



# Practical Considerations for K-Means Clustering Models

In this module, we look at some practical considerations for implementing k-means.

## Tuning K-Means Clustering Models

- The most important parameter for k-means clustering algorithm is the number of clusters  $k$ .
- This value is normally set by the analyst who has a good understanding of the business needs for the clustering algorithm.
- Recall that too few clusters may not properly separate observations (i.e. less homogenous clusters) while too many clusters may result in clusters that are meaningless or similar (i.e. small differences between clusters).
- In addition to the domain knowledge, there are also data driven methods that can help deciding the number of

The most important parameter to set here is  $k$ , the number of clusters. 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 different values for  $k$  and compare the resulting clusters. After choosing  $k$ , the  $n$  records are partitioned into these initial clusters. If there is external reasoning that suggests a certain partitioning, this information should be used. Alternatively, if there exists external information on the centroids of the  $k$  clusters, this can be used to initially allocate the records. Here, we will look at some data-driven methods to select  $k$ .

## Elbow Method

- Recall that the purpose of clustering is to find natural grouping of data points in data set such that points in same cluster are more similar to each other than points in different cluster.
- Therefore, the basic idea behind cluster partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation (known as total within-cluster sum of square) is minimized.
- The total within-cluster sum of square (WSS) measures the compactness of the clustering and we want it to be as small as possible.

Recall the idea behind clustering. This is so that we group similar items into a cluster, with each cluster being different from other clusters. In other words, we are grouping items so that within cluster variation among items in a cluster is small compared to the variation between clusters. This within cluster variation is called the within-cluster sum of squares (WSS). The smaller WSS is, the less the difference between items in a cluster. This provides a metric to determine  $k$ .

## Elbow Method Steps

1. Compute clustering algorithm (e.g., k-means clustering) for different values of  $k$ . For instance, by varying  $k$  from 1 to 10 clusters
  2. For each  $k$ , calculate the total within-cluster sum of square (WSS)
  3. Plot the curve of WSS according to the number of clusters  $k$ .
  4. The location of a bend (knee) in the plot (if there is any) is generally considered as an indicator of the appropriate number of clusters.
- `fviz_nbclust()` can be used to plot of WSS against the number of clusters,  $k$ .



An “elbow chart” is a line chart depicting the decline in cluster heterogeneity as we add more clusters. This simple chart allows us to graphically determine the best  $k$  for our data. Let us now apply this to our data.

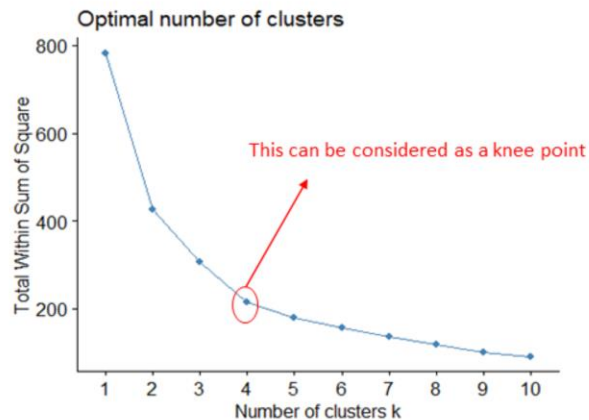
## Example: Elbow Method I

```
library(tidyverse) # data manipulation
library(factoextra) # clustering & visualization
library(ISLR)
set.seed(123)

df<-Auto[,c(1,6)]
# Scaling the data frame (z-score)
df <- scale(df)
fviz_nbclust(df, kmeans, method = "wss")
```

This example shows how to implement the elbow method.

## Example: Elbow Method II



The chart shows that the elbow point 4 provides the best value for  $k$ . While WSS will continue to drop for larger values of  $k$ , we have to make the tradeoff between overfitting, i.e., a model fitting both noise and signal, to a model having bias. Here, the elbow point provides that compromise where WSS, while still decreasing beyond  $k = 4$ , decreases at a much smaller rate. In other words,  $k=4$  provides the best value between bias and overfitting.

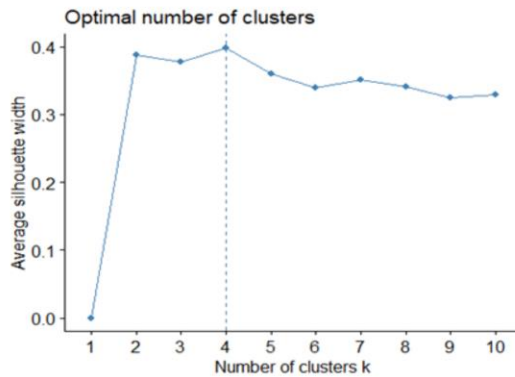
## Average Silhouette Method

- Silhouette refers to a method of interpretation and validation of consistency within clusters of data.
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- The silhouette ranges from  $-1$  to  $+1$ , where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate.
- Similar to the elbow method, `fviz_nbclust()` can be used to find the best  $k$  best on Average Silhouette Method

To ensure that the assignment of records to cluster is valid and consistent, we can apply the Silhouette Method. The idea is simple. The valid ranges for this method are between  $-1$  and  $+1$ , and it measures how similar an object is to its assigned cluster compared to the other clusters. High values indicate a good matching of data to clusters.

## Example: Clustering of Cars

```
fviz_nbclust(df, kmeans, method = "silhouette")
```



Again, we see that 4 is the ideal number of clusters. Here we look for large values for the Silhouette Width (Y Axis). This concludes our discussion on k-means.