

**DSC 40A**

*Theoretical Foundations of Data Science I*

# Announcements

- Homework 7 due 12/6.
- SET (currently = 50% < 80%)
- Review session on Friday

# Question

Answer at [q.dsc40a.com](http://q.dsc40a.com)

Remember, you can always ask questions at  
[q.dsc40a.com](http://q.dsc40a.com)!

If the direct link doesn't work, click the "Lecture Questions" link in the top right corner of [dsc40a.com](http://dsc40a.com).

# Lloyds Algorithm, or k-Means Clustering

1. Randomly initialize the k centroids.
2. Keep centroids fixed. Update groups.  
*Assign each point to the nearest centroid.*
3. Keep groups fixed. Update centroids.  
*Move each centroid to the center of its group.*
4. Repeat steps 2 and 3 until done.

# Outline

- Why does k-means clustering work?
- What are some practical considerations when using this algorithm?

# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) =$  total squared distance of each data point  $x_i$  to its nearest centroid  $\mu_j$

- Argue why updating the groups and centroids according to the algorithm reduces the cost with each iteration.
- With enough iterations, cost will be sufficiently small.

# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) =$  total squared distance of each data point  $x_i$  to its nearest centroid  $\mu_j$

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) = \text{Cost}(\mu_1) + \text{Cost}(\mu_2) + \dots + \text{Cost}(\mu_k)$  where


$\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) = \text{Cost}(\mu_1) + \text{Cost}(\mu_2) + \dots + \text{Cost}(\mu_k)$  where  
 $\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$   
to centroid  $\mu_j$

1. Randomly initialize the  $k$  centroids.

sets initial cost  
(before the  
process begins)





# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) = \text{Cost}(\mu_1) + \text{Cost}(\mu_2) + \dots + \text{Cost}(\mu_k)$  where  
 $\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$   
to centroid  $\mu_j$

2. Fix the centroids. Update the groups.

consider an  
arbitrary iteration


Certainly  $\text{Cost}(\mu_1, \mu_2, \dots, \mu_k)$  decreases in this step because assigning each point to the **closest** centroid is best.

# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) = \text{Cost}(\mu_1) + \text{Cost}(\mu_2) + \dots + \text{Cost}(\mu_k)$  where  
 $\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$   
to centroid  $\mu_j$

3. Fix the groups. Update the centroids.

consider an  
arbitrary iteration



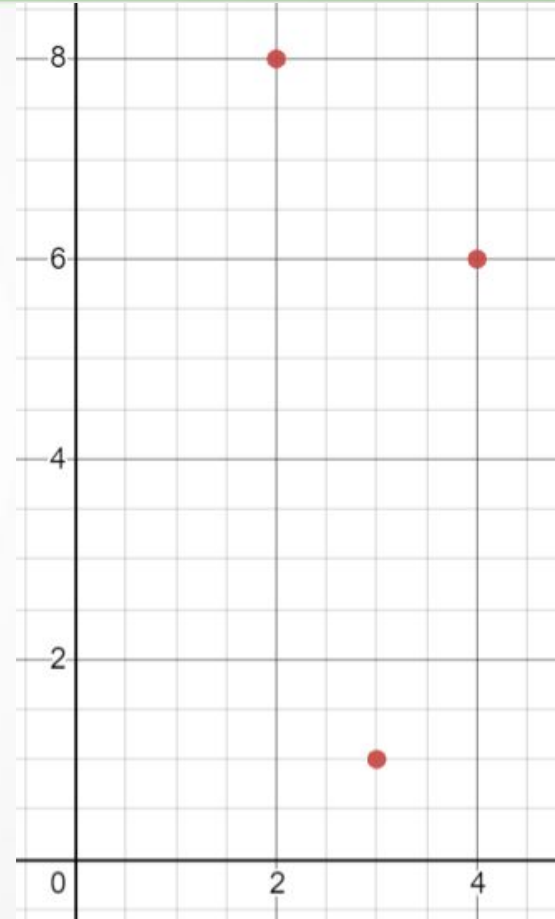
Argue that  $\text{Cost}(\mu_1, \mu_2, \dots, \mu_k)$  decreases in this step because for each group  $j$ ,  $\text{Cost}(\mu_j)$  is minimized when we update the centroid.

# Why does k-means clustering work?

$\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6)$ ,  $(2, 8)$ ,  $(3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?

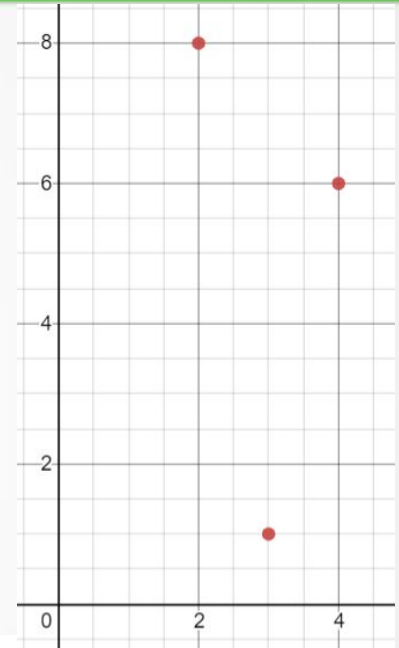


# Why does k-means clustering work?

$\text{Cost}(\mu_j)$  = total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6)$ ,  $(2, 8)$ ,  $(3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?



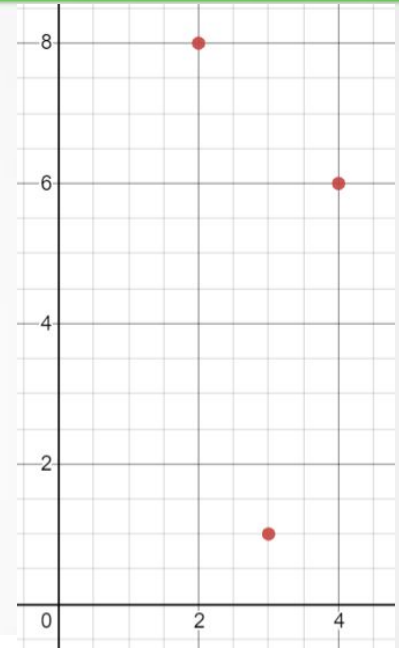
$$\begin{aligned}\text{Cost}(\mu_j) &= \left(\sqrt{(4 - c_1)^2 + (6 - c_2)^2}\right)^2 + \left(\sqrt{(2 - c_1)^2 + (8 - c_2)^2}\right)^2 + \left(\sqrt{(3 - c_1)^2 + (1 - c_2)^2}\right)^2 \\ &= (4 - c_1)^2 + (6 - c_2)^2 + (2 - c_1)^2 + (8 - c_2)^2 + (3 - c_1)^2 + (1 - c_2)^2\end{aligned}$$

# Why does k-means clustering work?

$\text{Cost}(\mu_j)$  = total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6)$ ,  $(2, 8)$ ,  $(3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?



$$\begin{aligned}\text{Cost}(\mu_j) &= \left(\sqrt{(4 - c_1)^2 + (6 - c_2)^2}\right)^2 + \left(\sqrt{(2 - c_1)^2 + (8 - c_2)^2}\right)^2 + \left(\sqrt{(3 - c_1)^2 + (1 - c_2)^2}\right)^2 \\ &= (4 - c_1)^2 + (6 - c_2)^2 + (2 - c_1)^2 + (8 - c_2)^2 + (3 - c_1)^2 + (1 - c_2)^2\end{aligned}$$

$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_1} = 2(c_1 - 4) + 2(c_1 - 2) + 2(c_1 - 3)$$

$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_2} = 2(c_2 - 6) + 2(c_2 - 8) + 2(c_2 - 1)$$

# Why does k-means clustering work?

$\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6)$ ,  $(2, 8)$ ,  $(3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?

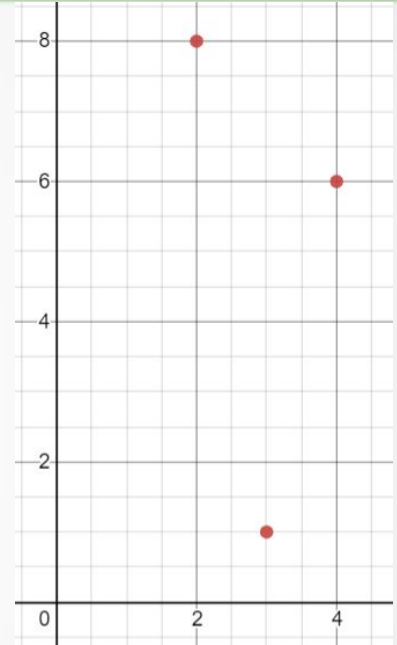
$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_1} = 2(c_1 - 4) + 2(c_1 - 2) + 2(c_1 - 3)$$

$$0 = 2(c_1 - 4) + 2(c_1 - 2) + 2(c_1 - 3)$$

$$0 = c_1 - 4 + c_1 - 2 + c_1 - 3$$

$$3c_1 = 4 + 2 + 3$$

$$c_1 = \frac{4 + 2 + 3}{3}$$



# Why does k-means clustering work?

$\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6)$ ,  $(2, 8)$ ,  $(3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?

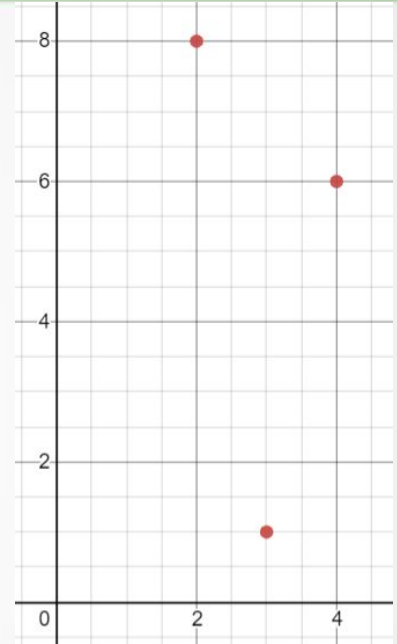
$$\frac{\partial \text{Cost}(\mu_j)}{\partial c_2} = 2(c_2 - 6) + 2(c_2 - 8) + 2(c_2 - 1)$$

$$0 = 2(c_2 - 6) + 2(c_2 - 8) + 2(c_2 - 1)$$

$$0 = c_2 - 6 + c_2 - 8 + c_2 - 1$$

$$3c_2 = 6 + 8 + 1$$

$$c_2 = \frac{6 + 8 + 1}{3}$$



# Why does k-means clustering work?

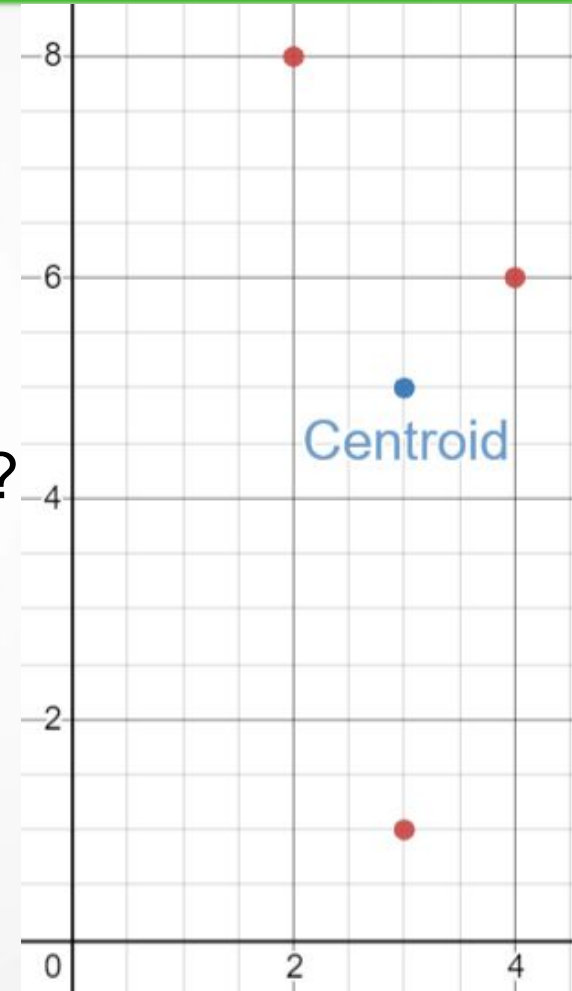
$\text{Cost}(\mu_j) =$  total squared distance of each data point  $x_i$  in group  $j$  to centroid  $\mu_j$

Example: group  $j$  contains  $(4, 6), (2, 8), (3, 1)$

How to place centroid  $\mu_j = (c_1, c_2)$  to minimize cost?

$$(c_1, c_2) = \left( \frac{4+2+3}{3}, \frac{6+8+1}{3} \right) = (3, 5)$$

Minimize cost by averaging in each coordinate.





# Cost, Loss, and Risk

The cost of placing the centroid at  $(c_1, c_2)$  is

$$\begin{aligned}\text{Cost}(\mu_j) &= \left(\sqrt{(4 - c_1)^2 + (6 - c_2)^2}\right)^2 + \left(\sqrt{(2 - c_1)^2 + (8 - c_2)^2}\right)^2 + \left(\sqrt{(3 - c_1)^2 + (1 - c_2)^2}\right)^2 \\ &= (4 - c_1)^2 + (6 - c_2)^2 + (2 - c_1)^2 + (8 - c_2)^2 + (3 - c_1)^2 + (1 - c_2)^2\end{aligned}$$

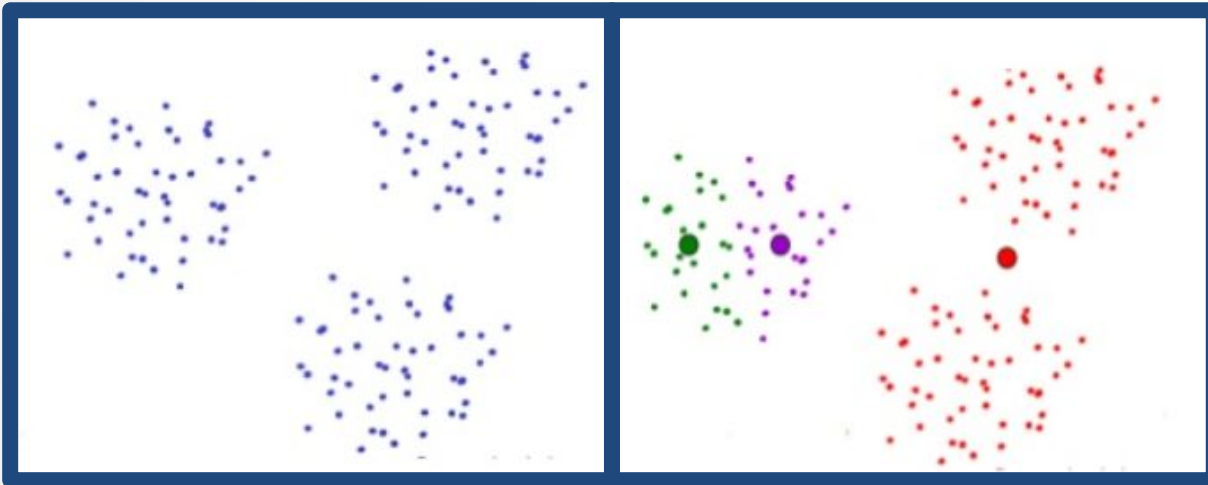
# Why does k-means clustering work?

$\text{Cost}(\mu_1, \mu_2, \dots, \mu_k) =$  total squared distance of each data point  $x_i$  to its nearest centroid  $\mu_j$

- Argue why updating the groups and centroids according to the algorithm reduces the cost with each iteration.
- With enough iterations, cost will be sufficiently small.

# k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.

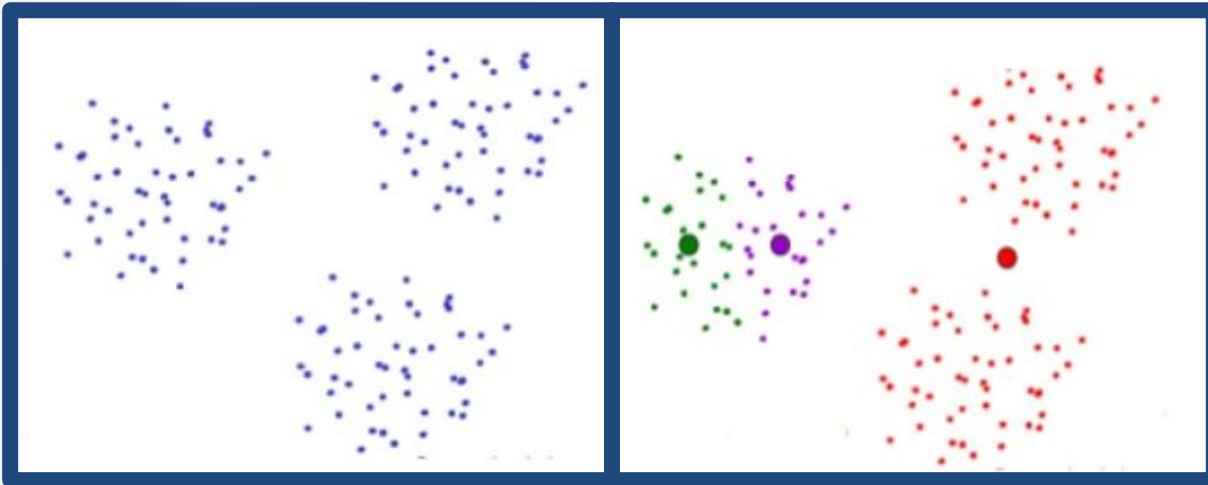


In general, how do we assess which result is the best?

- A. Clusters appear how we expect them to
- B. Clusters are evenly sized
- C. Cost function is lowest

# k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.



In general, how do we assess which result is the best?

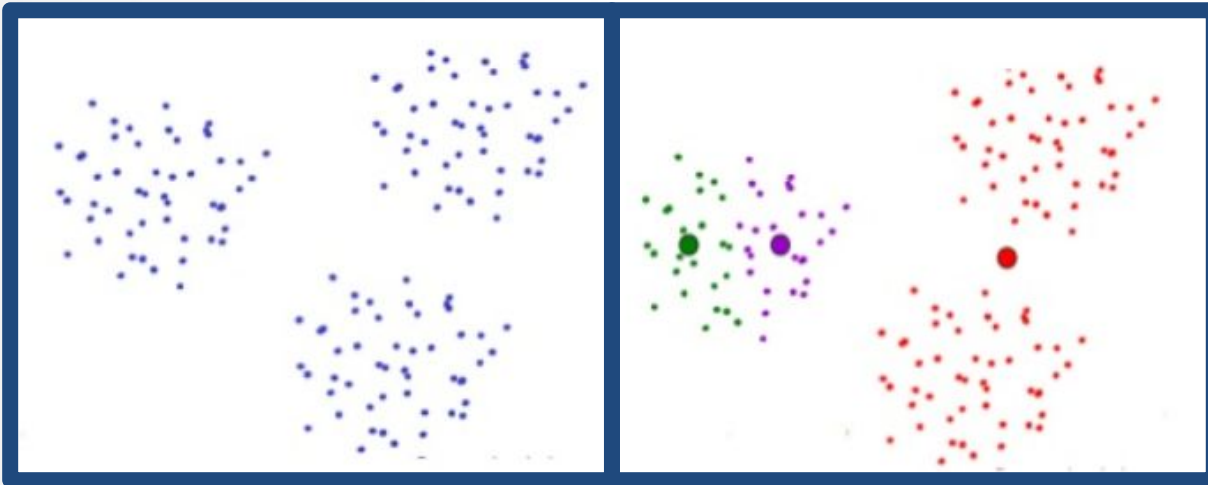
- A. Clusters appear how we expect them to
- B. Clusters are evenly sized
- C. Cost function is lowest

Solution?

- Try algorithm several times, pick the best result.
- Similar approach used in gradient descent.

# k-Means Clustering in Practice: Initialization

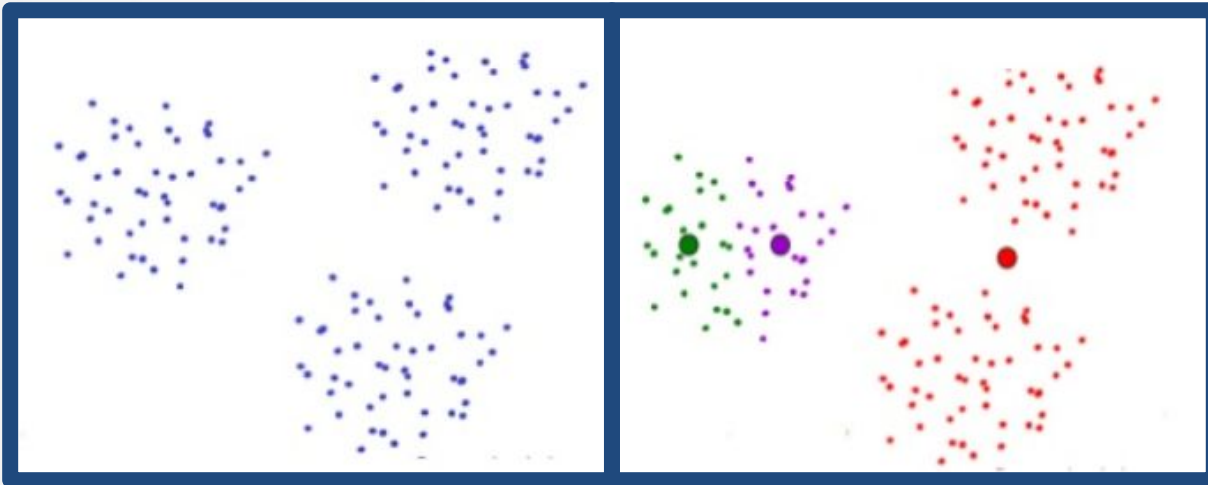
Can get unlucky with random initialization.



- No guarantees of a satisfactory solution with this algorithm.
- Brute force algorithm would try all assignments of points to clusters and choose the one with the lowest cost.

# k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.

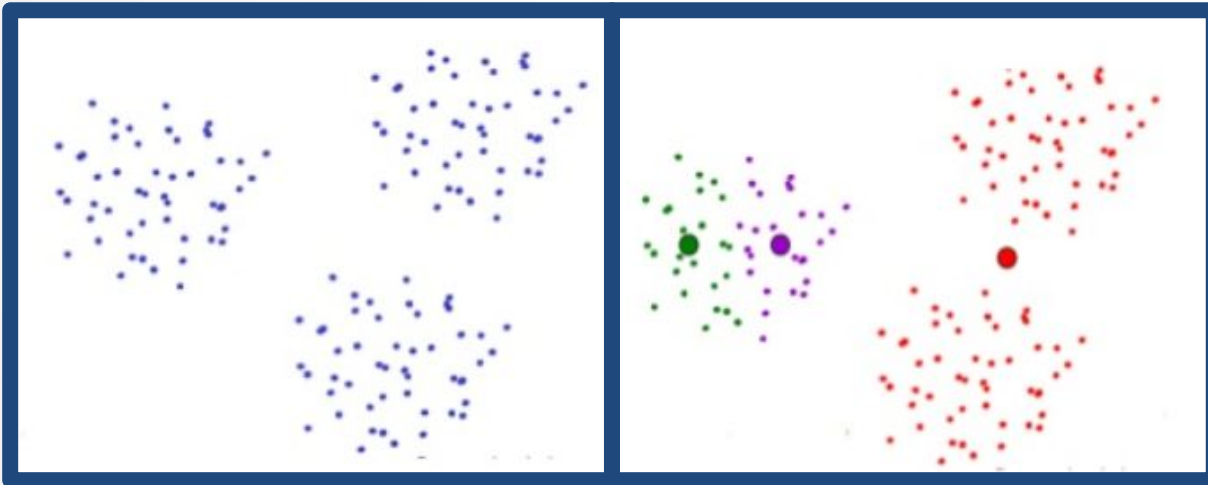


How many ways to assign  $n$  points to  $k$  clusters?

- No guarantees of a satisfactory solution with this algorithm.
- Brute force algorithm would try all assignments of points to clusters and choose the one with the lowest cost.

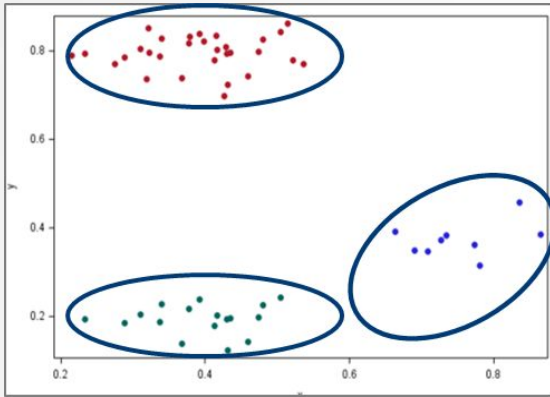
# k-Means Clustering in Practice: Initialization

Can get unlucky with random initialization.

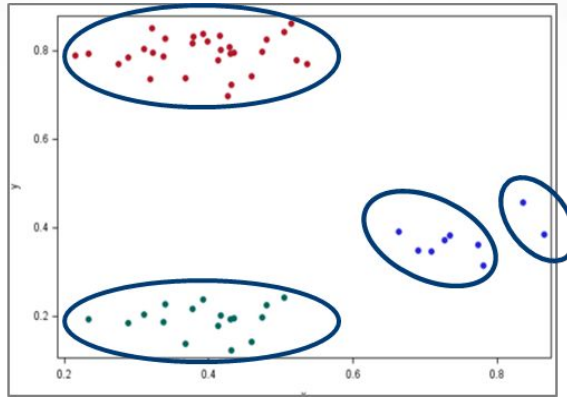


- No guarantees of a satisfactory solution with this algorithm.
- Any algorithm that is guaranteed to find the best coloring of data points takes exponential time (computationally infeasible).

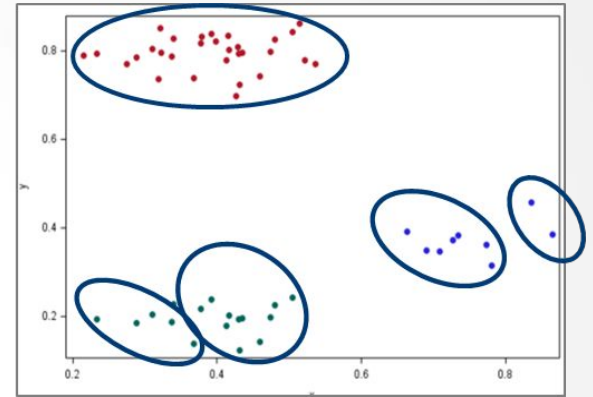
# k-Means Clustering in Practice: Choosing k



k=3

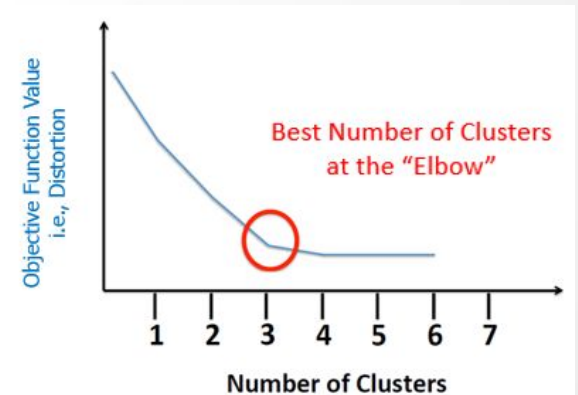


k=4



k=5

- Most commonly done by hand (visualizations, trial and error)
- Elbow method
- Context or domain knowledge
- Use a different clustering algorithm





# What if a centroid has no points in its group?

What should we do if a centroid has no points in its group?

- A. Terminate the algorithm.
- B. Wait for points get added to the group in a subsequent iteration.
- C. Set the centroid to be a data point, chosen at random.
- D. Set the centroid to be one of the other centroids, chosen at random.

# What if a centroid has no points in its group?

What should we do if a centroid has no points in its group?

- A. Terminate the algorithm.
- B. Wait for points get added to the group in a subsequent iteration.
- C. Set the centroid to be a data point, chosen at random.
- D. Set the centroid to be one of the other centroids, chosen at random.

Two options:

- Eliminate that centroid and find  $k-1$  clusters instead
- Randomly re-initialize that centroid

# Summary

- We saw that k-means clustering works because each step of the algorithm reduces the cost function, which measures the quality of a set of centroids.
- We discussed some practical considerations, including random initialization and choice of  $k$ .