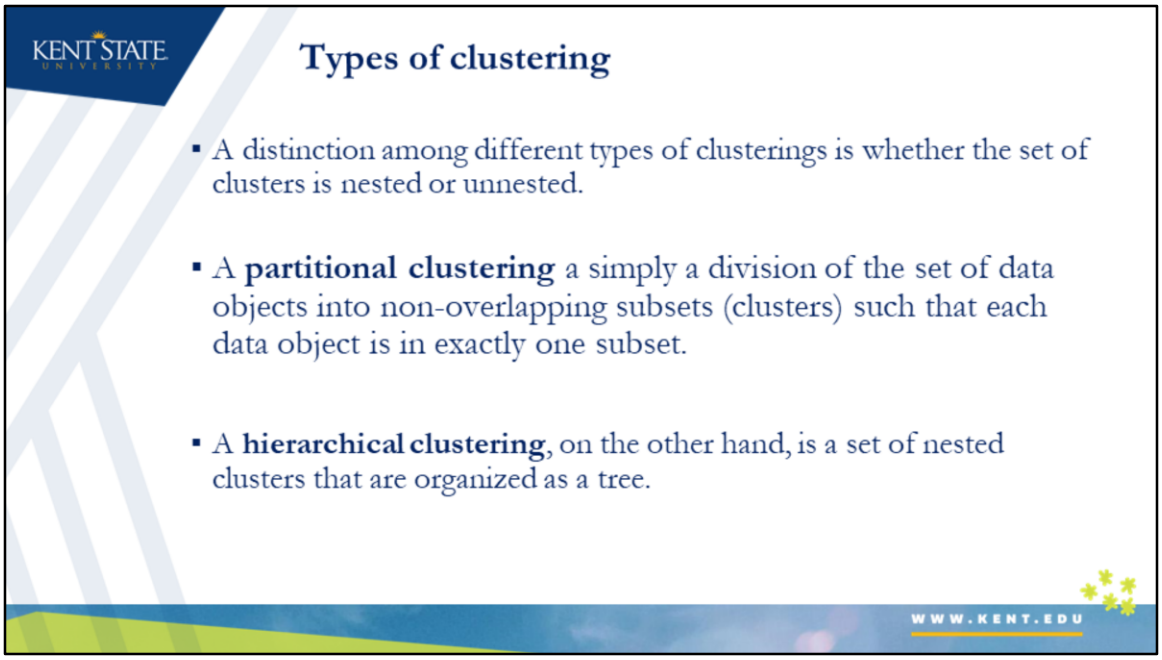
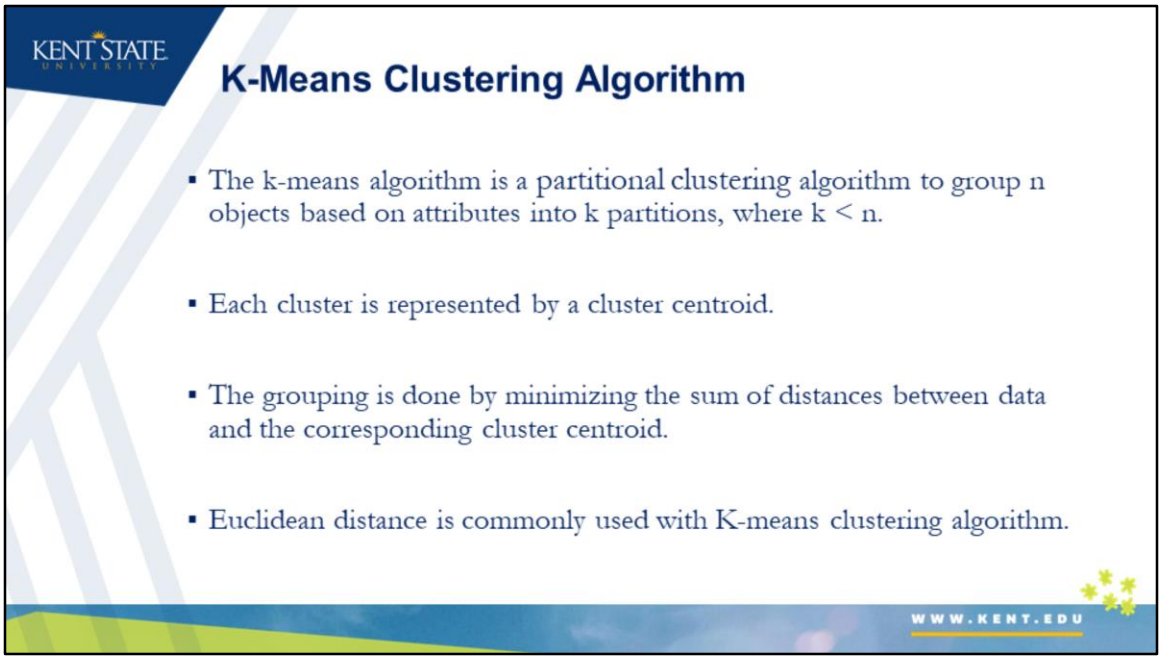
Let us now consider how the k-Means clustering algorithm works.



k-Means is a non-hierarchical clustering algorithm. Here, we specify the number of clusters k to group the data into non-overlapping sets.



All clustering algorithms rely on the concept of distance; Distance between records, and distance between clusters.

While distances dij between two records i and j can be defined in many ways, they in general have the following properties:

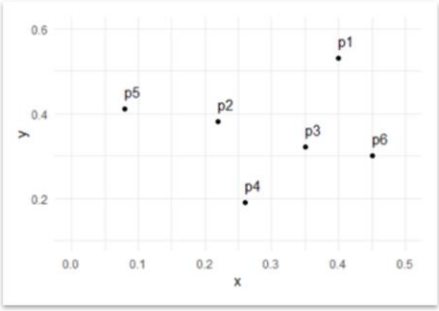
1. Non-negative: $d_{ij} \geq 0$
2. Self-proximity: $d_{ii} = 0$ (the distance from a record to itself is zero)
3. Symmetry: $d_{ij} = d_{ji}$
4. Triangle inequality: $d_{ij} \leq d_{ik} + d_{kj}$ (the distance between any pair cannot exceed the sum of distances between the other two pairs)

Secondly, where are the distances calculated from? For two records, this is simple. From the center of the observation. For a cluster, we normally use the centroid as the point from where the distances are calculated from. The most common measure of distance is the Euclidean distance, which we illustrate in the next few slides.

KENT STATE UNIVERSITY

Euclidean Distance: An Example

- Consider a set of 6 two-dimensional points



	X	Y
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

- Recall the formula for calculating Euclidean distance:

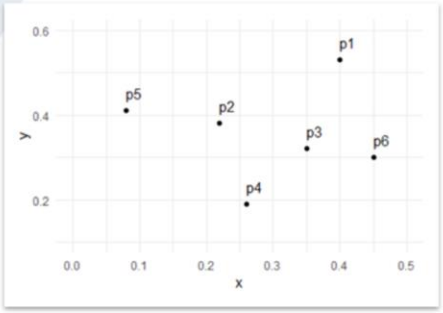
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

WWW.KENT.EDU

Here we are calculating the Euclidean distances between pairs of points. So, for example, the distance from p1 to p2 is sqrt((.4-.22)^2 + (.53-.38)^2) = 0.234. Note that the distance from p1 to p2 is the same as the distance from p2 to p1, and the distance of a point from itself is always 0.

KENT STATE UNIVERSITY

Euclidean Distance: An Example II



Euclidean Distance Matrix for 6 Points

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

WWW.KENT.EDU

As we discussed earlier, the distance from a point to itself will be 0, and dij = dji. Given that, we only need to specify distances as an upper or lower triangular matrix, as shown in the Table above.

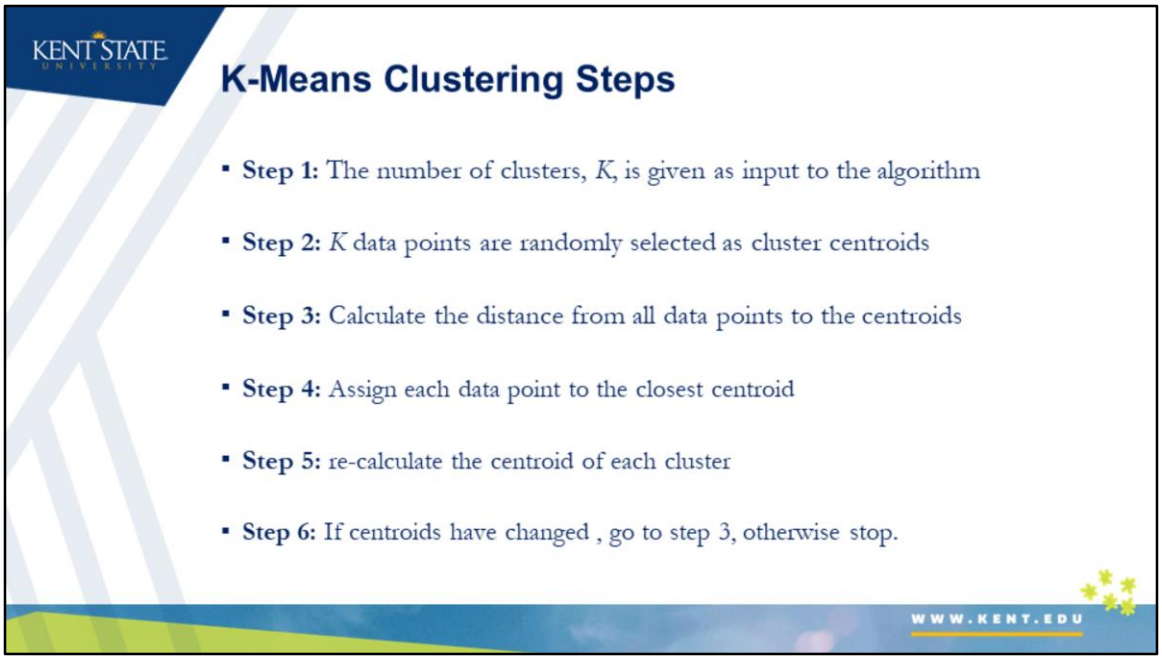
An important point to note is that the measure computed above is highly influenced by the scale of each variable, so that variables with larger scales have a much greater influence over the total distance. It is therefore customary to normalize continuous measurements before computing the Euclidean distance. This converts all measurements to the same scale. Normalizing a measurement means subtracting the average and dividing by the standard deviation (normalized values are also called z-scores).

The choice of the distance measure plays a major role in cluster analysis. The main guideline is domain dependent: What exactly is being measured? How are the different measurements related? What scale should each measurement be treated as (numerical, ordinal, or nominal)? Are there outliers? Finally, depending on the goal of the analysis, should the clusters be distinguished mostly by a small set of measurements, or should they be separated by multiple measurements that weight moderately?

Although Euclidean distance is the most widely used distance, it has three main features that need to be kept in mind.

1. It is scale dependent
2. It completely ignores any relationship between measurements, so if the measurements are correlated, maybe other measures of distances might be better
3. It is sensitive to outliers.

These are issues you should keep in mind.



A non-hierarchical approach to forming good clusters is to pre-specify a desired number of clusters, k, and assign each case to one of the k clusters so as to minimize a measure of dispersion within the clusters. In other words, the goal is to divide the sample into a predetermined number k of non-overlapping clusters so that clusters are as homogeneous as possible with respect to the measurements used.

A common measure of within-cluster dispersion is the sum of distances (or sum of squared Euclidean distances) of records from their cluster centroid.

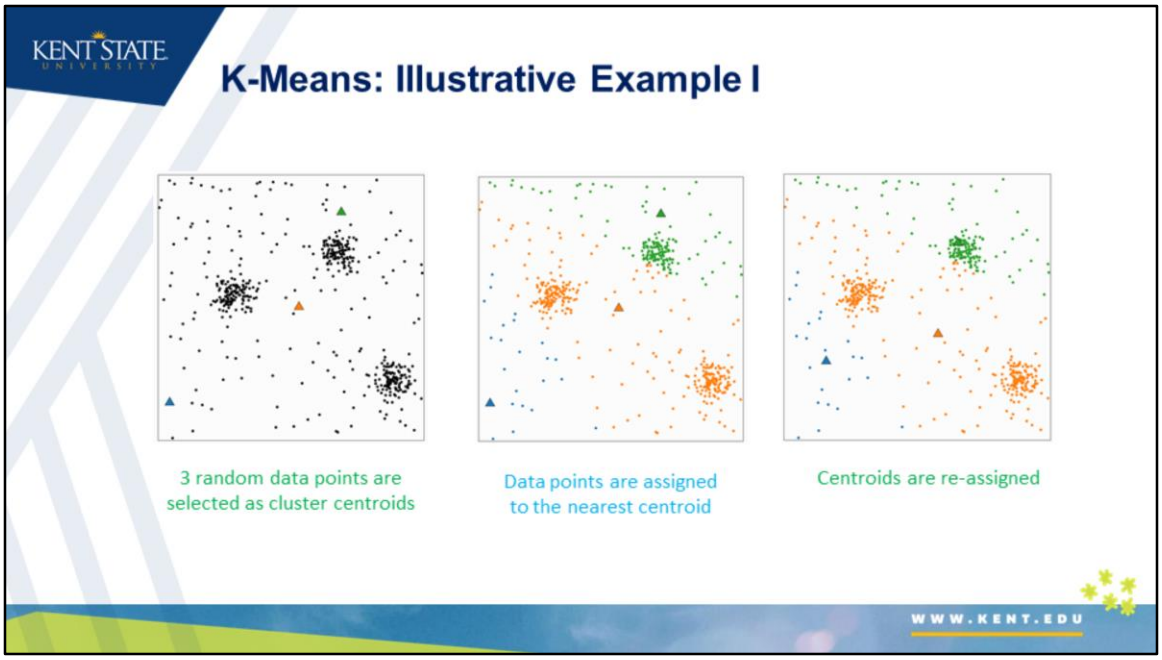
The k-means algorithm starts with an initial partition of the records into k clusters. Subsequent steps modify the partition to reduce the sum of the distances of each record from its cluster centroid. The modification consists of allocating each record to the nearest of the k centroids of the previous partition. This leads to a new partition for which the sum of distances is smaller than before. The means of the new clusters are computed and the improvement step is repeated until the improvement is very small.

There are a few key points to note.

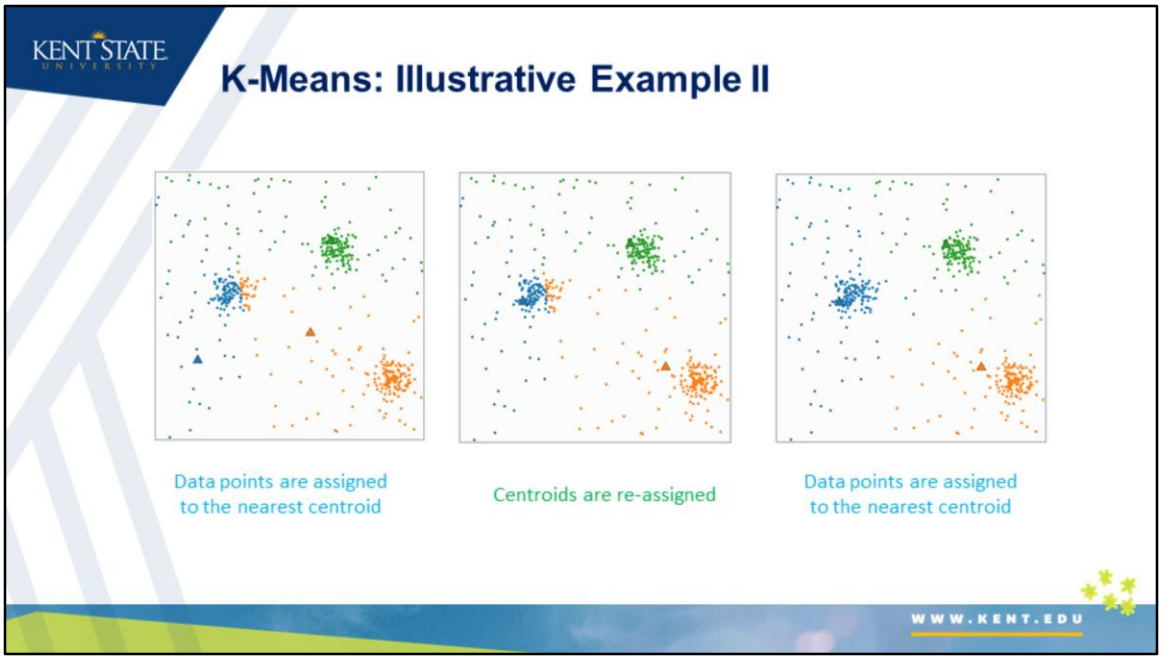
1. The choice of the number of clusters can either be driven by external

considerations (e.g., previous knowledge, practical constraints, etc.), or we can try a few values

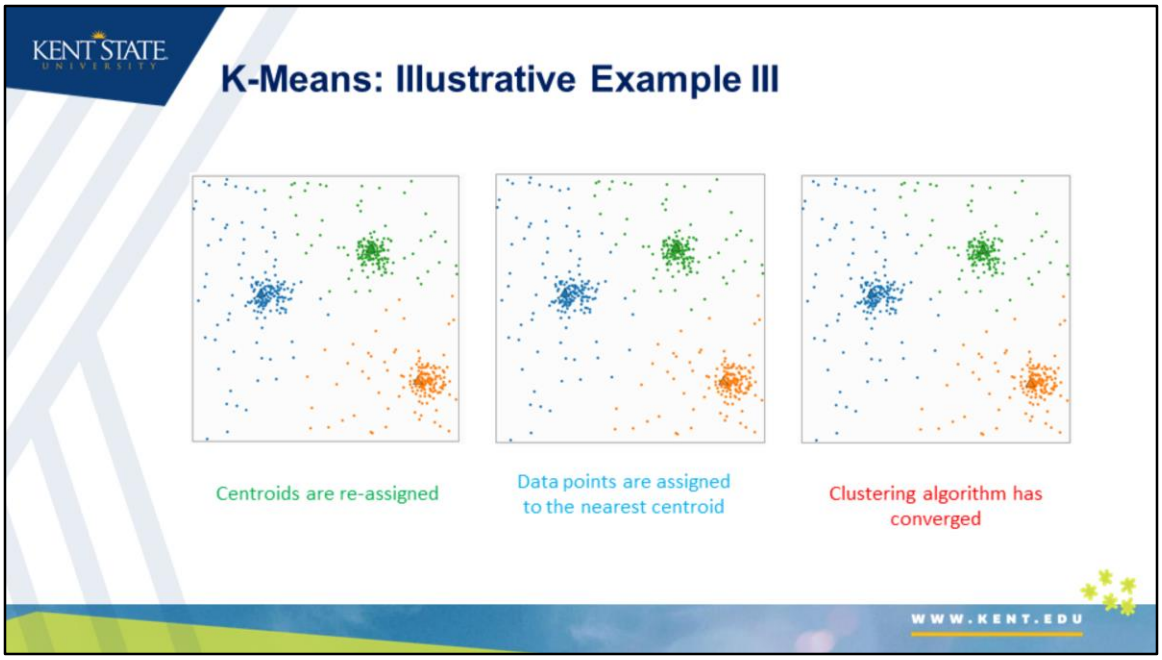
2. After choosing k, the n records are partitioned into these initial clusters. If there is external reasoning for the initial assignment, we can apply that, but otherwise, this is done randomly. In these cases, the algorithm can be rerun with different randomly generated starting partitions to reduce chances of the heuristic producing a poor solution.



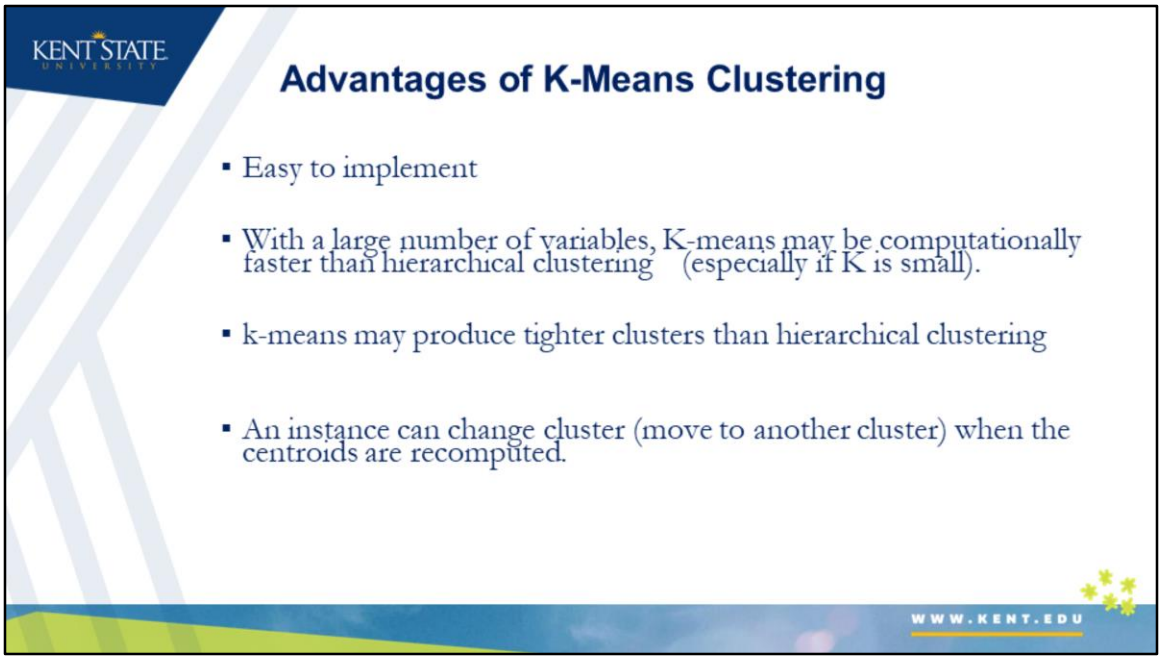
This illustrates the starting solution for the k-Means algorithm where the assignment is done at random. In subsequent steps, data is then assigned to the centroid to which it is closest, and the centroid then recalculated. In each iteration, points are assigned / reassigned to the centroid to which it is closest. This may shift both the centroid and the assignment of points. This is illustrated in the next few slides.



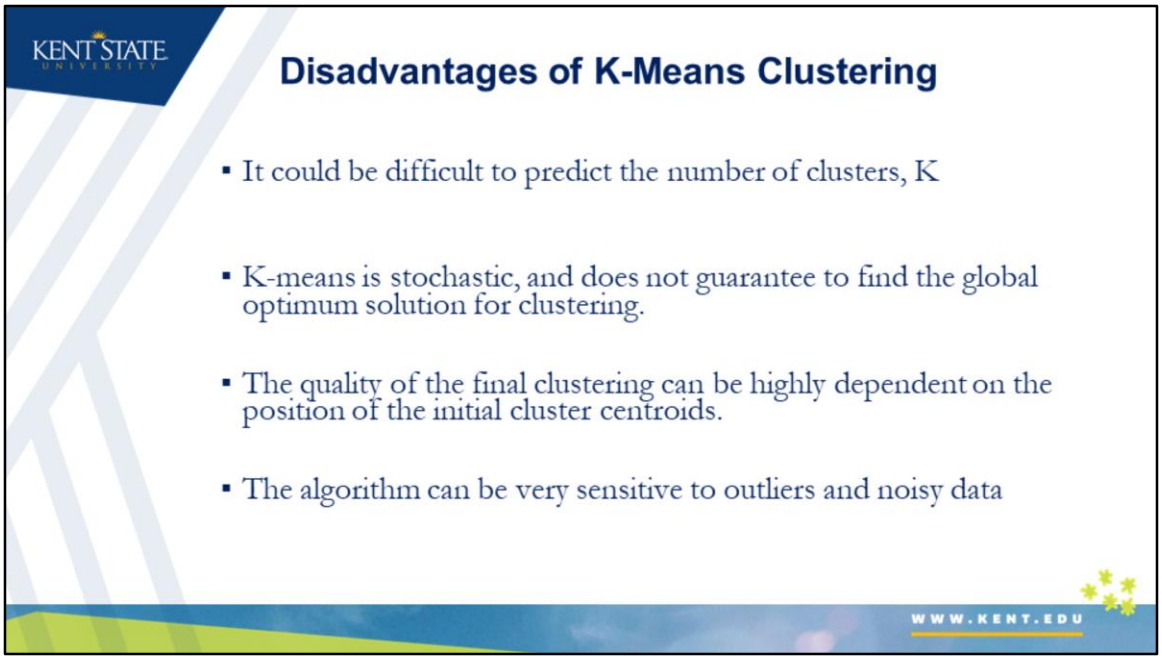
Here the centroids are reassigned and data points are assigned to the nearest centroid.



Eventually, the clustering algorithm should converge when no more improvements are possible. The algorithm is then repeated with a new starting assignment of points to clusters.



While the k-Means clustering algorithm can be solved by modeling it as an Integer Programming optimization problem, the heuristic presented here is easy to implement and is computationally efficient, especially for small k. The major difficulty is in deciding on the choice of k, though there are tools to help with that.

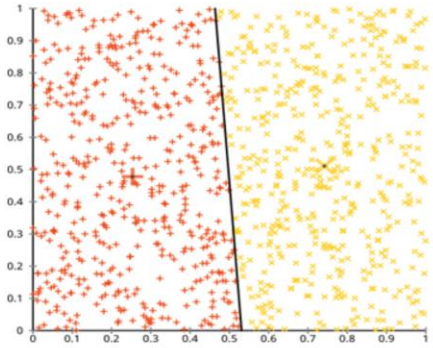


The k-means algorithm is stochastic in nature. By that we mean that if we ran the algorithm on the data again, we may get a different result. This is because the initial assignment of points to clusters is usually random, and if that assignment changes, then the possible solutions also change. As such, we are not guaranteed to get (or know) if the solution we have is the best possible one. The algorithm is also sensitive to outliers and the scale of data.

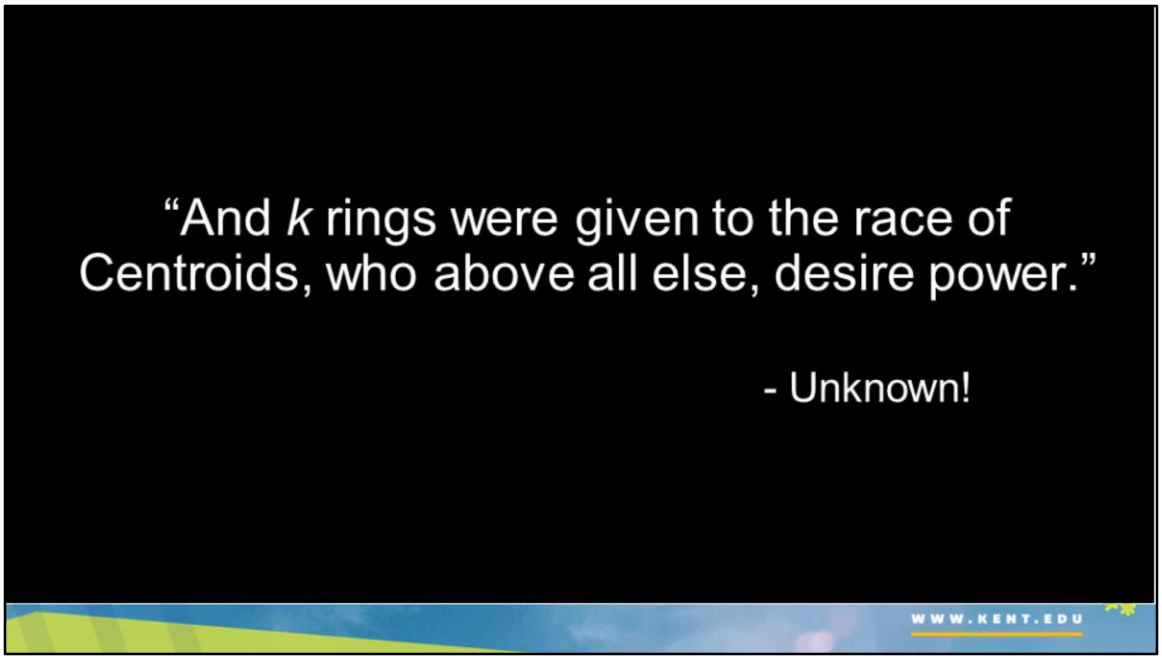
KENT STATE UNIVERSITY

Disadvantages of K-Means Clustering II

- K-means may give you clusters that do not exists! Running k-means on uniform data:



Having domain knowledge is important for the application of any ML method, and that applies to the k-means algorithm too.



This concludes our initial discussion on the k-means algorithm. We next examine the implementation of the algorithm in R.