K Means Clustering is an unsupervised learning algorithm that tries to cluster the data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. In k means clustering, we have to specify the number of clusters we want the data to be grouped into. The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster. Then, the algorithm iterates through two steps:

1. Reassign data points to the cluster whose centroid is closest.
2. Calculate new centroid of each cluster.

These two steps are repeated until the within cluster variation cannot be reduced any further. The within cluster variation is calculated as the sum of the Euclidean distance between the data points and their respective cluster centroids.

This is part 1 for K means model in which we already know about the number of cluster in the part 2 number of cluster unknown.

Exploring the data:

The iris dataset contains data about sepal length, sepal width, petal length, and petal width of flowers of different species. Let us see what it looks like:

library(datasets)

head(iris)

After a little bit of exploration, I found that Petal.Length and Petal.Width were similar among the same species but varied considerably between different species, as demonstrated below:

Install.package(“ggplot2”)

Library(ggplot2)

ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom\_point()

Clustering:

Okay, now that we have seen the data, let us try to cluster it. Since the initial cluster assignments are random, let us set the seed to ensure reproducibility.

Set.seed(20)

irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)

irisCluster

Since we know that there are 3 species involved, we ask the algorithm to group the data into 3 clusters, and since the starting assignments are random, we specify nstart = 20. This means that R will try 20 different random starting assignments and then select the one with the lowest within cluster variation.  
We can see the cluster centroids, the clusters that each data point was assigned to, and the within cluster variation.

Let us compare the clusters with the species.

table(irisCluster$cluster, iris$Species)

As we can see, the data belonging to the setosa species got grouped into cluster 3, versicolor into cluster 2, and virginica into cluster 1. The algorithm wrongly classified two data points belonging to versicolor and six data points belonging to virginica.

We can also plot the data to see the clusters:

irisCluster$cluster <- as.factor(irisCluster$cluster)

ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point()

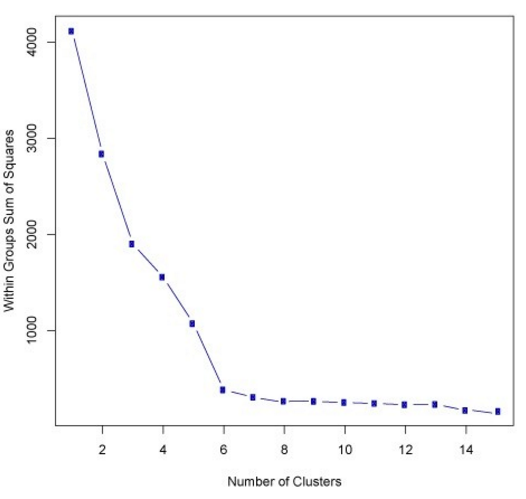
To see the error in clustering is:

ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point(size = 3, alpha = 0.5, aes(shape = iris$noerror))

You can choose the number of clusters by visually inspecting your data points, but you will soon realize that there is a lot of ambiguity in this process for all except the simplest data sets. This is not always bad, because you are doing unsupervised learning and there's some inherent subjectivity in the labeling process. Here, having previous experience with that particular problem or something similar will help you choose the right value.  
  
If you want some hint about the number of clusters that you should use, you can apply the *Elbow method*:

First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically:  
  


If you plot k against the SSE, you will see that *the error decreases as k gets larger*; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly. This produces an "elbow effect" in the graph, as you can see in the following picture:



In this case, k=6k=6 is the value that the Elbow method has selected. Take into account that the Elbow method is a heuristic and, as such, it may or may not work well in your particular case. Sometimes, there are more than one elbow, or no elbow at all. In those situations, you usually end up calculating the best k by evaluating how well k-means performs in the context of the particular clustering problem you are trying to solve.