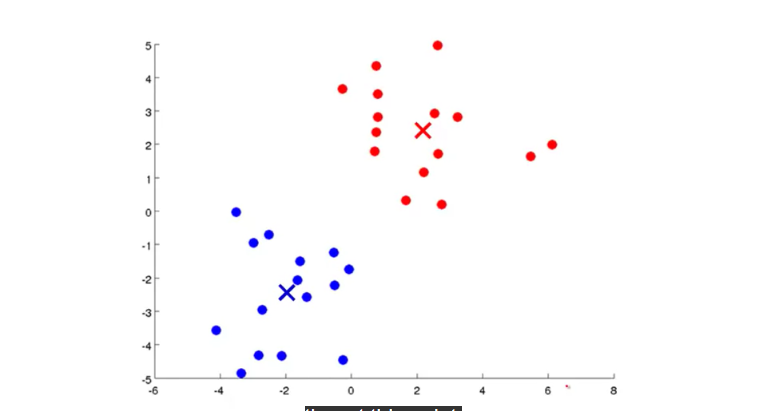


We repeat the above-mentioned process again and again until we arrive at the right classification:



If you keep running additional iterations of K means from here the cluster centroids will not change any further and the colours of the points will not change any further. And so, this is the, at this point, K means has converged and it's done a pretty good job finding .

***K-means algorithm explained mathematically:***

Note:

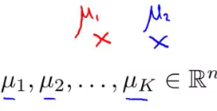
1.k=1 to K represents number of cluster centroids.

2.i=1 to m represents the number of datasets/rows of data

***Steps in which k means algorithm works:***

1.We first randomly initialise K cluster centroids at positions mu1, mu2,…….,muk

Note that mu1 ,mu2…mun are n row vectors:



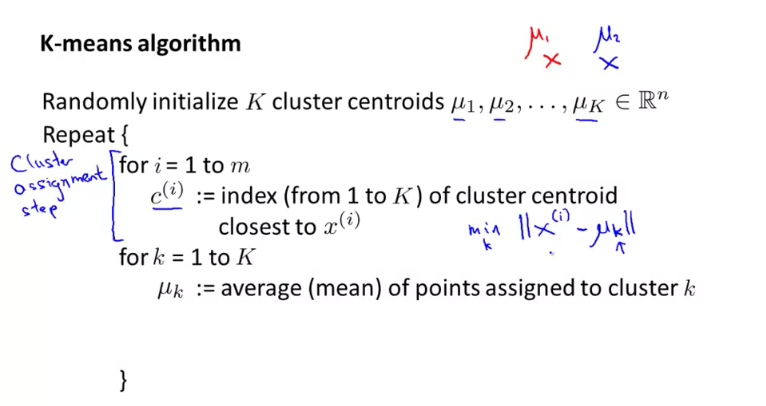
2. Let us consider we have only 2 columns/features in X i.e. x1 and x2. Now for each row of X, we plot x1 vs x2 and find the distance of these points from mu1 and mu2 and the points belong to the cluster centroid from which they have the minimum linear distance.

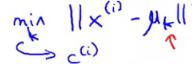
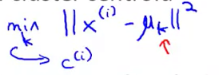
3. Say X1 has the least linear distance from mu2 and X2 from mu1 and X3 and X4 from mu1 and mu2 respectively and so on.

4. The we find the mean of the points belonging to mu1 and mu2 respectively. Since X1,X2… Xn are 2 row vectors( since they have 2 columns each representing 2 different features), their mean is also a 2 row vector . The mean of the points belonging to the two groups mu1 and mu2 indicates the new coordinates of mu1 and mu2.

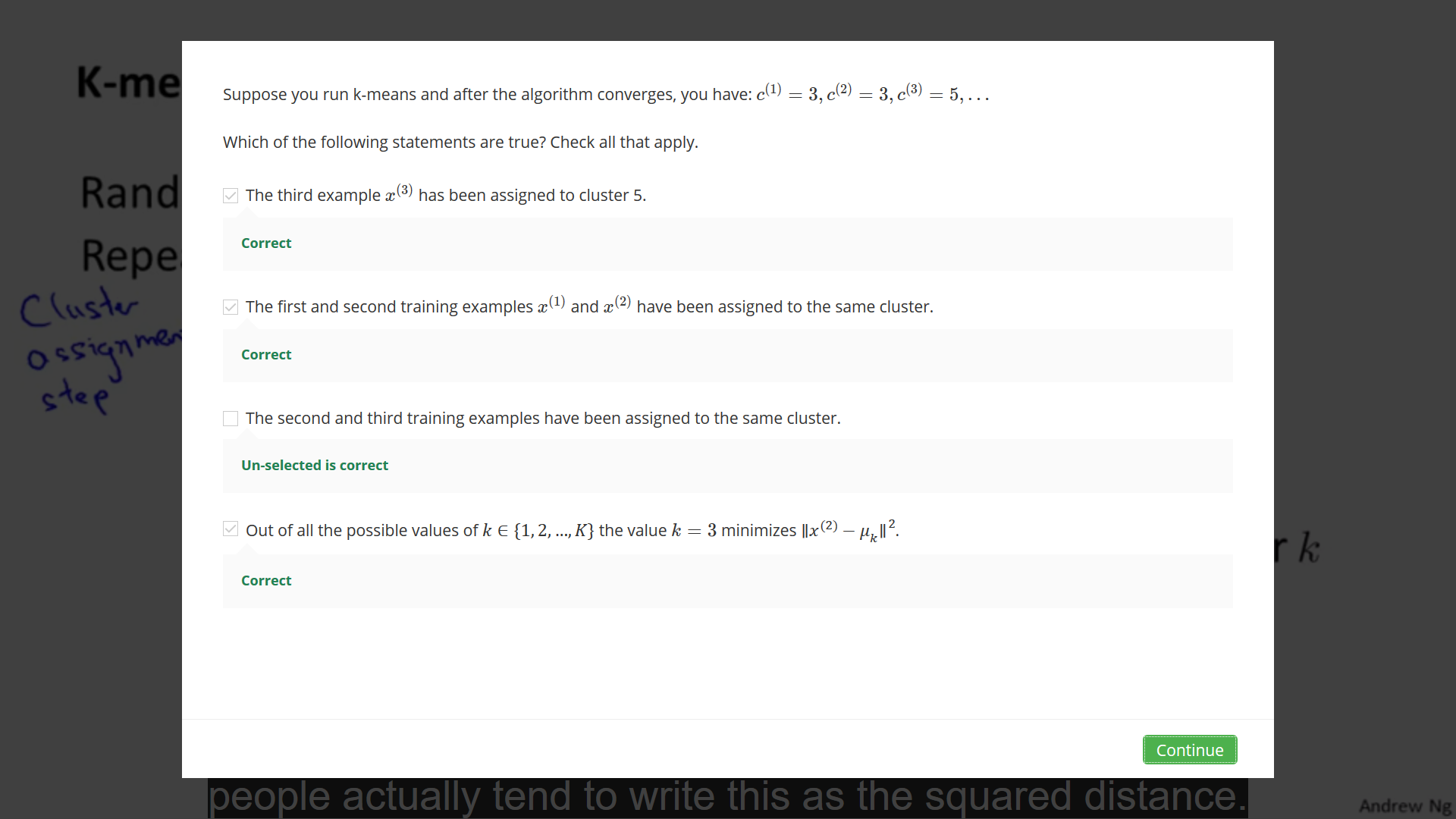
5. We then again find the distance of the points on the graph from mu1 and mu2 and the points belong to the cluster centroid from which they have the minimum linear distance.

6. Then we repeat steps 3 and 4.

The process is illustrated below:  


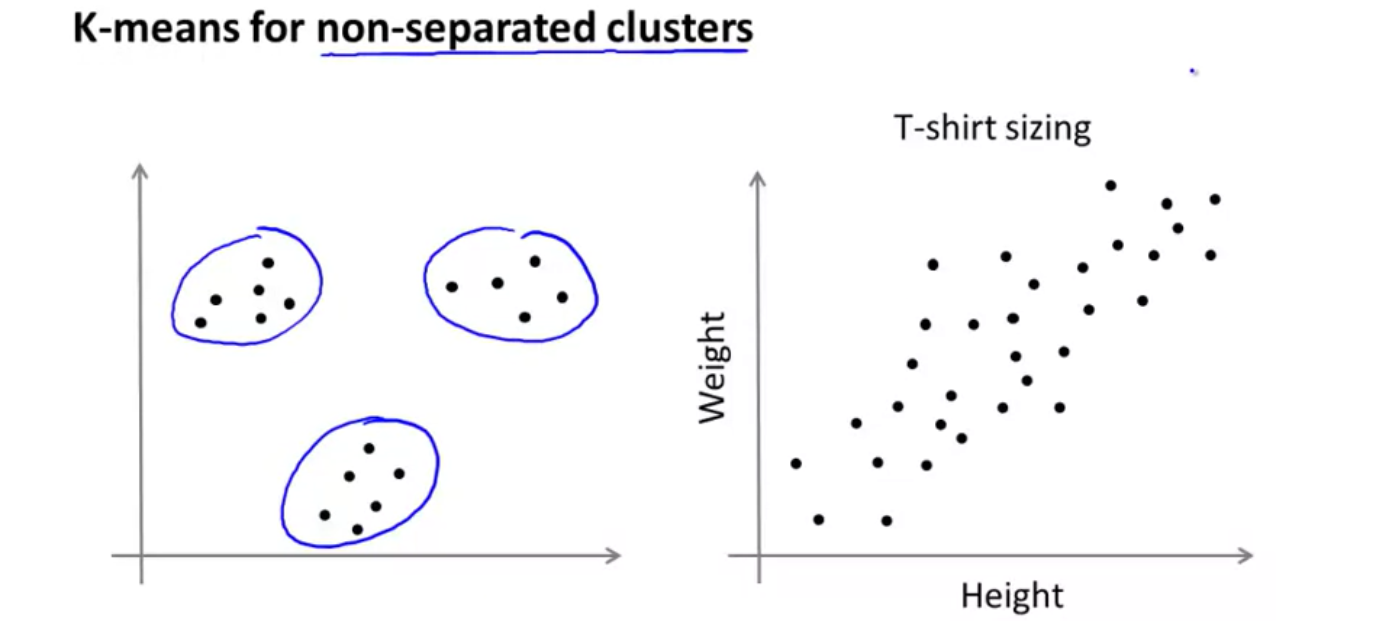
Instead of taking , we can also take i.e. square of the linear distance between cluster centroids and the points to find which points belong to which cluster centroid.

***Question:***

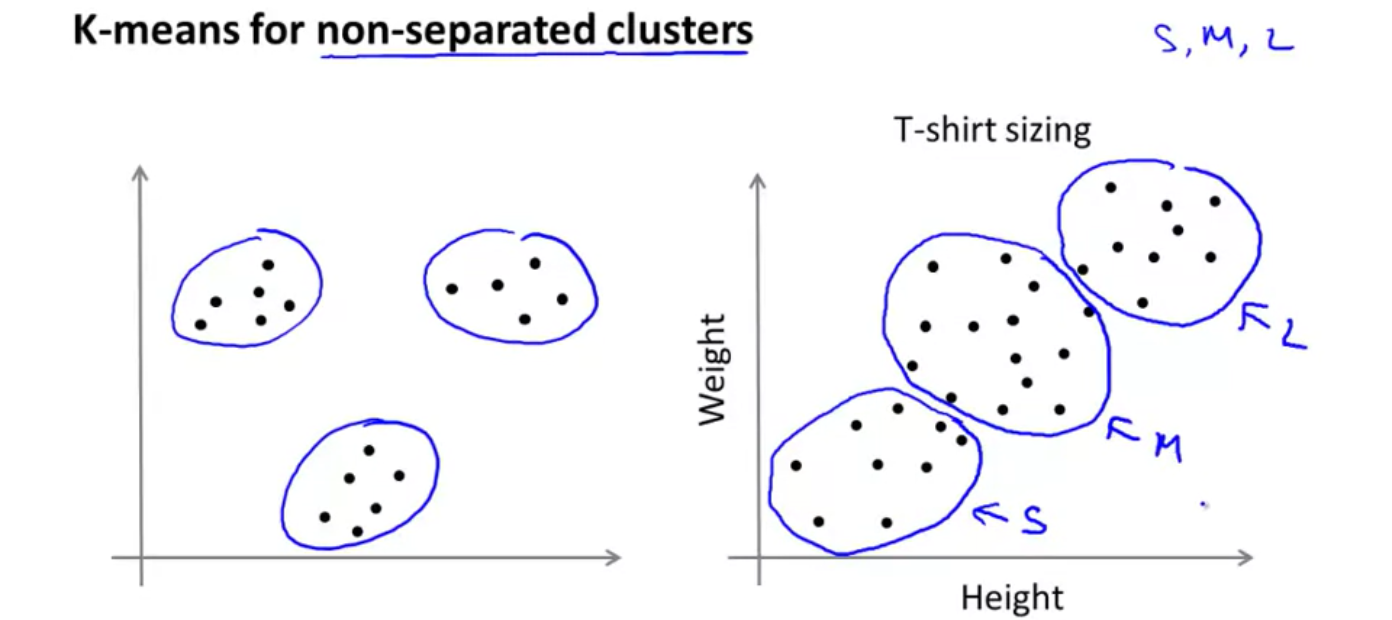


***K-means for non-separated clusters:***

In the graph shown on the right side of the figure , we see data that doesn’t seem to be easily separable into clusters:



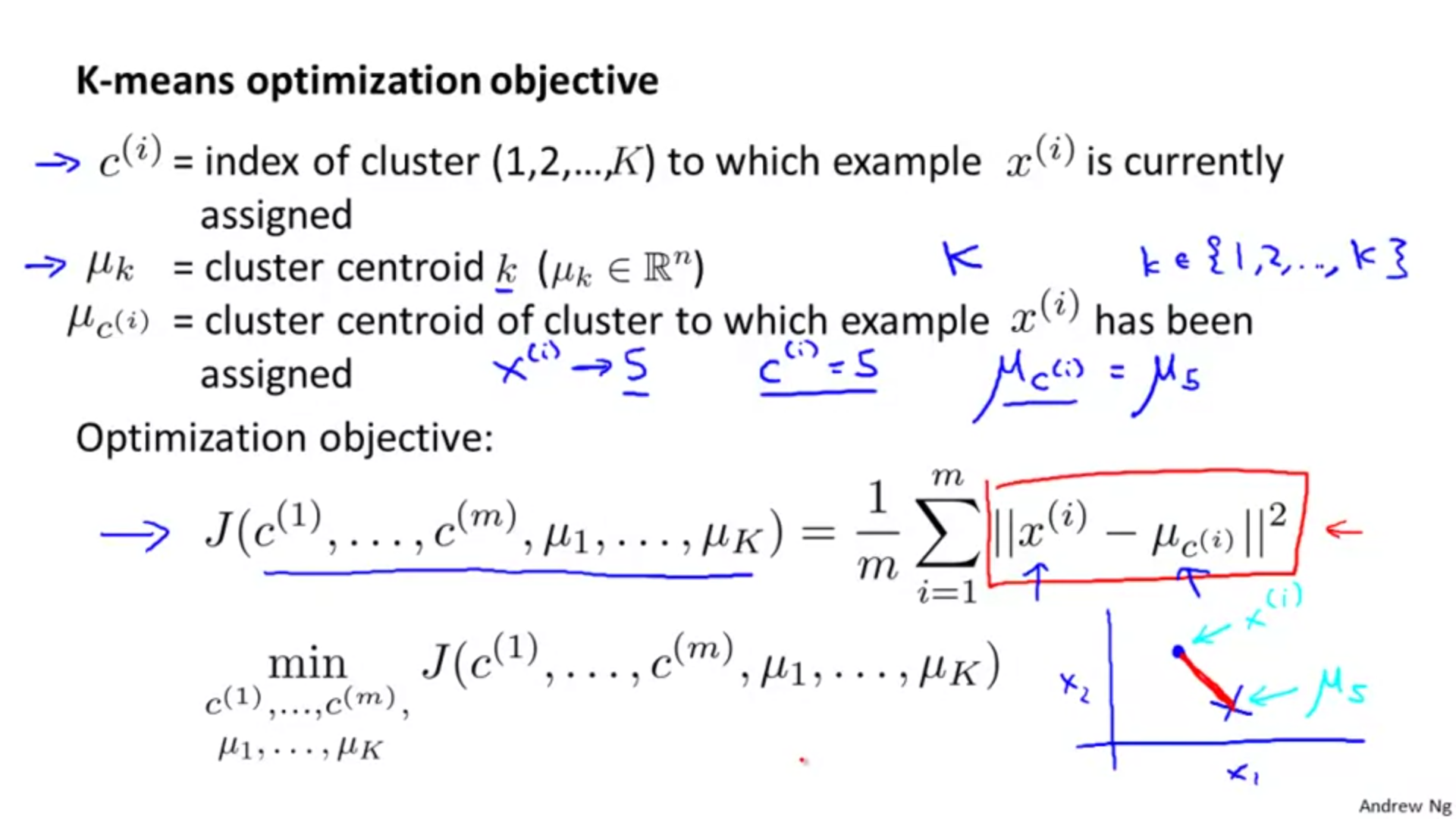
The K-means algorithm can separate this data into following cluster:



The clusters represent small, medium and large t-shirt sizes.

***Optimisation Objective:***

Refer to the image below:



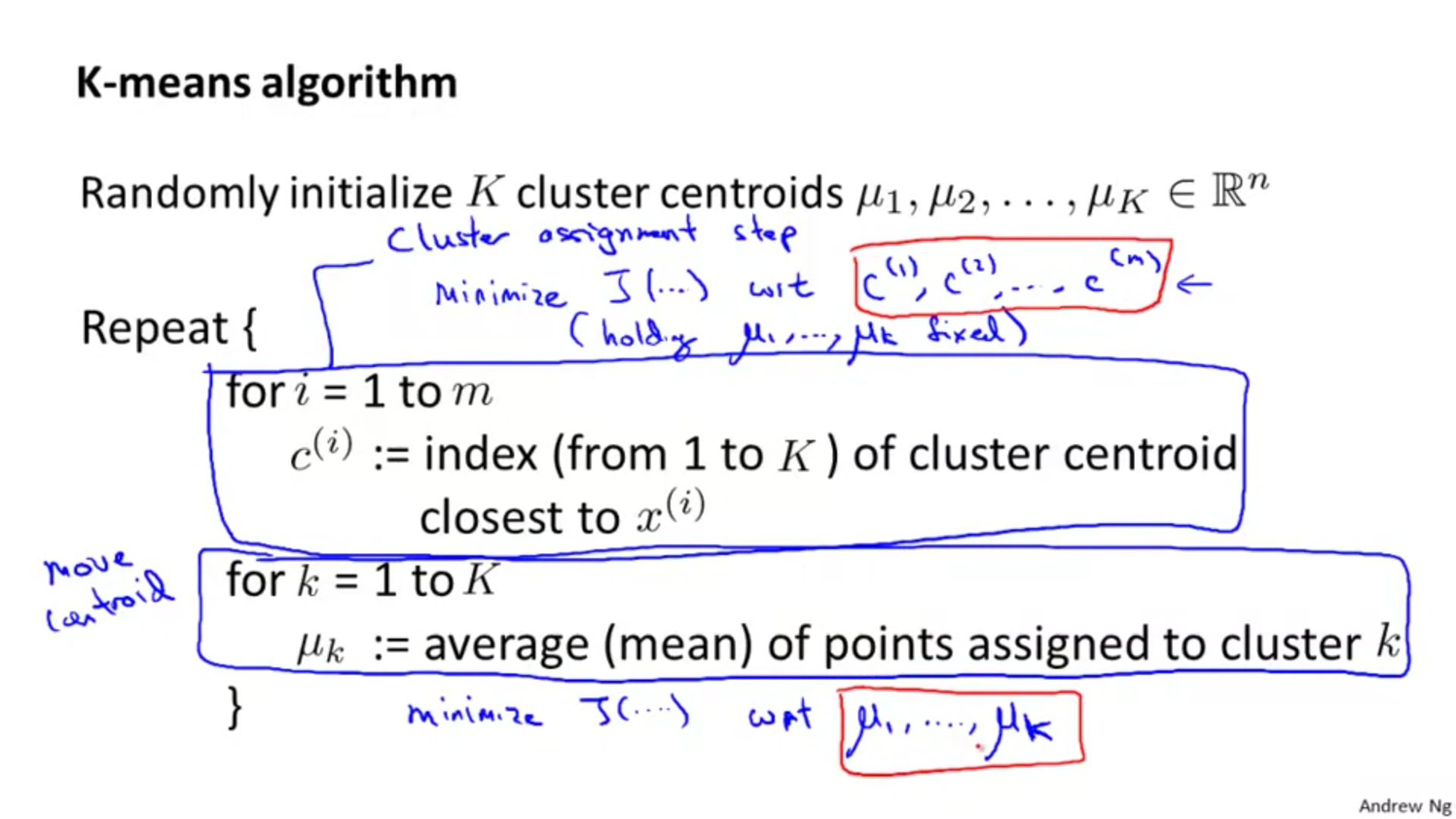
We want to minimize the cost function J by obtaining values of mu1 ,mu2…mun and c1,c2,…cn such that when we find the sum of the square of distance between the mean of the cluster centroid and a point say xi belonging to that cluster for all x1 to xm , the sum obtained is the minimum possible one.

The above lines mean that we want to classify the points in the best possible way.

***Showing how the cost function is calculated:***

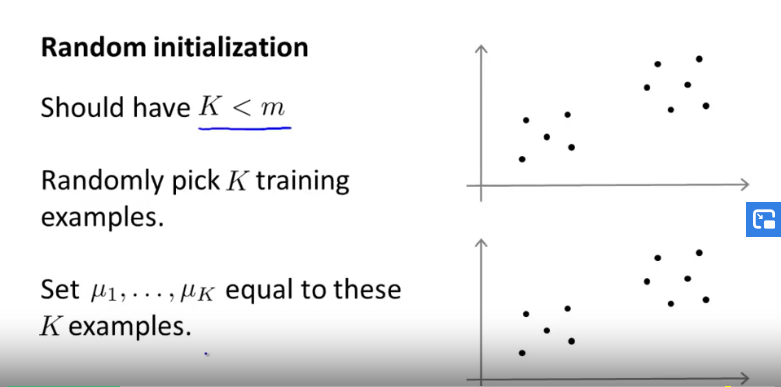
1.In the first step we randomly initialise mu values and find cost function for all i ranging from 1 to m.

2. In the second step we find mean of points assigned to cluster k and repeat step 1.

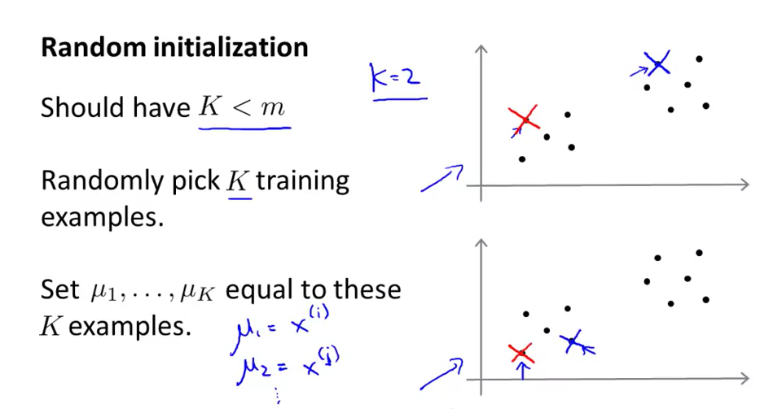


***Random initialisation:***

How do we initialise the mu1, mu2, ….. mun ?



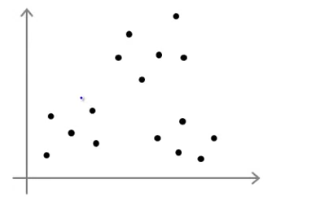
Now, it can be that we end up on different local optima values depending on how we initialise the mu values.



Above 2 graphs show 2 different initialisations of mu1 and mu2.

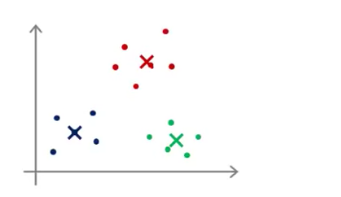
***The case of different local optima:***

Consider the graph below.

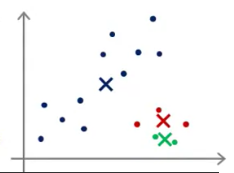


Now depending on how we initialise mu values, we may end up on different local optima and hence different clusters:

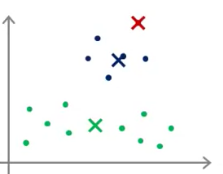
1. Initialisation 1:



2. Initialisation 2:



3. Initialisation 3:



***Solving the above problem***

We see above that different initialisations can lead to different clusters. So how do we solve this problem?

There are usually two cases that occur:

a) If K=2 – 10

Run the whole process of k-means algorithm anywhere between 50-1000 times depending on requirement. Then pick the value of clustering for which the cost is minimum. This prevents the algorithm from staying on the wrong local optima.

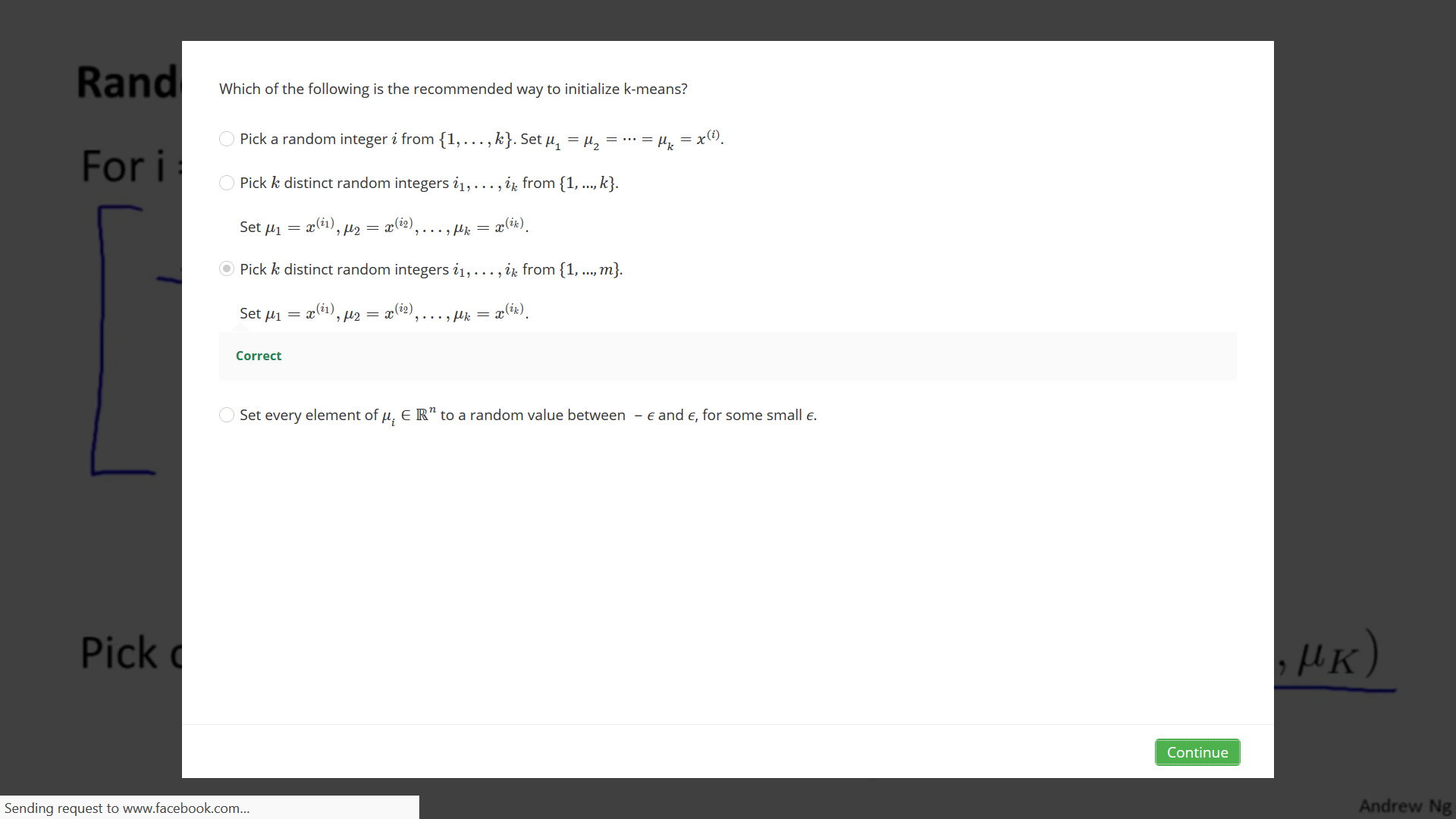


b) If K>10

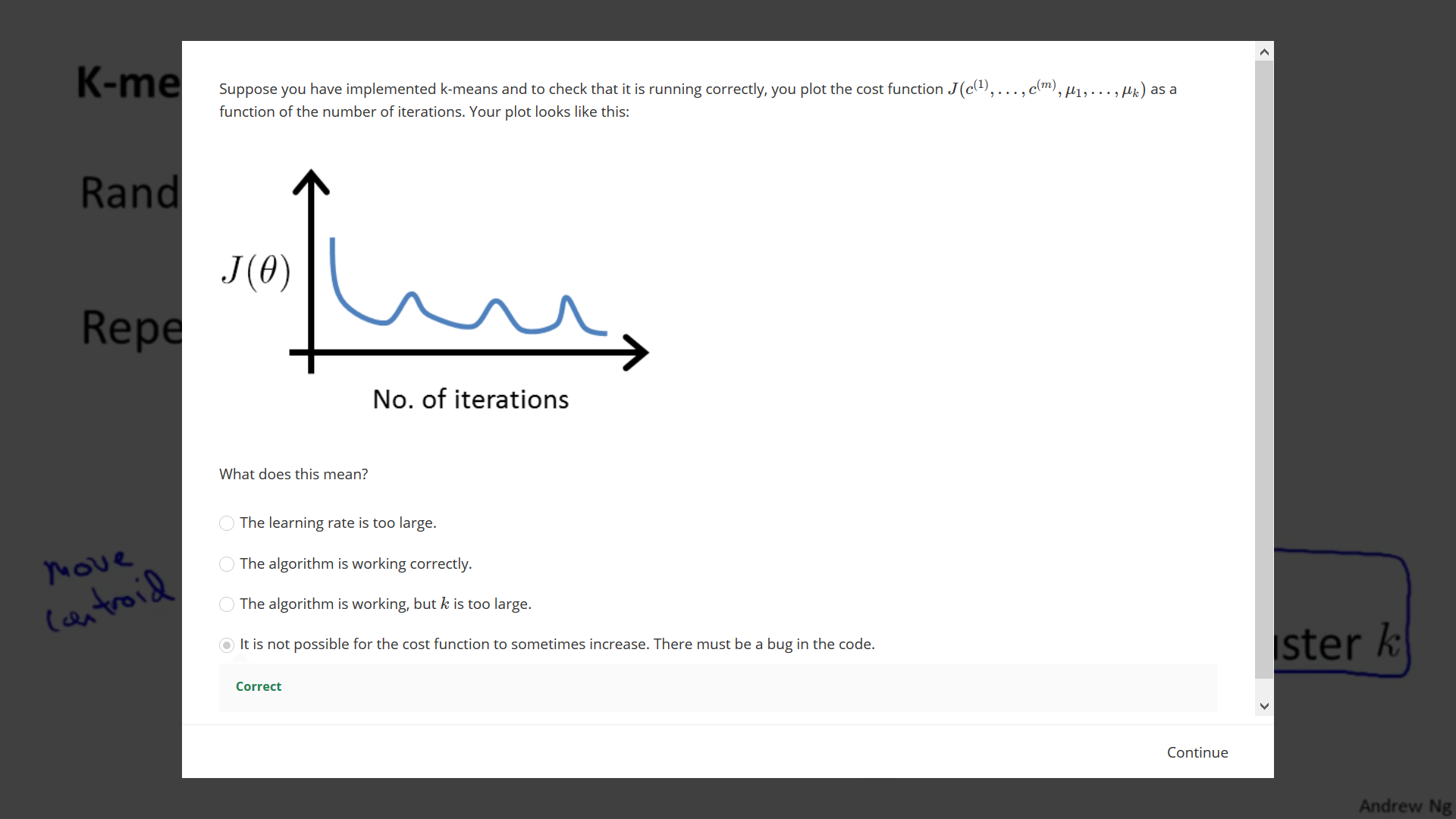
Usually doing the whole process pertaining to the algorithm once with single time initialisation of mu values produces the correct result. Although doing the whole process pertaining to the algorithm multiple times can lead to a little better result.

*Questions:*

*Q1:*

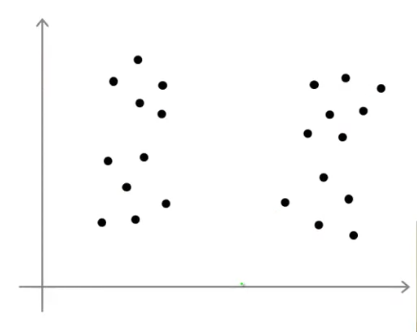


*Q2:*



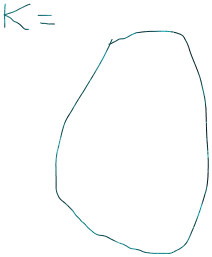
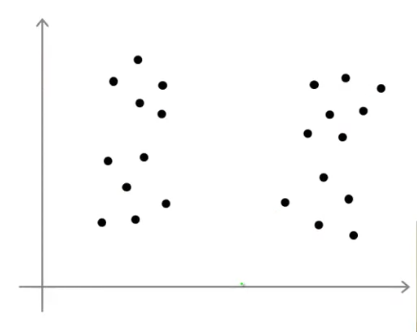
***Choosing the value of K:***

There is no particular way of choosing the value of K. Consider the figure below:

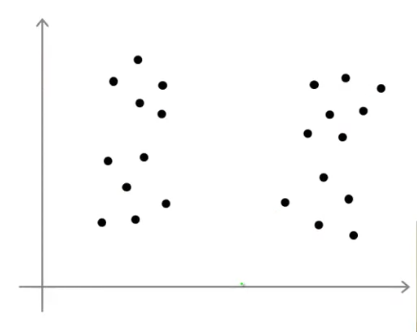


Now this can be made into clusters in many ways, two of which are shown below:

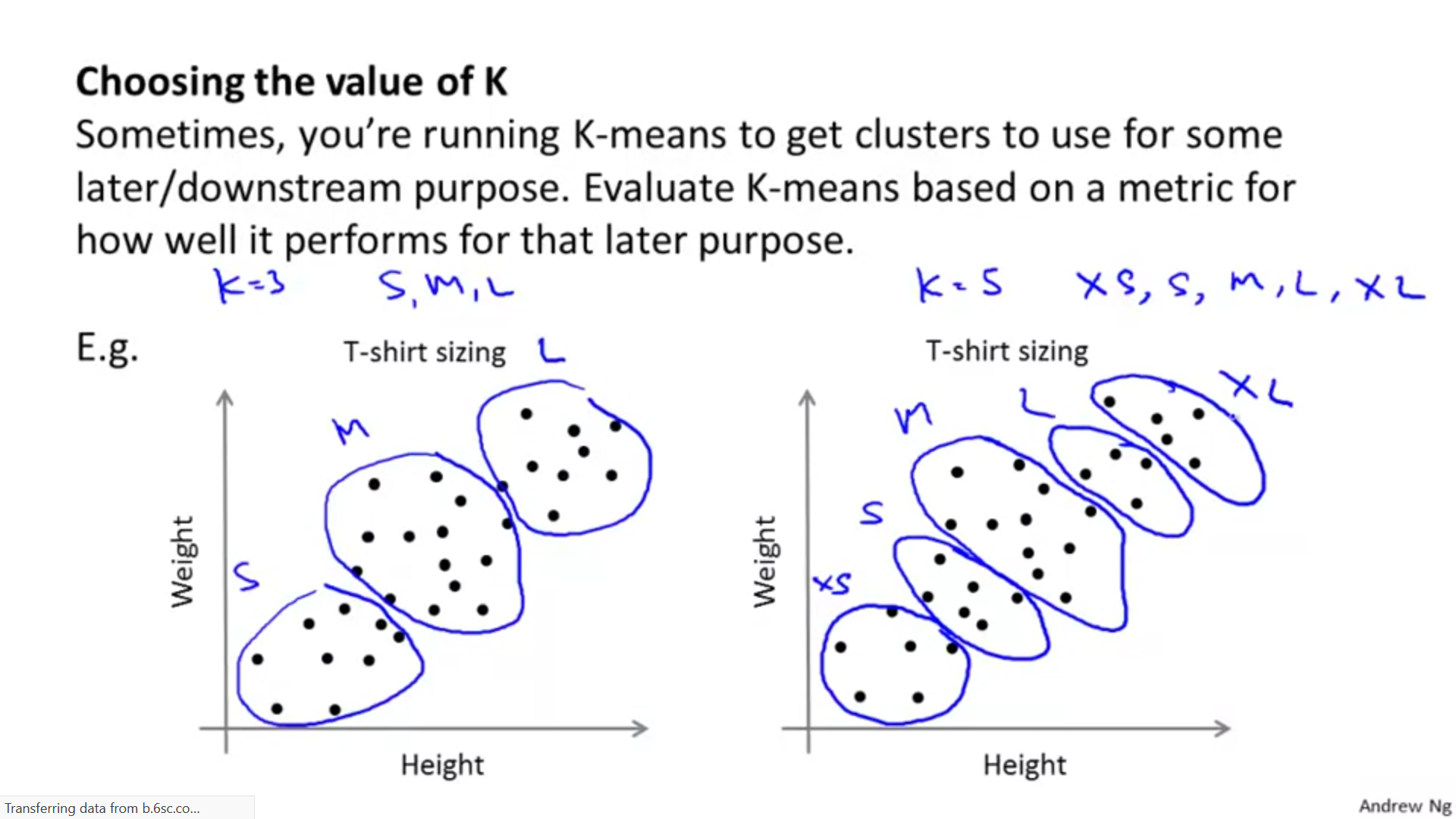
*Way* 1*:*

*Way* 2:

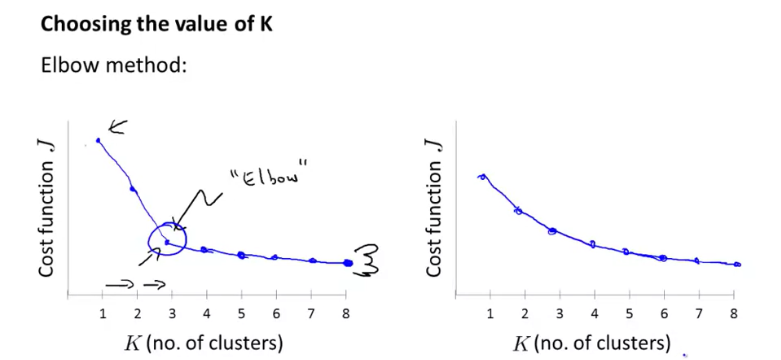


So, we see that there is no particular correct way of choosing a value of K and usually we choose a value by hand. Though there are two ways that may be used to choose a value of K:

1. Choosing K value according to need.

Now, if you are a T-shirt seller then you can choose a value of K according to your need. For example if you feel that using K=3 which classifies the t-shirts into 3 groups that is S,M or L is more useful than using K=5 which classifies the t-shirts into 5 groups that is XS,S,M,L or XL then you can choose K=3.

2. Using the elbow method:



If we plot cost function vs K and get the graph on the left side of above figure then the elbow point can be taken as value of K.

However usually we get the graph shown on the right side of the figure and there is no clear elbow. Hence this method is not always useful.

*Questions:*

