Cluster Analysis in Online Retail for Analyzing Consumer Buying Behaviour

Code ▾

Hide

```
library(readxl)
# Importing the dataset
dataset = read_xlsx('Online Retail.xlsx')
dataset=data.frame(dataset)
dim(dataset)
```

```
[1] 541909      8
```

Hide

```
str(dataset)
```

```
'data.frame':   541909 obs. of  8 variables:
 $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
 $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
 $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEA
RTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
 $ Quantity   : num  6 6 8 6 6 2 6 6 6 32 ...
 $ InvoiceDate: POSIXct, format: "2010-12-01 08:26:00" ...
 $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
 $ CustomerID : num  17850 17850 17850 17850 17850 ...
 $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

Hide

```
# Dropping Input variables: Repeated information and not useful
# Dropping: Invoice Number, Description & Unit Price (To avoid Redundant Information already in
 Stock Code), & InvoiceDate
dataset = dataset[,-c(1,3,5,6)]
dim(dataset)
```

```
[1] 541909      4
```

Hide

```
str(dataset)
```

```
'data.frame':   541909 obs. of  4 variables:
 $ StockCode : chr  "85123A" "71053" "84406B" "84029G" ...
 $ Quantity  : num  6 6 8 6 6 2 6 6 6 32 ...
 $ CustomerID: num  17850 17850 17850 17850 17850 ...
 $ Country   : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

Hide

```
# Checking for missing data
d3=dataset
for(i in 1:ncol(d3))
   {
    print(colnames(d3[i]))
    print(sum(is.na(d3[i])))
   }
```

```
[1] "StockCode"
[1] 0
[1] "Quantity"
[1] 0
[1] "CustomerID"
[1] 135080
[1] "Country"
[1] 0
```

Hide

```
# It seems that data for cutomer ID is missing only and is a huge fraction of the raw data, so I
 shall drop customer ID Column from our analysis
dataset = dataset[-3]
dim(dataset)
```

```
[1] 541909      3
```

Hide

```
str(dataset)
```

```
'data.frame':   541909 obs. of  3 variables:
 $ StockCode: chr  "85123A" "71053" "84406B" "84029G" ...
 $ Quantity : num  6 6 8 6 6 2 6 6 6 32 ...
 $ Country  : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

Hide

```
# Checking for missing data
d3=dataset
for(i in 1:ncol(d3))
   {
    print(colnames(d3[i]))
    print(sum(is.na(d3[i])))
   }
```

```
[1] "StockCode"
[1] 0
[1] "Quantity"
[1] 0
[1] "Country"
[1] 0
```

Hide

```
# There is no missing data
# Encoding the target feature as factor
dataset$StockCode = as.factor(dataset$StockCode)
# Training Set
training_set = dataset
# Feature Scaling
training_set[2] = scale(dataset[2])
str(training_set)
```

```
'data.frame':	541909 obs. of  3 variables:
 $ StockCode: Factor w/ 4070 levels "10002","10080",..: 3538 2795 3045 2986 2985 1663 801 1548 1
547 3306 ...
 $ Quantity : num [1:541909, 1] -0.01629 -0.01629 -0.00712 -0.01629 -0.01629 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr "Quantity"
  ..- attr(*, "scaled:center")= Named num 9.55
  .. ..- attr(*, "names")= chr "Quantity"
  ..- attr(*, "scaled:scale")= Named num 218
  .. ..- attr(*, "names")= chr "Quantity"
 $ Country  : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

Hide

```
# Defining the categorical and Numeric Input Data types
dataset$StockCode = as.factor(dataset$StockCode)
dataset$Quantity =  as.numeric(dataset$Quantity)
dataset$Country = as.character(dataset$Country)
str(dataset)
```

```
'data.frame':	541909 obs. of  3 variables:
 $ StockCode: Factor w/ 4070 levels "10002","10080",..: 3538 2795 3045 2986 2985 1663 801 1548 1
547 3306 ...
 $ Quantity : num  6 6 8 6 6 2 6 6 6 32 ...
 $ Country  : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

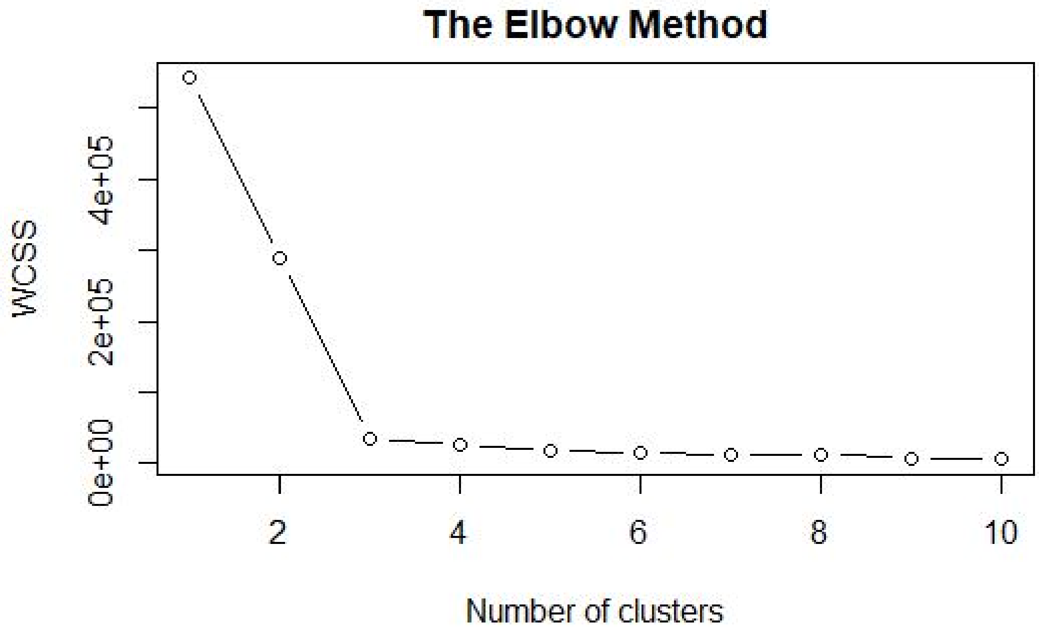
Hide

```
# Training Set
training_set = dataset
# Feature Scaling
training_set = scale(dataset[,2])
str(training_set)
```

```
 num [1:541909, 1] -0.01629 -0.01629 -0.00712 -0.01629 -0.01629 ...
 - attr(*, "scaled:center")= num 9.55
 - attr(*, "scaled:scale")= num 218
```

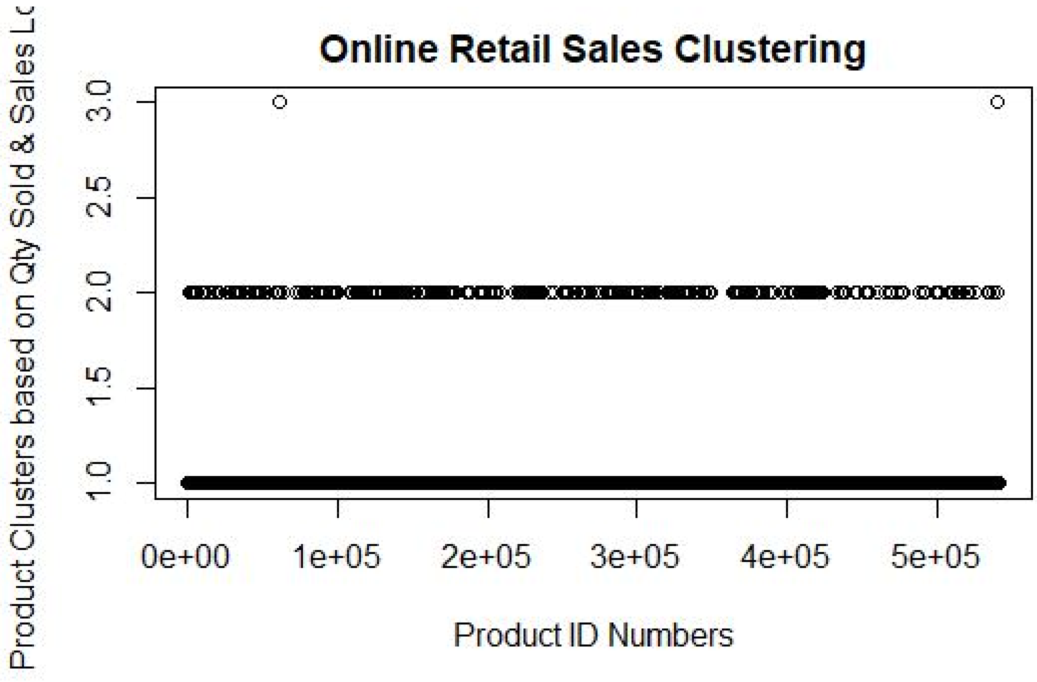
Hide

```
# Number of Clusters
k = 10
# Using the elbow method to find the optimal number of clusters
set.seed(123)
wcss = vector()
for (i in 1:k) wcss[i] = sum(kmeans(training_set, i, algorithm = "Hartigan-Wong")$withinss)
plot(1:k,
     wcss,
     type = 'b',
     main = paste('The Elbow Method'),
     xlab = 'Number of clusters',
     ylab = 'WCSS')
```



Hide

```
# Optimal Clusters Found
Optimal_Clusters = 3
# Fitting K-Means to the dataset
set.seed(456)
kmeans = kmeans(x = training_set, centers = Optimal_Clusters)
y_kmeans = kmeans$cluster
plot(y_kmeans,
     main = "Online Retail Sales Clustering",
     ylab = "Product Clusters based on Qty Sold & Sales Location",
     xlab = "Product ID Numbers")
```



Hide

```
# Histogram
print("Total Number of products in each cluster")
```

```
[1] "Total Number of products in each cluster"
```

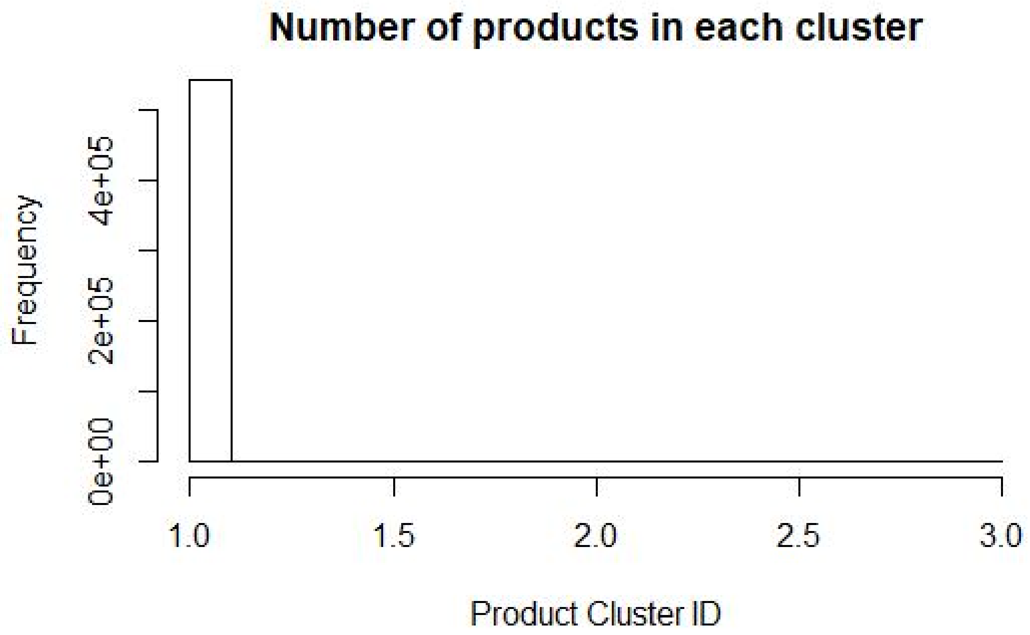
Hide

```
table(y_kmeans)
```

```
y_kmeans
     1      2      3 
541138    769      2 
```

Hide

```
hist(y_kmeans, main  = "Number of products in each cluster", xlab = "Product Cluster ID")
```


Number of products in each cluster
Frequency
0e+00
2e+05
4e+05
1.0
1.5
2.0
2.5
3.0
Product Cluster ID