# DATA420_KMEANS

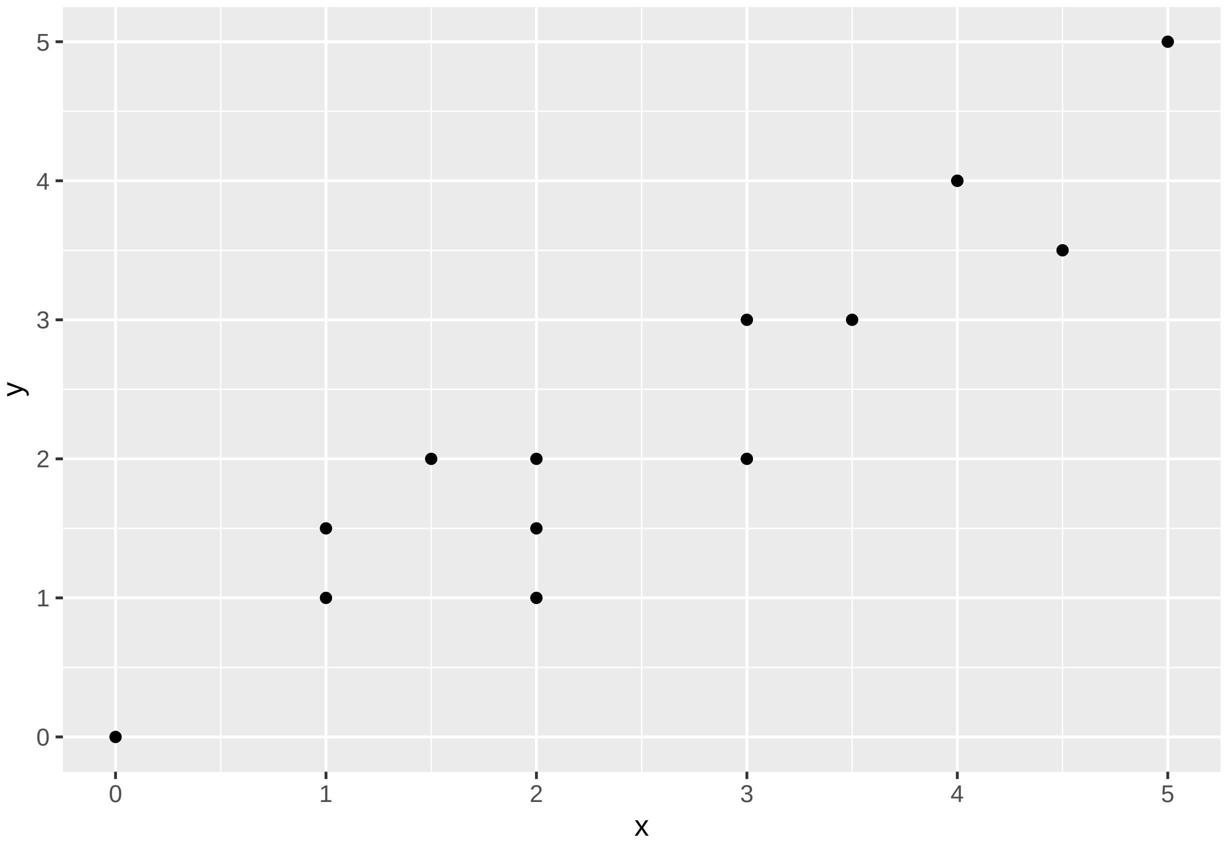
Jason Pemberton

2023-11-19

K Means Clustering is an Unsupervised Non-linear algorithm that clusters data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster. It is used in a variety of fields like banking, healthcare, retail, media, etc.

For this example we will build our own dataframe. When we visualize the data using a scatterplot we might be able to observe a number of clusters but sometimes it might not be obvious to the human eye.

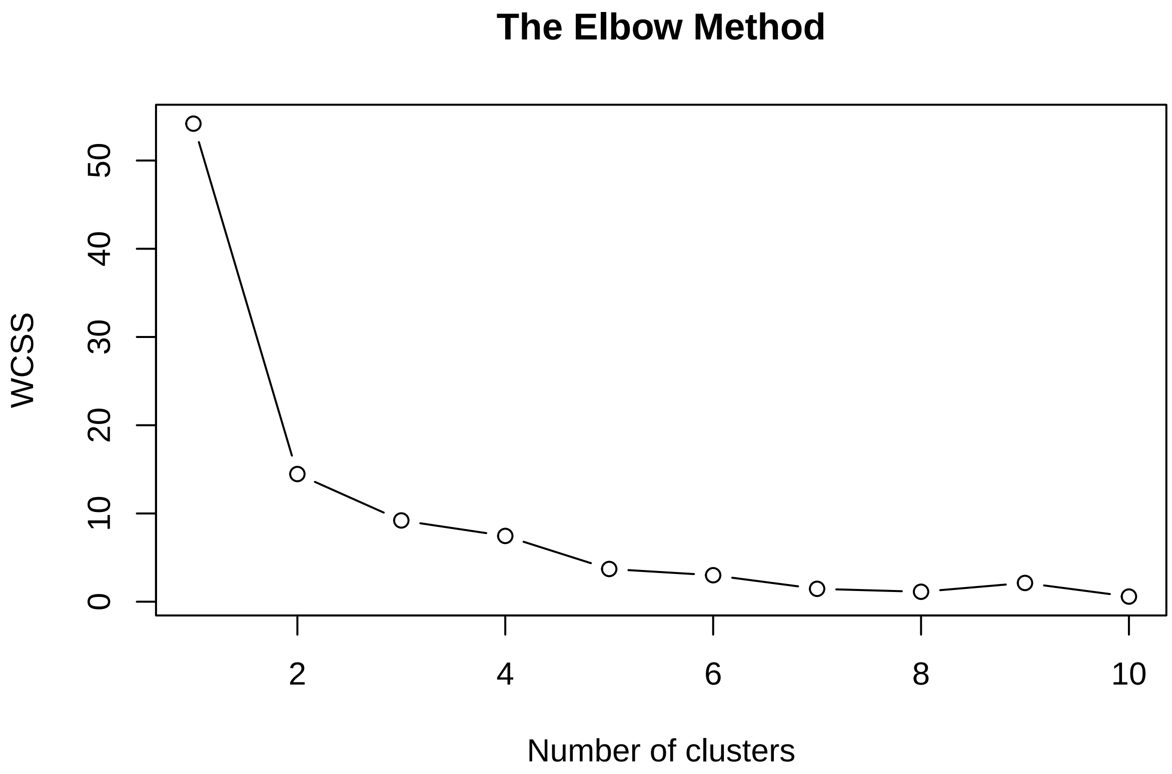
```
# Create a dataframe that contains two columns of numeric data
data <- data.frame(
  x = c(1,1.5,2,3,4,4.5,5,3,2,1,0,4,3.5,2),
  y = c(1,2,1.5,3,4,3.5,5,2,1,1.5,0,4,3,2))

# plot the data
ggplot(data, aes(x,y))+
  geom_point()
```



The kmeans function takes two arguments: (dataframe, k). We do not know how many clusters might be present in our data or which number (k) is optimal.

Using the elbow method, we can calculate the optimal number of clusters (k). For each value of K, we calculate WCSS (Within-Cluster Sum of Square). WCSS is the sum of the squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph, we can see that the graph will rapidly change at a point and thus creating an elbow shape.

```r
set.seed(123)
# make an empty vector we'll populate via our loop
wcss = vector()
# Use a for loop to run ten kmeans models and plot the k value versus wcss
for (i in 1:10) wcss[i] <- sum(kmeans(data, i)$withinss)
plot(1:10,
     wcss,
     type = 'b', # for lines and points
     main = paste('The Elbow Method'),
     xlab = 'Number of clusters',
     ylab = 'WCSS')
```

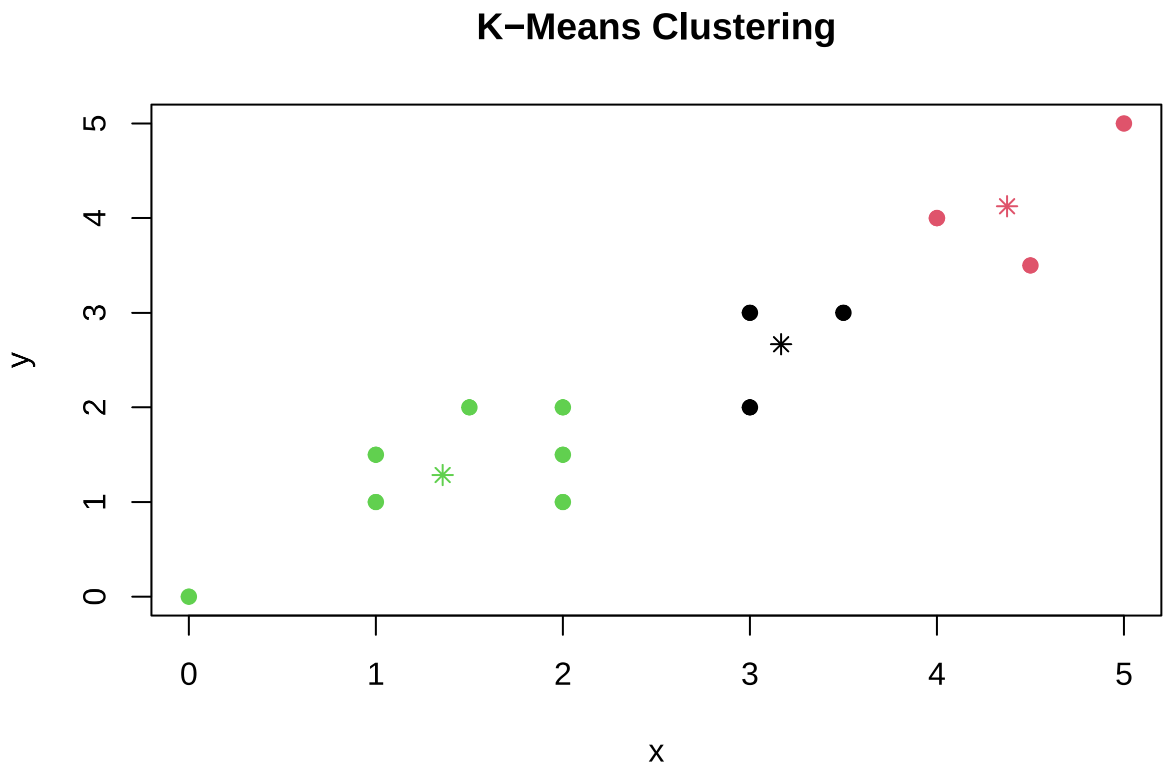


kmeans does not move your data, rather it calculates centroids (centres) based on the value of k. With each iteration of the model kmeans moves the centroids until it has successfully assigned each data point to its closest centroid. Now that we know the optimal k value from the elbow method we can run a final kmeans model and visualize the clusters

```r
# Define number of clusters, k
k <- 3

# Create a model using kmeans function. By assigning the model to a variable we can call on the model p

kmeans_model <- kmeans(data, centers=k)

# Display model results
print(kmeans_model$cluster)
```

```
##  [1] 3 3 3 1 2 2 2 1 3 3 3 2 1 3
```

```r
# Create a scatterplot of your data, colour the points by their assigned cluster from the kmeans model
plot(data, col=kmeans_model$cluster, pch=19, main="K-Means Clustering")
# Add centroids
points(kmeans_model$centers, col=1:k, pch = 8, cex=1)
```


K−Means Clustering
y
x