Data Science Project – Customer Segmentation using Machine Learning in R

LANGAT ERICK

2023-08-20

Customer Segmentation Project in R

Customer Segmentation is one the most important applications of unsupervised learning. Using clustering techniques, companies can identify the several segments of customers allowing them to target the potential user base. In this machine learning project, we will make use of ***K-means clustering*** which is the essential algorithm for clustering unlabeled dataset. Before ahead in this project, learn what actually customer segmentation is.

What is Customer Segmentation?

Customer Segmentation is the process of division of customer base into several groups of individuals that share a similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits.

Companies that deploy customer segmentation are under the notion that every customer has different requirements and require a specific marketing effort to address them appropriately. Companies aim to gain a deeper approach of the customer they are targeting. Therefore, their aim has to be specific and should be tailored to address the requirements of each and every individual customer. Furthermore, through the data collected, companies can gain a deeper understanding of customer preferences as well as the requirements for discovering valuable segments that would reap them maximum profit. This way, they can strategize their marketing techniques more efficiently and minimize the possibility of risk to their investment.

The technique of customer segmentation is dependent on several key differentiators that divide customers into groups to be targeted. Data related to demographics, geography, economic status as well as behavioral patterns play a crucial role in determining the company direction towards addressing the various segments.

How to Implement Customer Segmentation in R?

In the first step of this data science project, we will perform data exploration. We will import the essential packages required for this role and then read our data. Finally, we will go through the input data to gain necessary insights about it.

```
#load libraries
suppressPackageStartupMessages(require(tidyverse))
library(bookdown)
```

```
#Import data
customer_data <- read.csv('C:/Users/langa/OneDrive/Desktop/Dataset/Mall_Customers.csv')
head(customer_data)
```

```
##   CustomerID Gender 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
## 6          6 Female  22                 17                     76
```

```
str(customer_data)
```

```
## 'data.frame':    200 obs. of  5 variables:
##  $ CustomerID            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender                : chr  "Male" "Male" "Female" "Female" ...
##  $ Age                   : int  19 21 20 23 31 22 35 23 64 30 ...
##  $ Annual.Income..k..    : int  15 15 16 16 17 17 18 18 19 19 ...
##  $ Spending.Score..1.100.: int  39 81 6 77 40 76 6 94 3 72 ...
```

```
names(customer_data)
```

```
## [1] "CustomerID"             "Gender"                 "Age"
## [4] "Annual.Income..k.."     "Spending.Score..1.100."
```

```
summary(customer_data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.75   36.00   38.85   49.00   70.00
```

```
sd(customer_data$Age)
```

```
## [1] 13.96901
```

```
summary(customer_data$Annual.Income..k..)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   41.50   61.50   60.56   78.00  137.00
```

```
sd(customer_data$Annual.Income..k..)
```

```
## [1] 26.26472
```

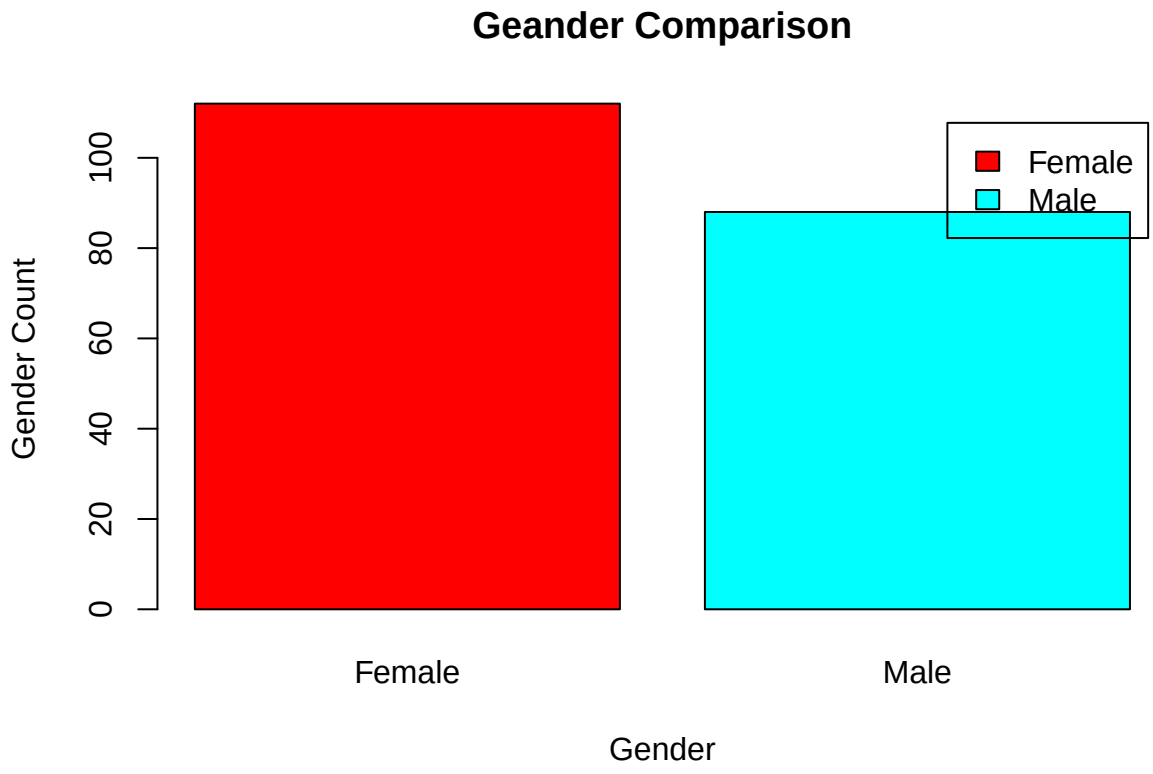
Customer Gender Visualization

In this, we will create a barplot and a piechart to show the gender distribution across our customer_data dataset.

```
(a <- table(customer_data$Gender))
```

```
##
## Female   Male
##    112     88
```

```
barplot(a,main = 'Geander Comparison',
        ylab = 'Gender Count',
        legend.text = rownames(a),
        xlab = 'Gender', col=rainbow(2))
```



From the above barplot, we observe that the number of females is higher than the males. Now, let us visualize a pie chart to observe the ratio of male and female distribution.

```
(pct <- round(a/sum(a)*100))
```

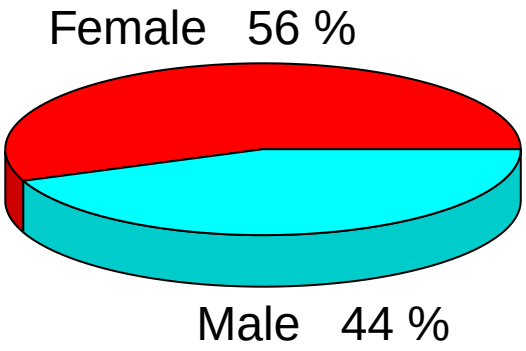
```
##
## Female   Male
##     56     44
```

```
(lbs <- paste(c('Female', 'Male'), ' ', pct, '%', sep=' '))
```

```
## [1] "Female   56 %" "Male   44 %"
```

```
suppressPackageStartupMessages( library(plotrix))
pie3D(a, labels = lbs, main=' Pie Chart')
```

Pie Chart



From the above graph, we conclude that the percentage of females is **56%**, whereas the percentage of male in the customer dataset is **44%**.

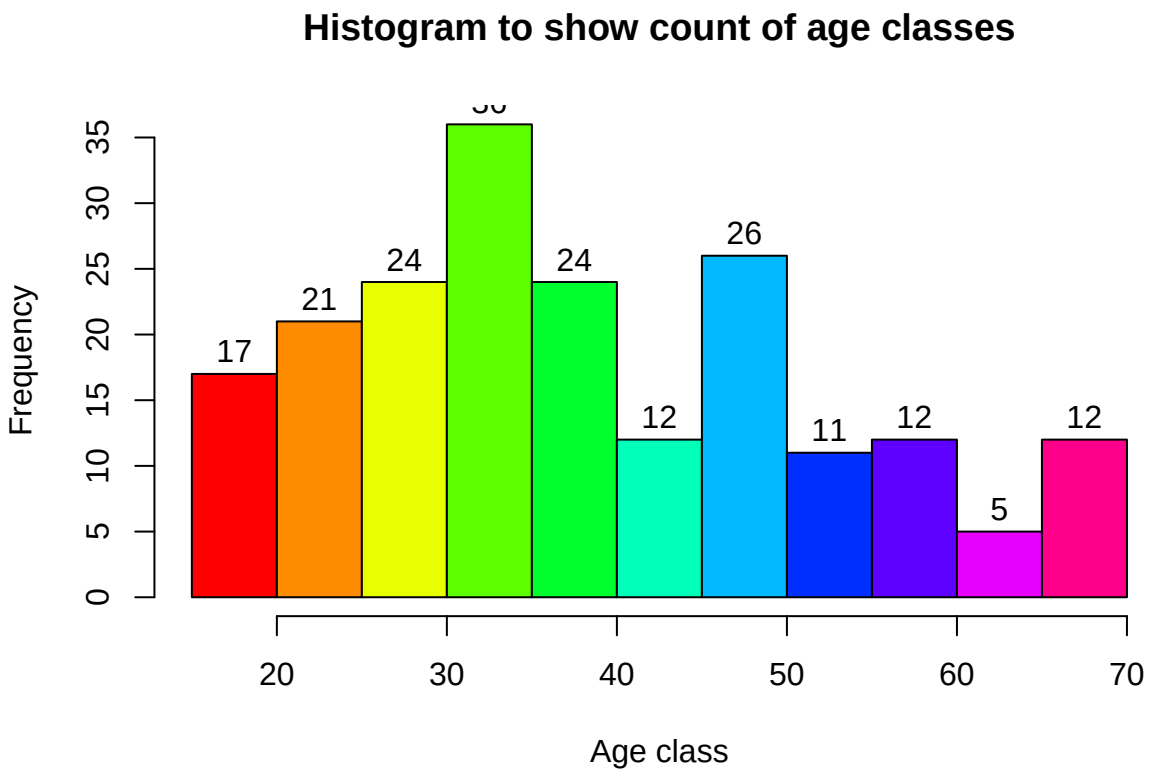
Visualization of Age Distribution

Let us plot a histogram to view the distribution to plot the frequency of customer ages. We will first proceed by taking summary of the Age variable.

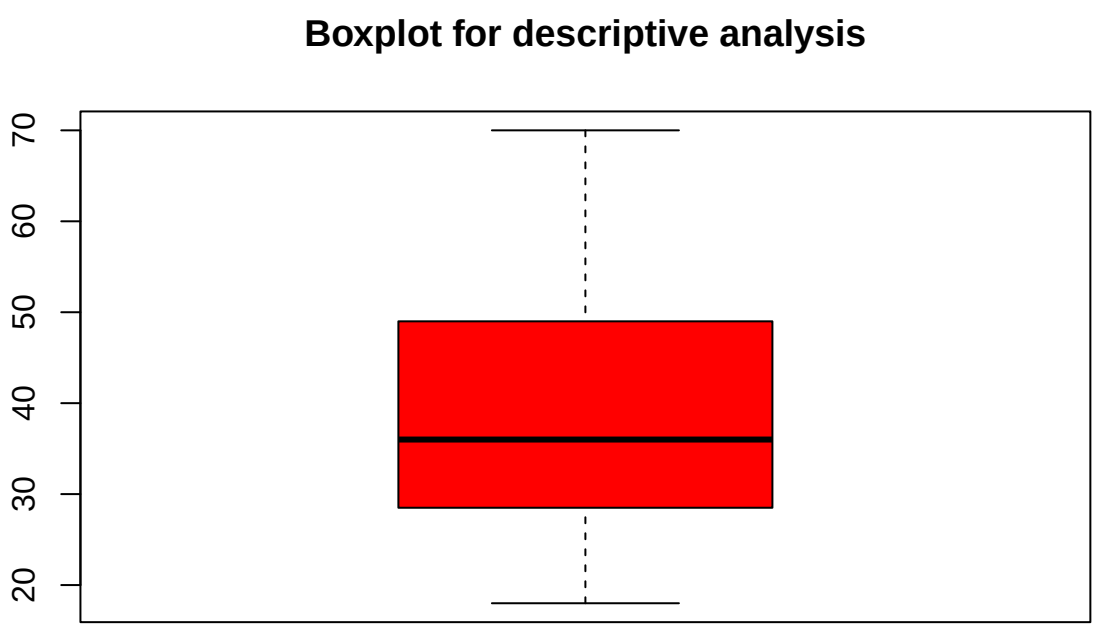
```
summary(customer_data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.75   36.00   38.85   49.00   70.00
```

```
hist(customer_data$Age, col=rainbow(11),
     main = 'Histogram to show count of age classes',
     labels = TRUE,
     xlab = 'Age class', ylab = 'Frequency')
```




```
boxplot(customer_data$Age,
        main='Boxplot for descriptive analysis',
        col='red')
```



From the above two visualizations, we conclude that the maximum customer ages are between 30 and 35. The minimum age of customers is 18, whereas, the maximum age is 70.

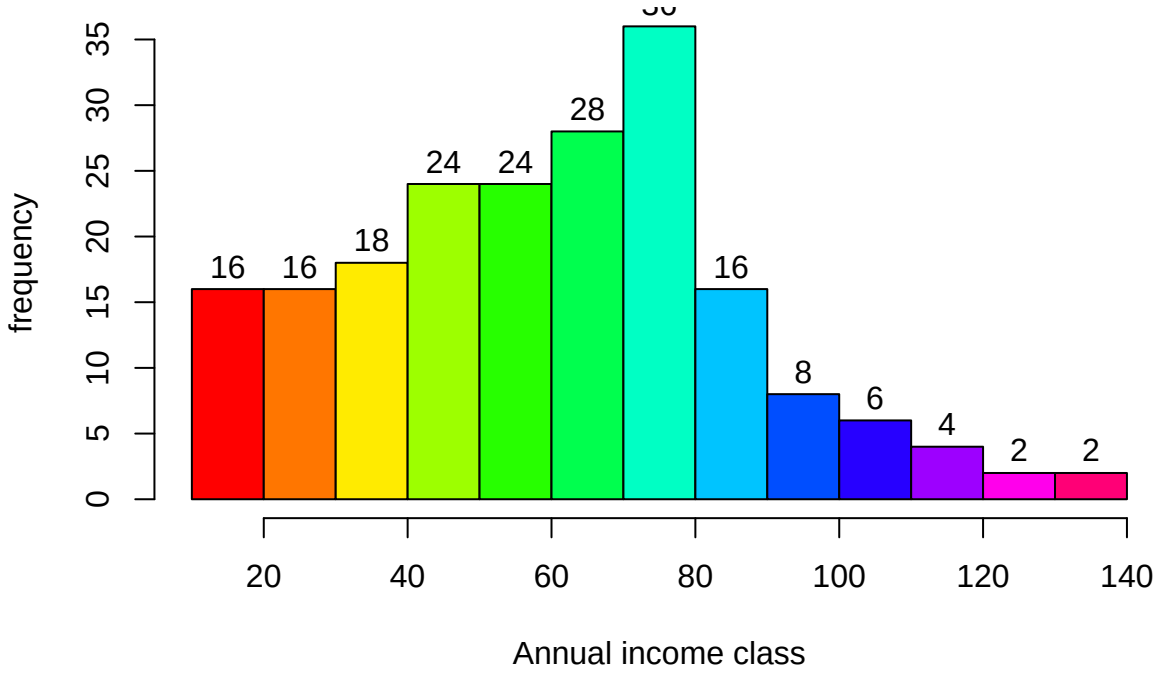
Analysis of the Annual Income of the Customers

In this section of the R project, we will create visualizations to analyze the annual income of the customers. We will plot a histogram and then we will proceed to examine this data using a density plot.

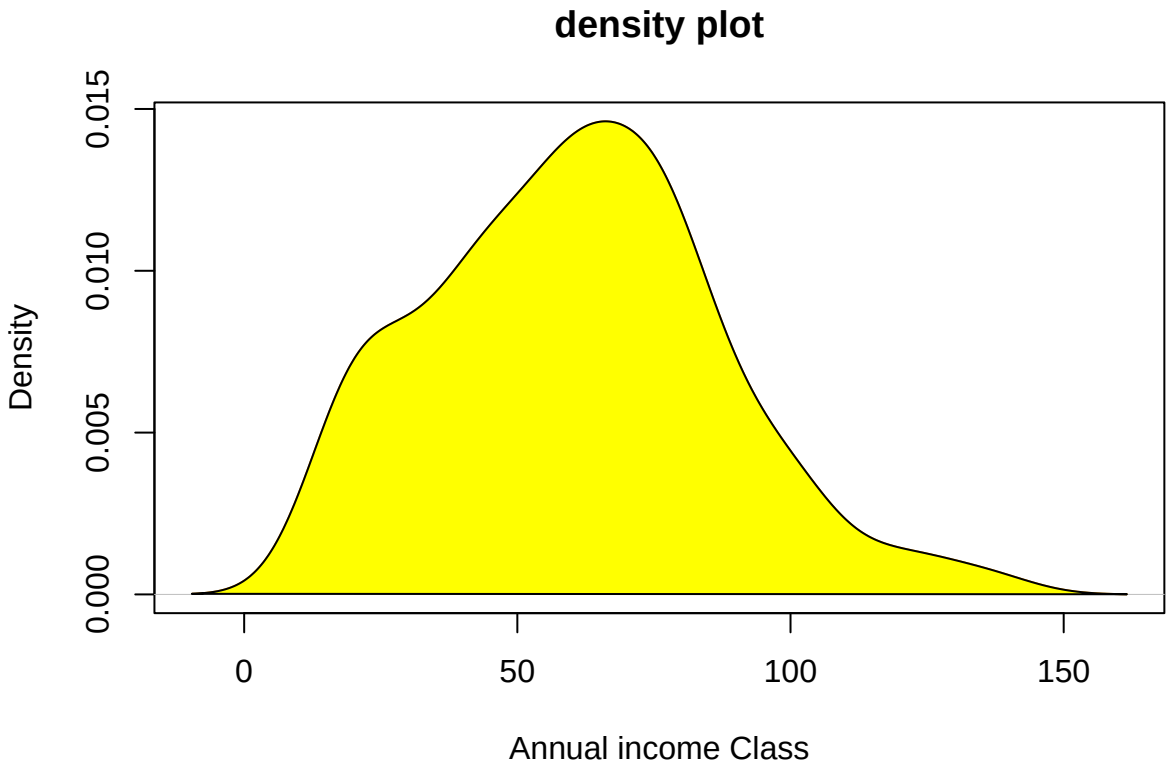
```
summary(customer_data$Annual.Income..k..)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   41.50   61.50   60.56   78.00  137.00
```

```
hist(customer_data$Annual.Income..k..,
     main = 'Histogram for Annual Income', col = rainbow(13),
     labels = TRUE,
     xlab = 'Annual income class',
     ylab = 'frequency')
```




```
plot(density(customer_data$Annual.Income..k..), col='orange',
             main='density plot',
             ylab='Density',
             xlab='Annual income Class')
polygon(density(customer_data$Annual.Income..k..), col = 'yellow')
```



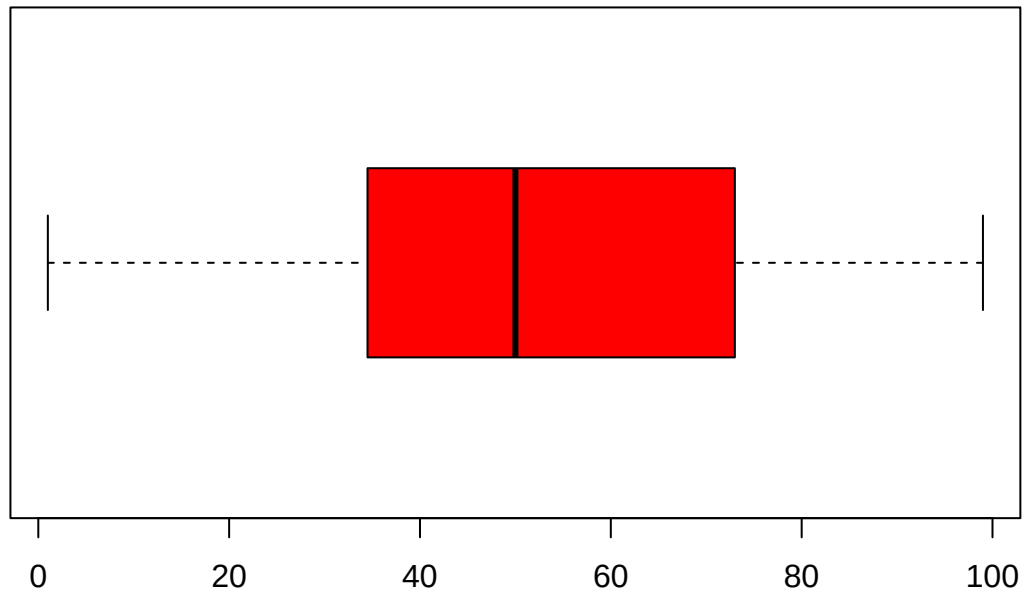
From the above descriptive analysis, we conclude that the minimum annual income of the customers is 15 and the maximum income is 137. People earning an average income of 70 have the highest frequency count in our histogram distribution. The average salary of all the customers is 60.56. In the Kernel Density Plot that we displayed above, we observe that the annual income has a ***normal distribution***.

Analyzing Spending Score of the Customers

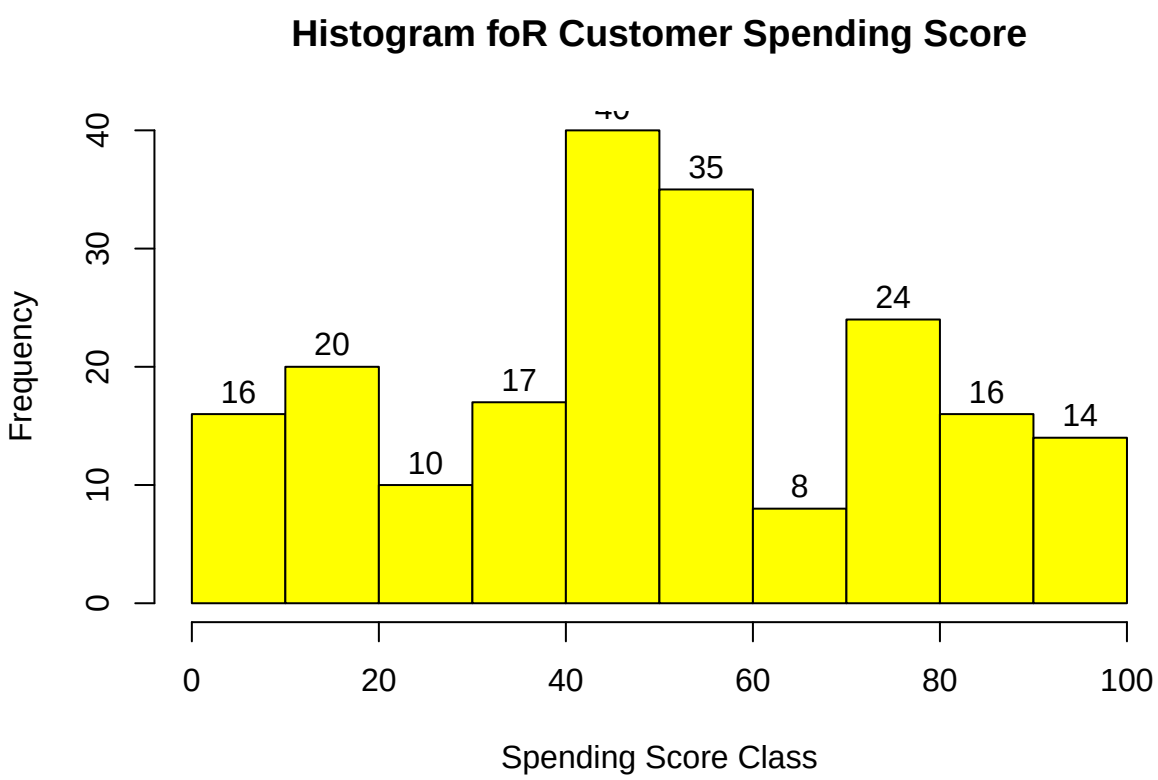
```
summary(customer_data$Spending.Score..1.100.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   34.75   50.00   50.20   73.00   99.00
```

```
boxplot(customer_data$Spending.Score..1.100.,
        col='red',
        main=' Boxplot for descriptive analysis of Spending Score',
        horizontal =T)
```




```r
hist(customer_data$Spending.Score..1.100.,
     col = "yellow",
     main = 'Histogram foR Customer Spending Score',
     xlab = 'Spending Score Class',
     ylab = 'Frequency',
     labels = TRUE)
```



The minimum spending score is 1, maximum is 99 and the average is 50.20. We can see Descriptive Analysis of Spending Score is that Min is 1, Max is 99 and avg. is 50.20. From the histogram, we conclude that customers between class 40 and 50 have the highest spending score among all the classes.

K-means Algorithm

While using the k-means clustering algorithm, the first step is to indicate the number of clusters (k) that we wish to produce in the final output. The algorithm starts by selecting k objects from dataset randomly that will serve as the initial centers for our clusters. These selected objects are the cluster means, also known as centroids. Then, the remaining objects have an assignment of the closest centroid. This centroid is defined by the Euclidean Distance present between the object and the cluster mean. We refer to this step as "cluster assignment". When the assignment is complete, the algorithm proceeds to calculate new mean value of each cluster present in the data. After the recalculation of the centers, the observations are checked if they are closer to a different cluster. Using the updated cluster mean, the objects undergo reassignment. This goes on repeatedly through several iterations until the cluster assignments stop altering. The clusters that are present in the current iteration are the same as the ones obtained in the previous iteration.

Determining Optimal Clusters

Gap Statistic Method In 2001, researchers at Stanford University – **R. Tibshirani, G.Walther and T. Hastie** published the Gap Statistic Method. We can use this method to any of the clustering method like K-means, hierarchical clustering etc. Using the gap statistic, one can compare the total intracluster variation for different values of k along with their expected values under the null reference distribution of data. With the help of **Monte Carlo simulations**, one can produce the sample dataset. For each variable

in the dataset, we can calculate the range between min(xi) and max (xj) through which we can produce values uniformly from interval lower bound to upper bound.

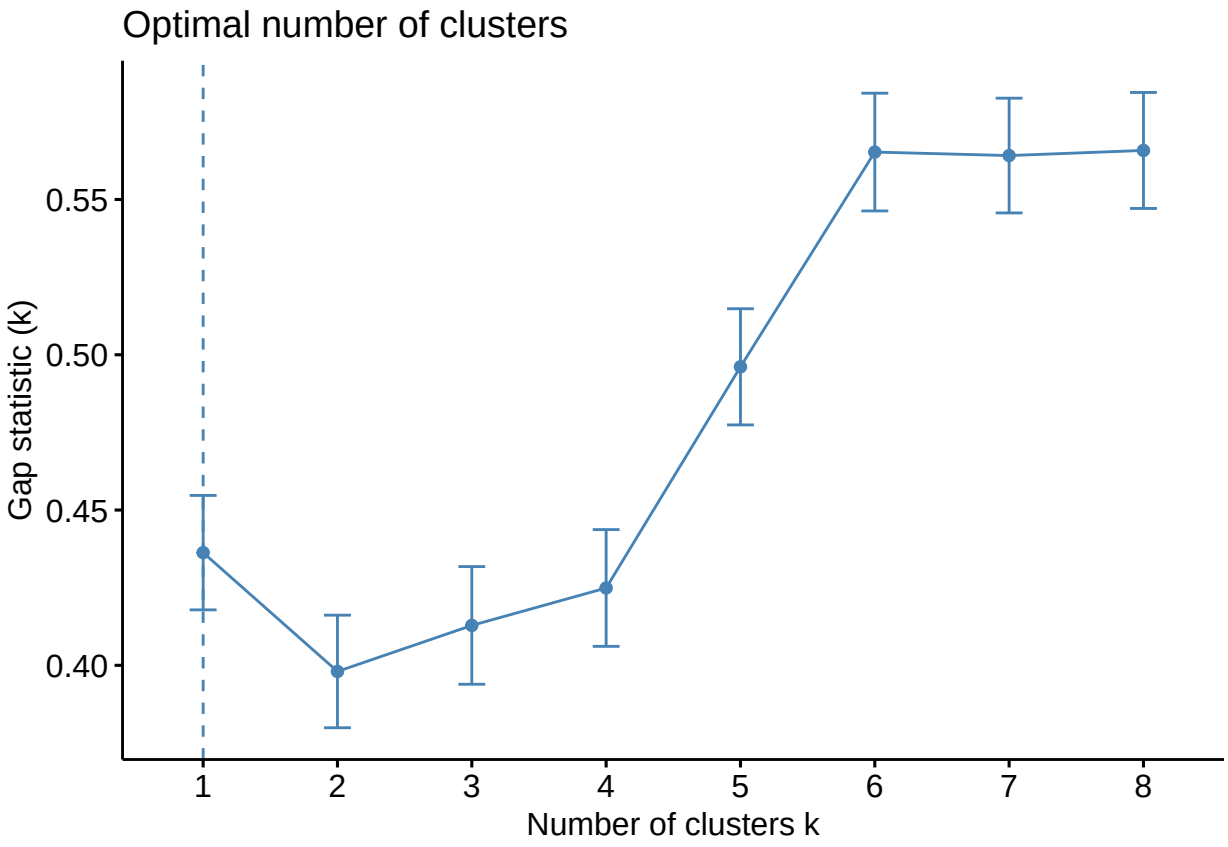
For computing the gap statistics method we can utilize the clusGap function for providing gap statistic as well as standard error for a given output.

```
library(NbClust)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(cluster)
set.seed(125)
# fviz_nbclust(customer_data[, 3:5], kmeans, method = 'silhouette')
sta_gap <- clusGap(customer_data[,3:5], FUN=kmeans, nstart=25, K.max = 8)

fviz_gap_stat(sta_gap)
```



Now, let us take k = 6 as our optimal cluster

```
k6 <- kmeans(customer_data[, 3:5],centers =  6, iter.max = 1000,
             nstart = 20, algorithm = 'Lloyd')
k6
```

```
## K-means clustering with 6 clusters of sizes 39, 38, 35, 44, 22, 22
```

```
##
## Cluster means:
##        Age Annual.Income..k.. Spending.Score..1.100.
## 1 32.69231           86.53846               82.12821
## 2 27.00000           56.65789               49.13158
## 3 41.68571           88.22857               17.28571
## 4 56.34091           53.70455               49.38636
## 5 25.27273           25.72727               79.36364
## 6 44.31818           25.77273               20.27273
##
## Clustering vector:
##   [1] 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6 5 6
##  [38] 5 6 5 4 5 6 2 6 5 4 2 2 2 4 2 2 4 4 4 4 4 2 4 4 2 4 4 4 2 4 4 2 2 4 4 4 4
##  [75] 4 2 4 2 2 4 4 2 4 4 2 4 4 2 2 4 4 2 4 2 2 2 4 2 4 2 2 4 4 2 4 2 4 4 4 4 4
## [112] 2 2 2 2 2 4 4 4 4 2 2 2 1 2 1 3 1 3 1 3 1 2 1 3 1 3 1 3 1 3 1 2 1 3 1 3 1
## [149] 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3
## [186] 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
##
## Within cluster sum of squares by cluster:
## [1] 13972.359  7742.895 16690.857  7607.477  4099.818  8189.000
##  (between_SS / total_SS =  81.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

In the output of our kmeans operation, we observe a list with several key information. From this, we conclude the useful information being –

- **cluster** – This is a vector of several integers that denote the cluster which has an allocation of each point.
- **totss** – This represents the total sum of squares.
- **centers** – Matrix comprising of several cluster centers
- **withinss** – This is a vector representing the intra-cluster sum of squares having one component per cluster.
- **tot.withinss** – This denotes the total intra-cluster sum of squares.
- **betweenss** – This is the sum of between-cluster squares.
- **size** – The total number of points that each cluster holds.

Visualizing the Clustering Results using the First Two Principle Components

```
pcclust <- prcomp(customer_data[, 3:5], scale. = FALSE)
summary(pcclust)
```

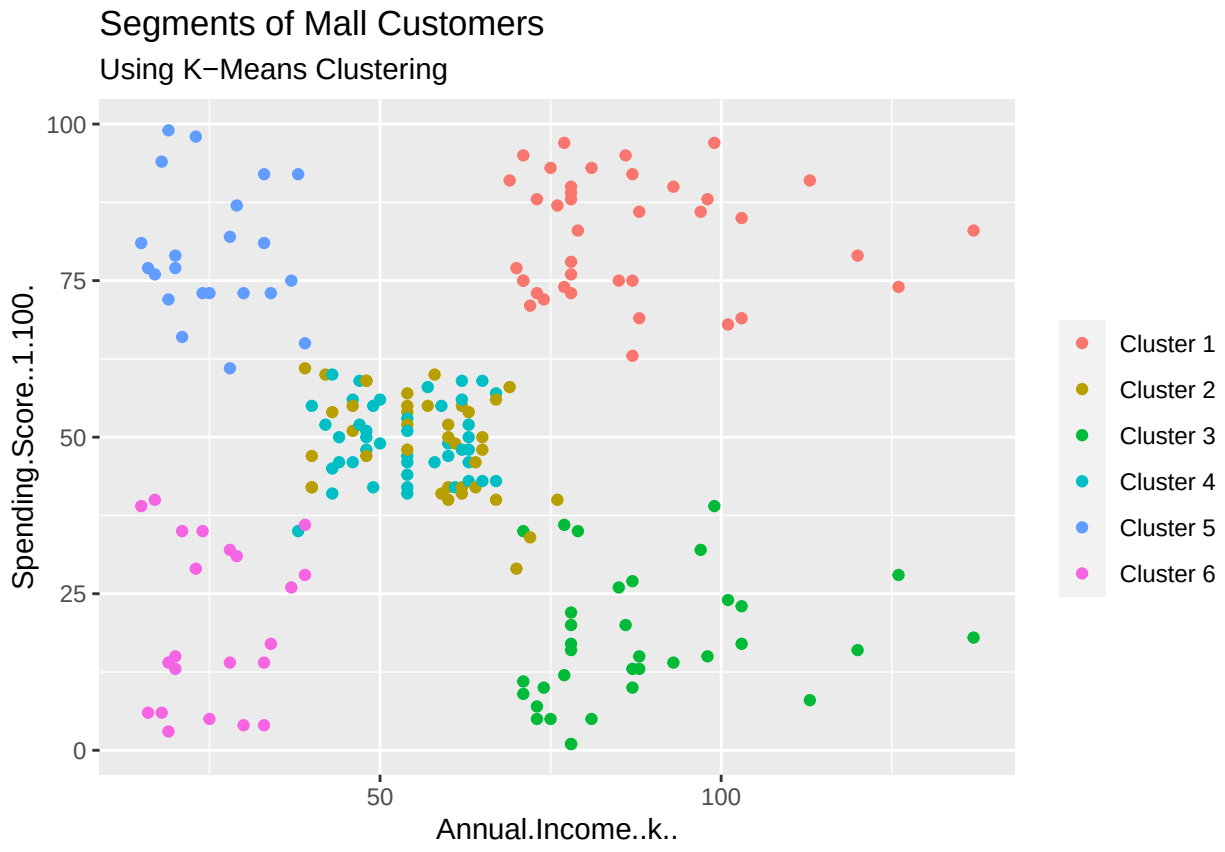
```
## Importance of components:
##                           PC1     PC2     PC3
```

```
## Standard deviation     26.4625 26.1597 12.9317
## Proportion of Variance  0.4512  0.4410  0.1078
## Cumulative Proportion   0.4512  0.8922  1.0000
```

```
pcclust$rotation[,1:2]
```

```
##                               PC1        PC2
## Age                     0.1889742 -0.1309652
## Annual.Income..k..     -0.5886410 -0.8083757
## Spending.Score..1.100. -0.7859965  0.5739136
```

```
# names(customer_data)
set.seed(1)
ggplot(customer_data, aes(Annual.Income..k.., Spending.Score..1.100.))+
     geom_point(stat = 'identity', aes(color=as.factor(k6$cluster))) +
      scale_color_discrete(name=" ",
                           breaks= c("1", "2", "3", "4", "5", "6"),
                           labels=c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5","Clu
      ggtitle("Segments of Mall Customers", subtitle = 'Using K-Means Clustering')
```



From the above visualization, we observe that there is a distribution of 6 clusters as follows –

Cluster 6 and 3 – These clusters represent the customer_data with the medium income salary as well as the medium annual spend of salary.

Cluster 5 – This cluster represents the customer_data having a high annual income as well as a high annual spend.

Cluster 2 – This cluster denotes the customer_data with low annual income as well as low yearly spend of income.

Cluster 4 – This cluster denotes a high annual income and low yearly spend.

Cluster 1 – This cluster represents a low annual income but its high yearly expenditure.

With the help of clustering, we can understand the variables much better, prompting us to take careful decisions. With the identification of customers, companies can release products and services that target customers based on several parameters like income, age, spending patterns, etc. Furthermore, more complex patterns like product reviews are taken into consideration for better segmentation.

Summary

In this data science project, we went through the customer segmentation model. We developed this using a class of machine learning known as unsupervised learning. Specifically, we made use of a clustering algorithm called K-means clustering. We analyzed and visualized the data and then proceeded to implement our algorithm. Hope you enjoyed this customer segmentation project of machine learning using R.