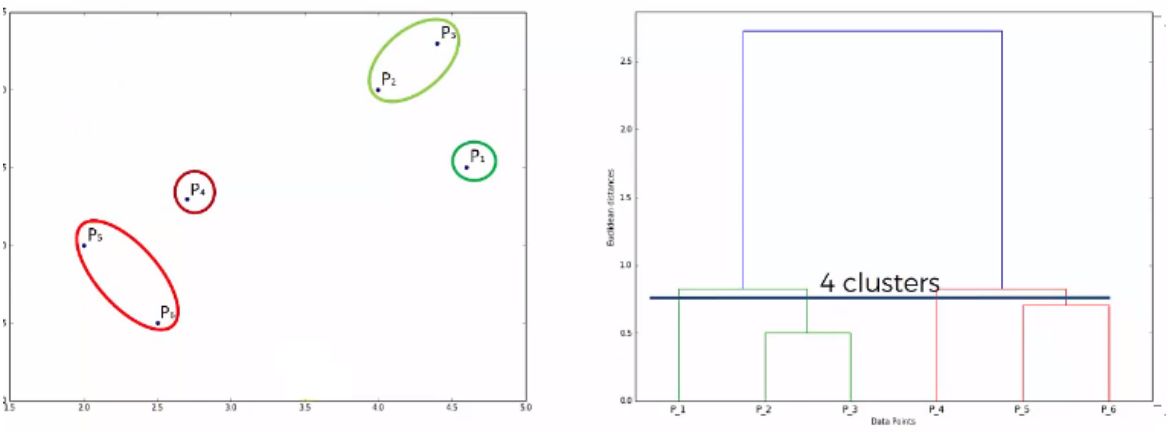
Dendogram gives extra information about distance and also sequence in which the clustering happened.

Now in above diagram if we drae a line at 1.7 this indicates 2 clusters but no 2 points of 2 clusters are away from each other more than by 1.5 distance
If the horizontal line crosses how many vertical lines i.e 2 so number of clusters are 2

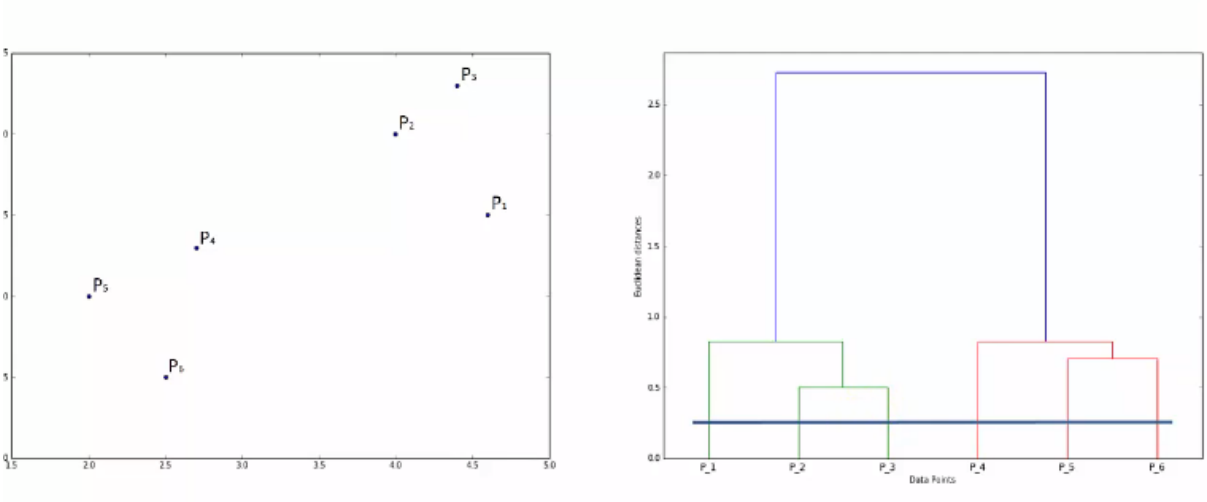


If we keep disimilarity ratio lower than this

i.e



This horizontal line crosses 4 verticle line so below this we will 4 clusters as horizontal line crosses 4 verticle lines.



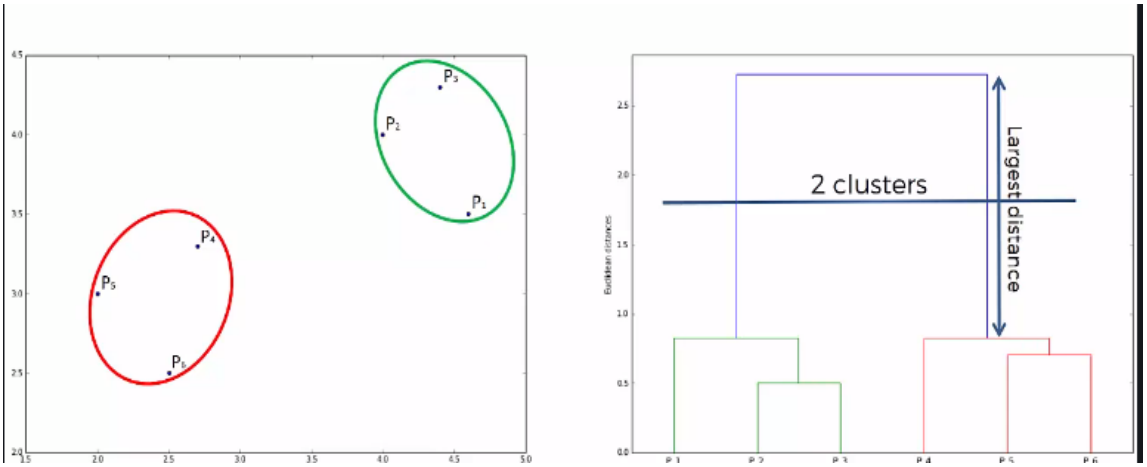
If we put it here then ite will create 6 clusters as horizontal line crosses 6 verticle line.

This will help us to find optimal number of cluster

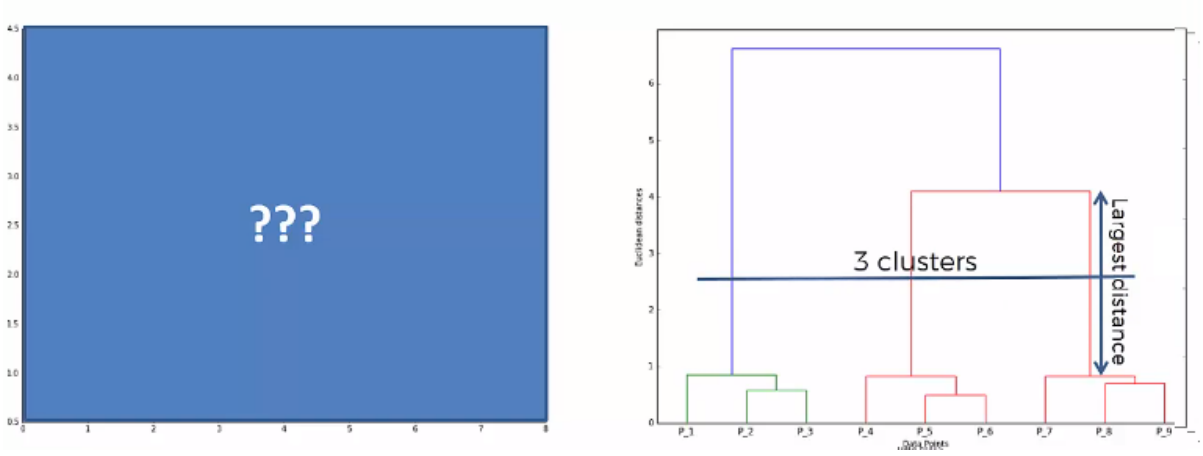
How to select optimum number of clusters

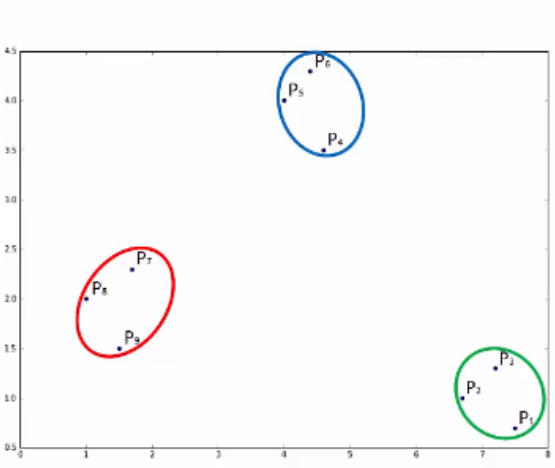
To select optimum number of clusters find longest vertical line that cannot be crossed by any horizontal line if we extend them

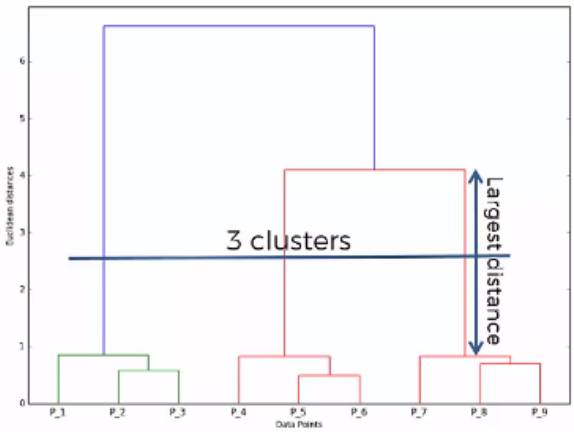
This is not strongest technique but can give us some idea about it



Another example







In python program sch.linkage is algorithm for dendogram we need linkage on x
And method=”ward”
This method helps in designing dendogram by minimizing variance

```
# Using the dendrogram to find the optimal number of clusters
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()

# Fitting hierarchical clustering to the mall dataset
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
```

Affinity – how are we calculating distance between 2 points or cluster
Linkage =ward this helps in reduces the variance
N_cluster is number of clusters which we found by using dendogram
Y_ht=hc.fit_prdict

Now if you open y_ht will show you each line has been added into which cluster

Clustering

Clustering Model	Pros	Cons
K-Means	Simple to understand, easily adaptable, works well on small or large datasets, fast, efficient and performant	Need to choose the number of clusters
Hierarchical Clustering	The optimal number of clusters can be obtained by the model itself, practical visualisation with the dendrogram	Not appropriate for large datasets

Details of hierarchical clustering

NCSS Statistical Software NCSS.com 445-1 © NCSS, LLC. All Rights Reserved. Chapter 445 Hierarchical Clustering / Dendrograms Introduction The agglomerative hierarchical clustering algorithms available in this program module build a cluster hierarchy that is commonly displayed as a tree diagram called a dendrogram. They begin with each object in a separate cluster. At each step, the two clusters that are most similar are joined into a single new cluster. Once fused, objects are never separated. The eight methods that are available represent eight methods of defining the similarity between clusters. Suppose we wish to cluster the bivariate data shown in the following scatter plot. In this case, the clustering may be done visually. The data have three clusters and two singletons, 6 and 13. NCSS Statistical Software NCSS.com Hierarchical Clustering / Dendrograms 445-2 © NCSS, LLC. All Rights Reserved. Following is a dendrogram of the results of running these data through the Group Average clustering algorithm. The horizontal axis of the dendrogram represents the distance or dissimilarity between clusters. The vertical axis represents the objects and clusters. The dendrogram is fairly simple to interpret. Remember that our main interest is in similarity and clustering. Each joining (fusion) of two clusters is represented on the graph by the splitting of a horizontal line into two horizontal lines. The horizontal position of the split, shown by the short vertical bar, gives the distance (dissimilarity) between the two clusters. Looking at this dendrogram, you can see the three clusters as three branches that occur at about the same horizontal distance. The two outliers, 6 and 13, are fused in rather arbitrarily at much higher distances. This is the interpretation. In this example we can compare our interpretation with an actual plot of the data. Unfortunately, this usually will not be possible because our data will consist of more than two variables. Dissimilarities The first task is to form the distances (dissimilarities) between individual objects. This is described in the Medoid Clustering chapter and will not be repeated here. Hierarchical Algorithms The algorithm used by all eight of the clustering methods is outlined as follows. Let the distance between clusters i and j be represented as dij and let cluster i contain ni objects. Let D represent the set of all

remaining dij . Suppose there are N objects to cluster. 1. Find the smallest element dij remaining in D. 2. Merge clusters i and j into a single new cluster, k. 3. Calculate a new set of distances dkm using the following distance formula. d d d d d d km = α i im +α j jm + β ij + im − jm γ 8.00 6.00 4.00 2.00 0.00 Dendrogram Dissimilarity 1 3 2 4 5 6 13 14 15 17 18 20 21 16 19 22 7 8 10 9 11 12 Row NCSS Statistical Software NCSS.com Hierarchical Clustering / Dendrograms 445-3 © NCSS, LLC. All Rights Reserved. Here m represents any cluster other than k. These new distances replace dim and d jm in D. Also let nk = ni + n j . Note that the eight algorithms available represent eight choices for α i ,α j , β, and γ . 4. Repeat steps 1 - 3 until D contains a single group made up off all objects. This will require N-1 iterations. We will now give brief comments about each of the eight techniques. Single Linkage Also known as nearest neighbor clustering, this is one of the oldest and most famous of the hierarchical techniques. 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. The coefficients of the distance equation are α α β γ i = j = 0.5, = 0, = −0.5. Complete Linkage Also known as furthest neighbor or maximum method, this method defines the distance between two groups as the distance between their two farthest-apart members. This method usually yields clusters that are well separated and compact. The coefficients of the distance equation are α α β γ i = j = 0.5, = 0, = 0.5. Simple Average Also called the weighted pair-group method, this algorithm defines the distance between groups as the average distance between each of the members, weighted so that the two groups have an equal influence on the final result. The coefficients of the distance equation are α α β γ i = j = 0.5, = 0, = 0. Centroid Also referred to as the unweighted pair-group centroid method, this method defines the distance between two groups as the distance between their centroids (center of gravity or vector average). The method should only be used with Euclidean distances. The coefficients of the distance equation are α α β α α γ i i k j j k i j n n n n = , = , = − , = 0. Backward links may occur with this method. These are recognizable when the dendrogram no longer exhibits its simple tree-like structure in which each fusion results in a new cluster that is at a higher distance level (moves from right to left). With backward links, fusions can take place that result in clusters at a lower distance level (move from left to right). The dendrogram is difficult to interpret in this case. Median Also called the weighted pair-group centroid method, this defines the distance between two groups as the weighted distance between their centroids, the weight being proportional to the number of individuals in each group. Backward links (see discussion under Centroid) may occur with this method. The method should only be used with Euclidean distances. The coefficients of the distance equation are α α β γ i = j = 0.5, = −0.25, = 0. NCSS Statistical Software NCSS.com Hierarchical Clustering / Dendrograms 445-4 © NCSS, LLC. All Rights Reserved. Group Average Also called the unweighted pair-group method, this is perhaps the most widely used of all the hierarchical cluster techniques. The distance between two groups is defined as the average distance between each of their members. The coefficients of the distance equation are α α β γ i i k j j k n n n n = , = , = 0, = 0. Ward’s Minimum Variance With this method, groups are formed so that the pooled within-group sum of squares is minimized. That is, at each step, the two clusters are fused which result in the least increase in the pooled within-group sum of squares. The coefficients of the distance equation are α α β γ i i m k m j j m k m m k m n n n n n n n n n n n n = + + = + + = − + , , ,

= 0. Flexible Strategy Lance and Williams (1967) suggested that a continuum could be made between single and complete linkage. The program lets you try various settings of these parameters which do not conform to the constraints suggested by Lance and Williams. The coefficients of the distance equation should conform to the following constraints α β α α β α β γ i = 1− − j , j = 1− − i , −1≤ ≤ 1, = 0. One interesting exercise is to vary these values, trying to find the set that maximizes the cophenetic correlation coefficient. Goodness-of-Fit Given the large number of techniques, it is often difficult to decide which is best. One criterion that has become popular is to use the result that has largest cophenetic correlation coefficient. This is the correlation between the original distances and those that result from the cluster configuration. Values above 0.75 are felt to be good. The Group Average method appears to produce high values of this statistic. This may be one reason that it is so popular. A second measure of goodness of fit called delta is described in Mather (1976). These statistics measure degree of distortion rather than degree of resemblance (as with the cophenetic correlation). The two delta coefficients are given by A A j