Machine Learning

K-Means Clustering

Ayşe Ceren Çiçek

K-means clustering aims to partition data into k clusters in a way that data points in the same cluster are similar and data points in the different clusters are farther apart. It's an unsupervised machine learning algorithm. It computes the centroids and iterates until it finds optimal centroid. It assumes that the number of clusters are already known.

The k-means clustering works as follows:

- Choose the k number of clusters
- Select k random points, the centroids (they don't have to be part of the dataset)
- Assign each point to the closest centroid
- Compute and replace the new centroid of each cluster
- Reassign each data point to the new closest centroid. If any reassignment happens, go back to previous step.

Random Initialization Trap

Random initialization trap may sometimes prevent us from developing the correct clusters. Depending on the position the centroids are randomly initialized, we can get very different clusters.

Determining Optimal Number of Clusters

The number of clusters that we choose for a given dataset cannot be random. Each cluster is formed by calculating and comparing the distances of data points within a cluster to its centroid. An ideal way to figure out the right number of clusters would be to calculate the Within-Cluster-Sum-of-Squares (WCSS).

WCSS is the sum of squares of the distances of each data point in all clusters to their respective centroids.

We can use some techniques to determine optimal number of clusters. **Elbow method** is one of them. he method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

Dataset: https://www.kaggle.com/shwetabh123/mall-customers

The columns are as follows:

- CustomerID: It is the unique ID given to a customer
- Gender: Gender of the customer
- Age: The age of the customer

- Annual Income(k$): It is the annual income of the customer
- Spending Score: It is the score(out of 100) given to a customer by the mall authorities, based on the money spent and the behavior of the customer.

Importing libraries

```
library(cluster)
```

Loading dataset

```
dataset = read.csv('dataset.csv')
head(dataset, n=5)
```

```
##   CustomerID  Genre Age Annual.Income..k.. Spending.Score..1.100.
## 1          1   Male  19                 15                     39
## 2          2   Male  21                 15                     81
## 3          3 Female  20                 16                      6
## 4          4 Female  23                 16                     77
## 5          5 Female  31                 17                     40
```

We will use annual income and spending score to cluster customers.

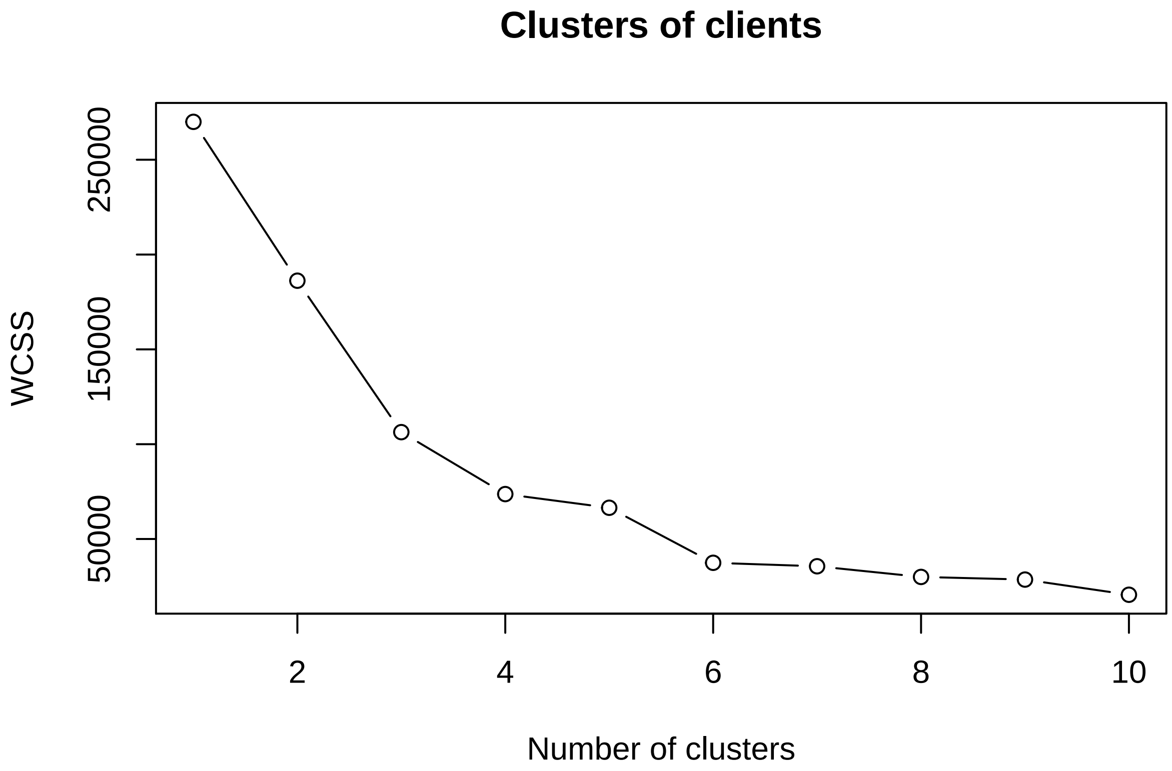
```
X <- dataset[4:5]
head(X, n=5)
```

```
##   Annual.Income..k.. Spending.Score..1.100.
## 1                 15                     39
## 2                 15                     81
## 3                 16                      6
## 4                 16                     77
## 5                 17                     40
```

Elbow method

We are going to use the Elbow Method to decide the optimal number of clusters.

```
set.seed(6)
wcss <- vector()
for (i in 1:10) wcss[i] <-  sum(kmeans(X, i)$withinss)
plot(1:10, wcss, type = "b", main = paste("Clusters of clients"), xlab = "Number of clusters", ylab = "
```



As seen on the plot, the optimal number of clusters seems as 5.

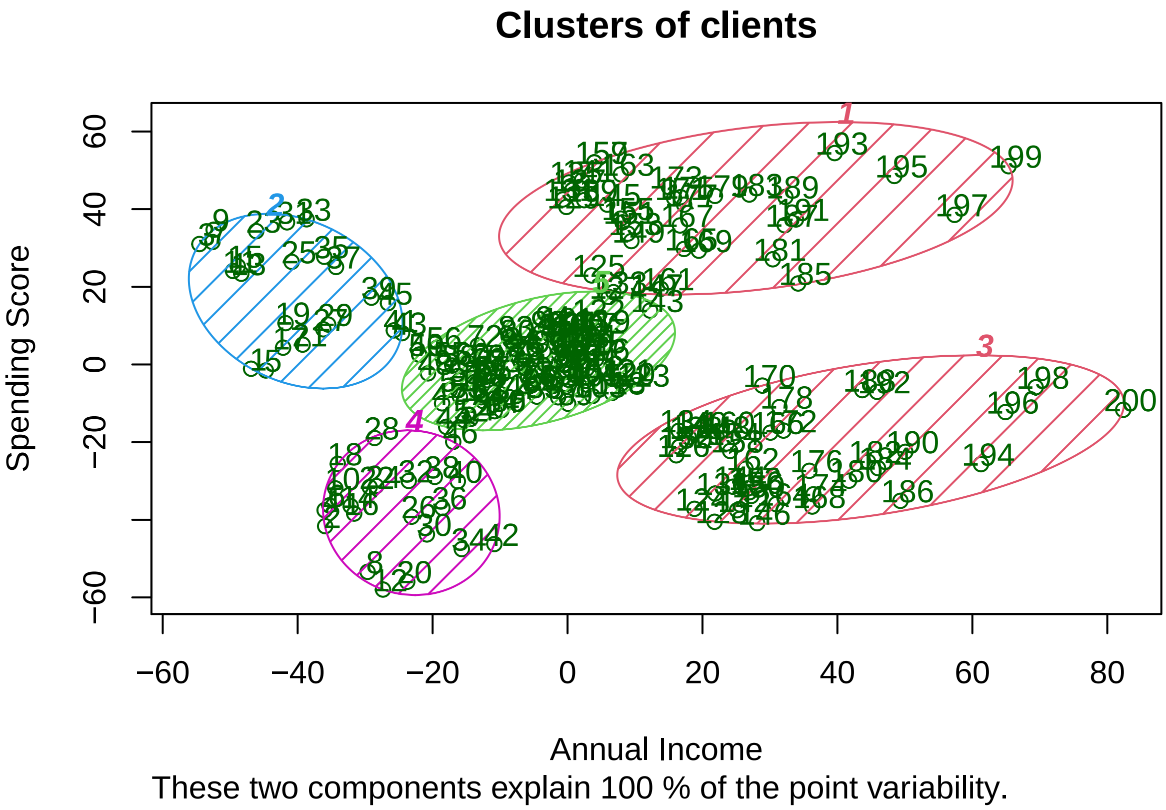
Apply K-Means

We will split our data into 5 clusters. The nstart parameter attempts multiple initial configurations and reports on the best one.

```
set.seed(29)
kmeans <- kmeans(X, 5, iter.max = 300, nstart = 10)
```

With clusplot function we can draw a 2 dimensional clustering plot with our clusters.

```
clusplot(X, clus = kmeans$cluster, lines = 0, shade = TRUE, color = TRUE, labels = 2, plotchar = FALSE,
         main = paste("Clusters of clients"), xlab = "Annual Income", ylab = "Spending Score")
```


Clusters of clients
Spending Score
60
40
20
0
−20
−60
Annual Income
−60
−40
−20
0
20
40
60
80
1
2
3
4
5
These two components explain 100 % of the point variability.