Customer Segmentation Report

Pius Mbeti

2023-07-09

Problem Statement
Many companies lack a comprehensive understanding of their diverse customer base, which hinders their ability to effectively address distinct customer needs, preferences, and behaviors. This leads to the utilization of generic marketing and customer engagement strategies, resulting in suboptimal customer satisfaction, low customer retention rates, and inefficient resource allocation. To address this industry-wide challenge, there is a need to develop effective customer segmentation frameworks that can identify and target specific customer groups with tailored marketing initiatives. By leveraging advanced statistical techniques, data analysis tools, and machine learning algorithms, companies can gain valuable insights to enhance customer satisfaction, drive customer loyalty, and foster sustainable business growth.

Loading the packages library(NbClust)
library(plotrix)
library(purrr)
library(gridExtra)
library(factoextra)
library(cluster)
library(grid)
library(ggplot2)
library(tidyr)
library(rmarkdown)

Reading the dataset
I sourced the dataset for this project from kaggle

```
customer_data <- read.csv("C:/Users/Pius/OneDrive/Desktop/Customer Segmentation/Mall_Customers.csv")

head(customer_data, n = 10)
```

```
##    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
## 7           7 Female  35                 18                      6
## 8           8 Female  23                 18                     94
## 9           9   Male  64                 19                      3
## 10         10 Female  30                 19                     72
```

```
names(customer_data)
```

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

```
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
```

```
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] 13.95149
```

```
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] 26.23179
```

```
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
```

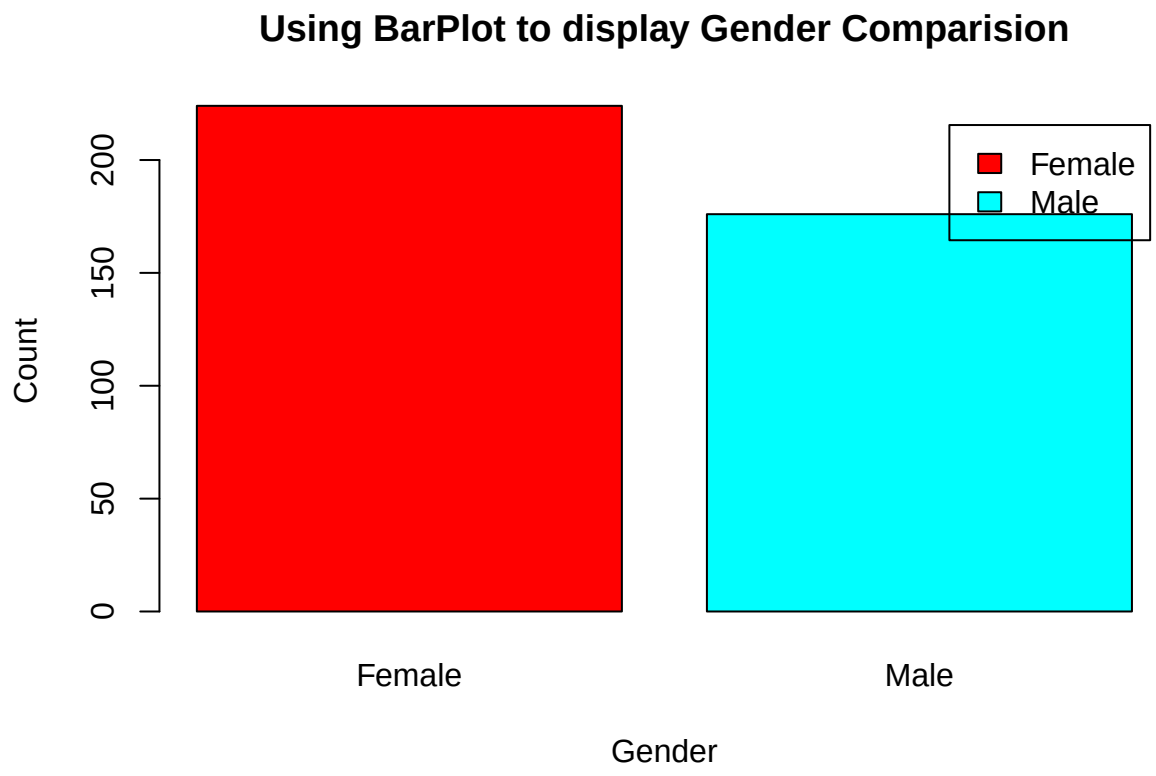
```
sd(customer_data$Spending.Score..1.100.)
```

```
## [1] 25.79114
```

Customer Gender Visualization

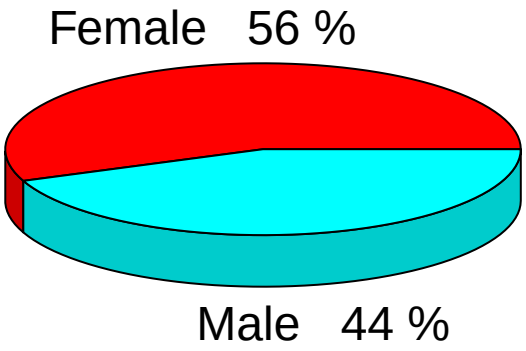
I carried out the task of creating visualizations to analyze the gender distribution in the customer_data dataset. To represent this information, I created a barplot and a pie chart. Upon analyzing the generated graphs, it is evident that the majority of the customers are females, accounting for 56% of the dataset. On the other hand, males make up 44% of the customers. These findings were derived from the visualizations below.

```r
#Customer Gender Visualization

a=table(customer_data$Gender)
barplot(a,main="Using BarPlot to display Gender Comparision",
        ylab="Count",
        xlab="Gender",
        col=rainbow(2),
        legend=rownames(a))
```




```r
pct=round(a/sum(a)*100)
lbs=paste(c("Female","Male")," ",pct,"%",sep=" ")
library(plotrix)
pie3D(a,labels=lbs,
      main="Pie Chart Depicting Ratio of Female and Male")
```

Pie Chart Depicting Ratio of Female and Male



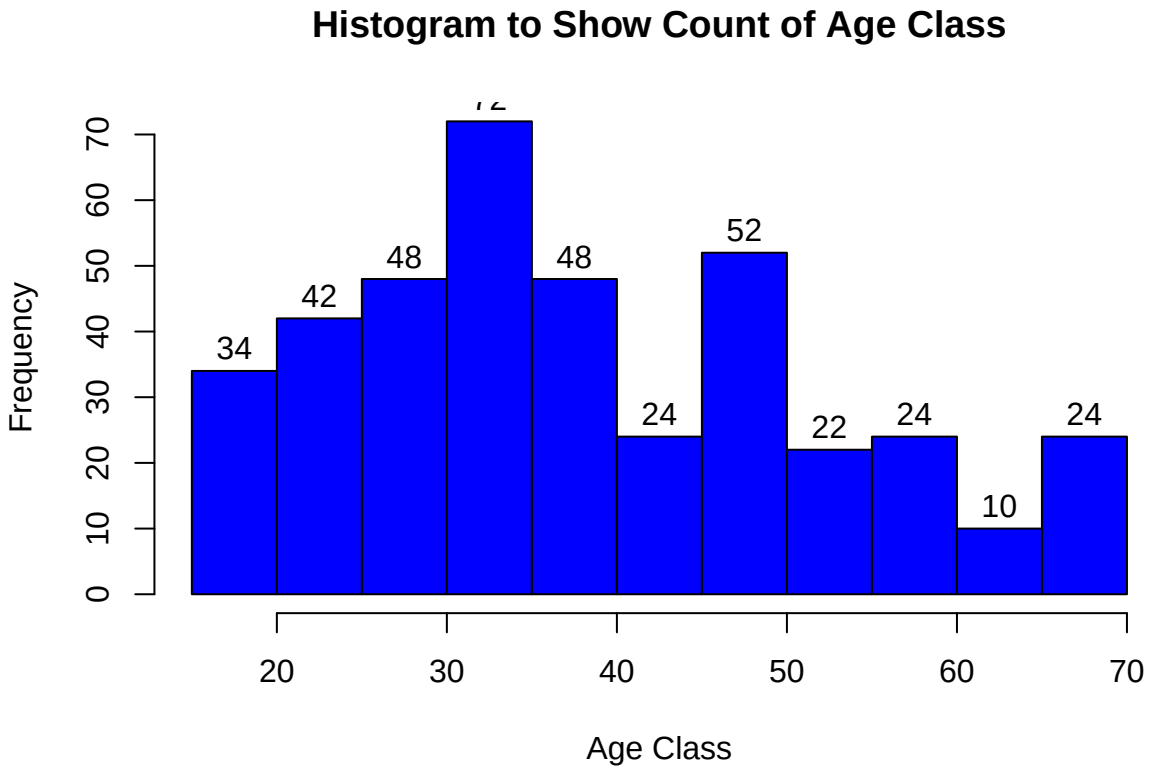
Age Distribution Visualization

For the distribution of customer ages, I used a histogram and a boxplot. The histogram shows the frequency of customer ages, with the highest frequency observed between 30 and 35. The boxplot provides additional insights into the age distribution, including the median and any outliers.

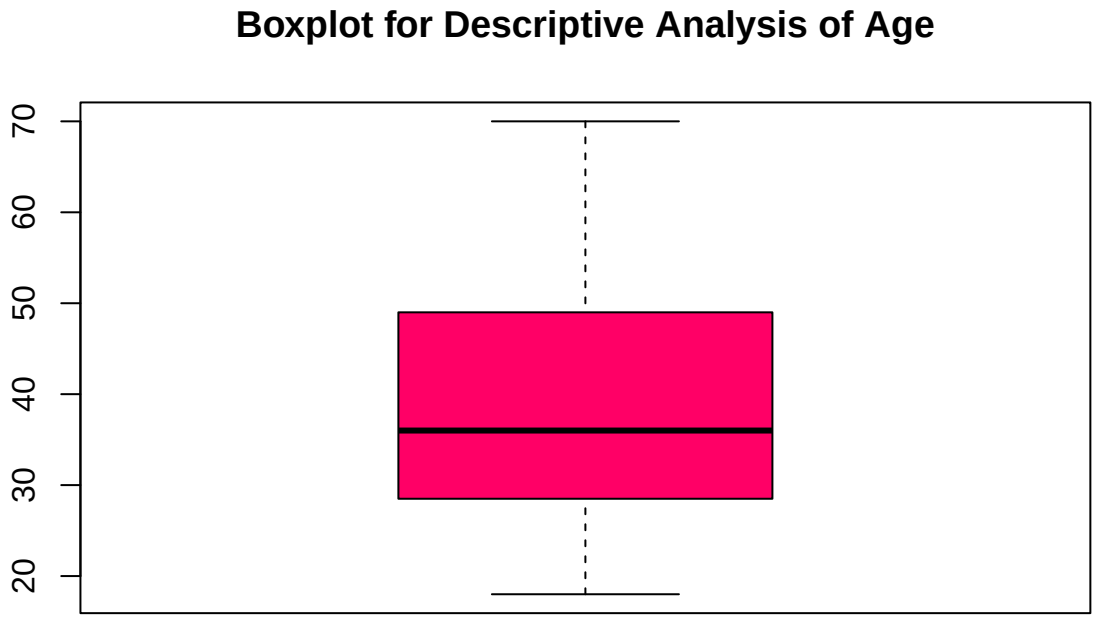
```
summary(customer_data$Age)
```

```
##    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="blue",
     main="Histogram to Show Count of Age Class",
     xlab="Age Class",
     ylab="Frequency",
     labels=TRUE)
```




```
boxplot(customer_data$Age,
        col="#ff0066",
        main="Boxplot for Descriptive Analysis of Age")
```



Analysis of the Annual income of the Customers
I conducted visualizations to analyze the annual income of the customers. Below are the key findings:

Histogram:
By plotting a histogram, it is observed the distribution of annual incomes among the customers. The analysis revealed that the minimum annual income is 15, and the maximum income is 137. The histogram showed that individuals earning an average income of 70 had the highest frequency count. The average salary of all customers in the dataset is 60.56.

Density Plot:
In addition, I used a Kernel Density Plot to examine the distribution of annual income. From the plot, I observed that the annual income followed a normal distribution.

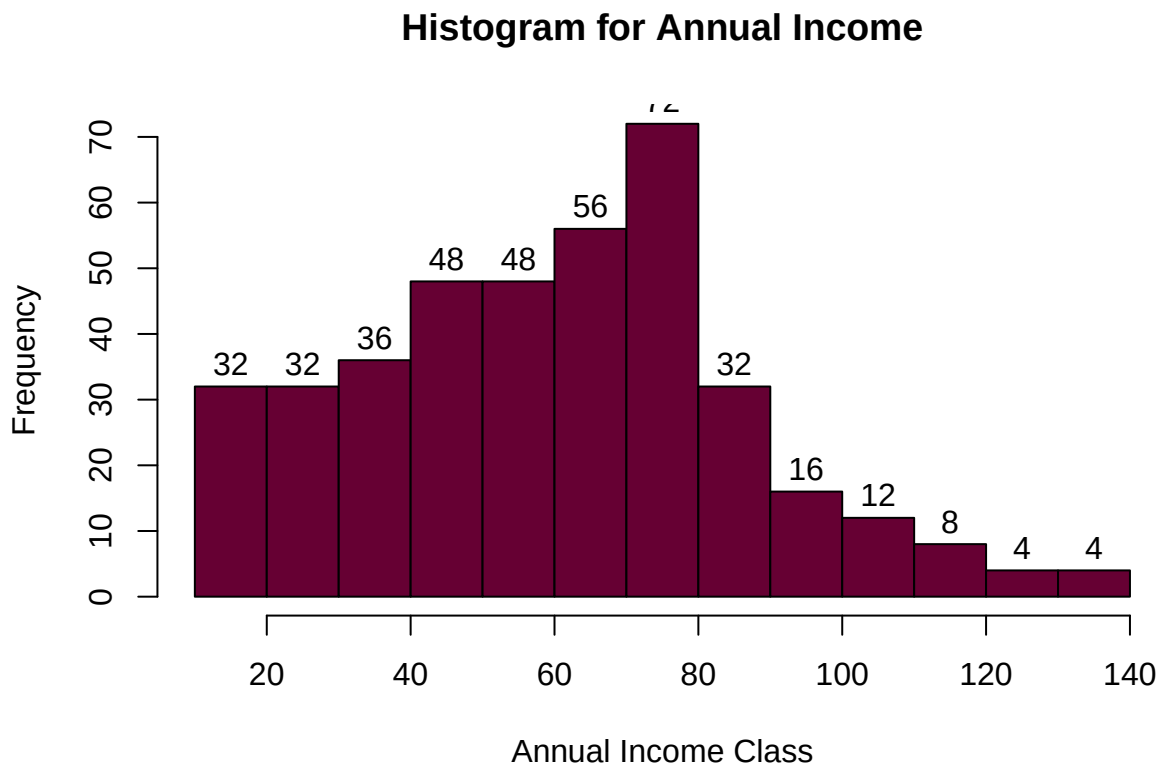
These visualizations provided valuable insights into the distribution and characteristics of customer annual incomes in my project.

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

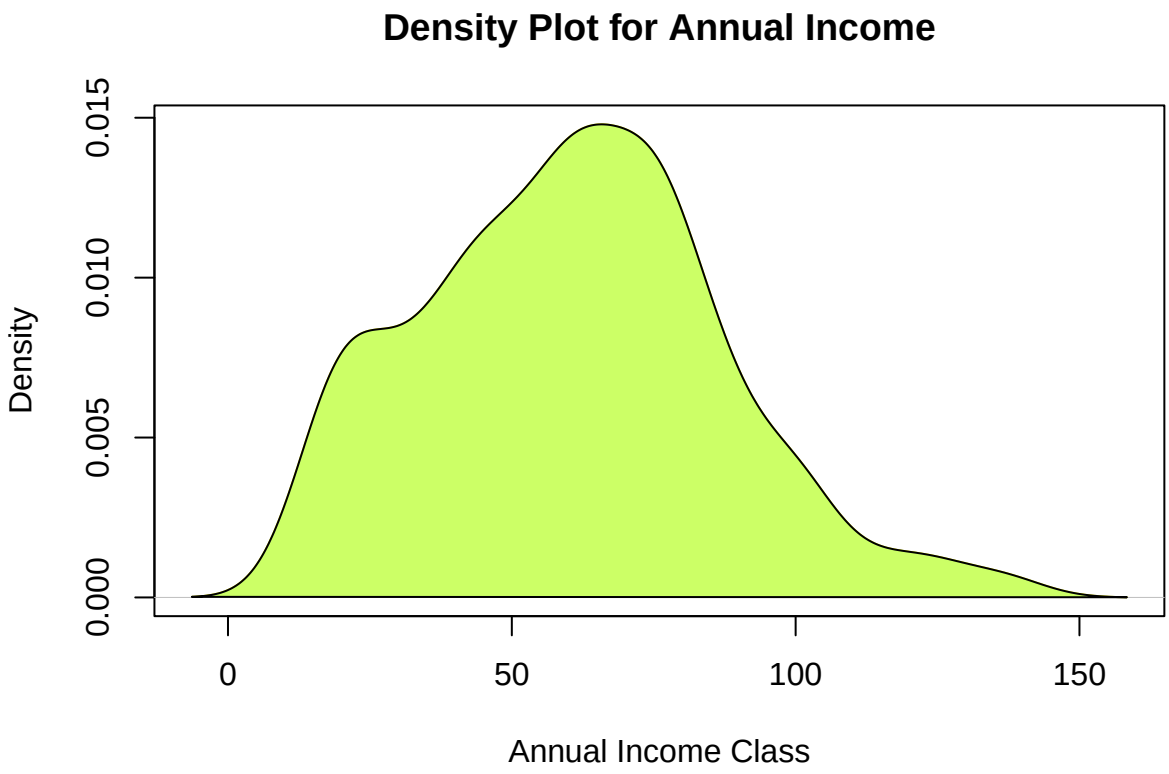
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   41.50   61.50   60.56   78.00  137.00
```

```
hist(customer_data$Annual.Income..k..,
     col="#660033",
     main="Histogram for Annual Income",
     xlab="Annual Income Class",
```

```
     ylab="Frequency",
     labels=TRUE)
```




```
plot(density(customer_data$Annual.Income..k..),
     col="yellow",
     main="Density Plot for Annual Income",
     xlab="Annual Income Class",
     ylab="Density")
polygon(density(customer_data$Annual.Income..k..),
        col="#ccff66")
```



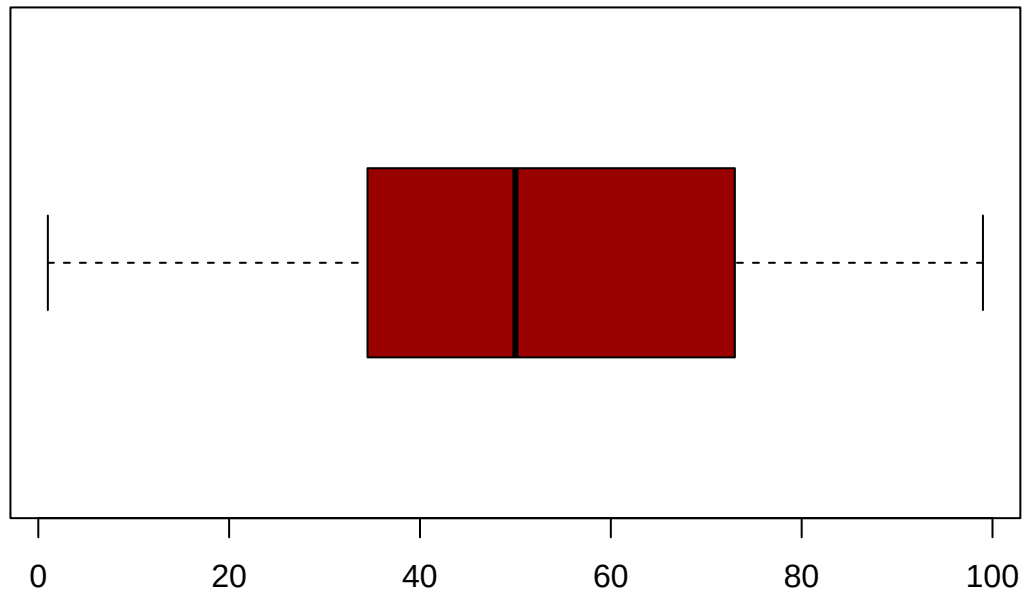
Analysis of the Annual Expenditure of the customers

I used a histogram and a boxplot to analyze the annual expenditure (Spending Score) of customers. The histogram provided insights into the distribution of spending scores, showing the majority of customers clustered around the median value of 50.00. The boxplot revealed the central tendency and variability of the data, including the minimum, maximum, median, and quartiles. These visualizations provided a comprehensive understanding of customer spending behavior, highlighting the range, concentration, and potential outliers in annual expenditure.

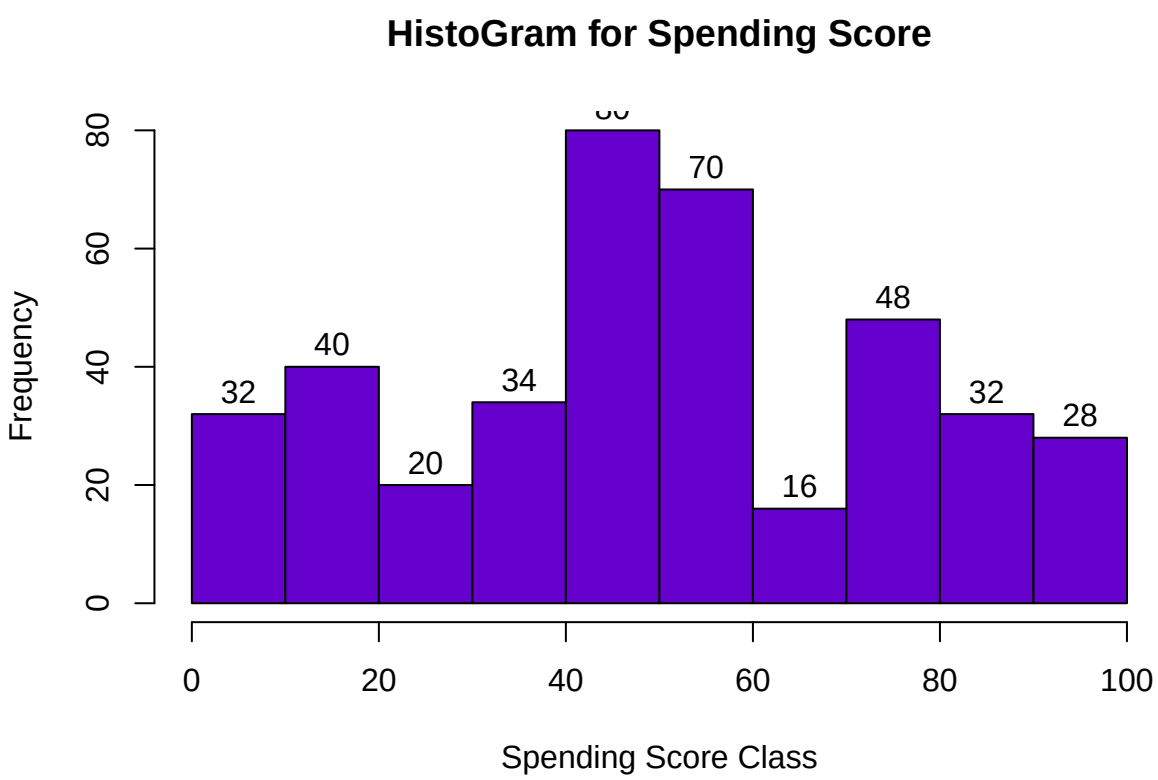
```
summary(customer_data$Spending.Score..1.100)
```

```
##    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.,
        horizontal=TRUE,
        col="#990000",
        main="BoxPlot for Descriptive Analysis of Spending Score")
```




```
hist(customer_data$Spending.Score..1.100.,
     main="HistoGram for Spending Score",
     xlab="Spending Score Class",
     ylab="Frequency",
     col="#6600cc",
     labels=TRUE)
```



K-Means Algorithm
I applied the K-means clustering algorithm to group data points into clusters. The algorithm involved the following steps:

1. Number of Clusters (k): I specified the desired number of clusters to be produced in the final output.

2. Initial Centers: Randomly selecting k objects from the dataset served as the initial cluster centers or centroids.

3. Cluster Assignment: I assigned each remaining object to the closest centroid based on the Euclidean distance between the object and the cluster mean. This iterative process ensured that each object was assigned to the most suitable cluster.

4. Recalculation of Cluster Means: After the initial assignment, I recalculated the new mean value for each cluster based on the observations within the cluster.

5. Reassignment: Objects were checked if they were closer to a different cluster, and if so, they were reassigned based on the updated cluster mean.

6. Iterations: I repeated steps 3 to 5 iteratively until the cluster assignments no longer changed, ensuring stable and consistent clusters.

To determine the optimal number of clusters, I calculated the clustering algorithm for various values of k and evaluated the total intra-cluster sum of squares (ISS). By plotting ISS against the number of clusters, I identified the appropriate number of clusters indicated by the bend or knee in the plot.

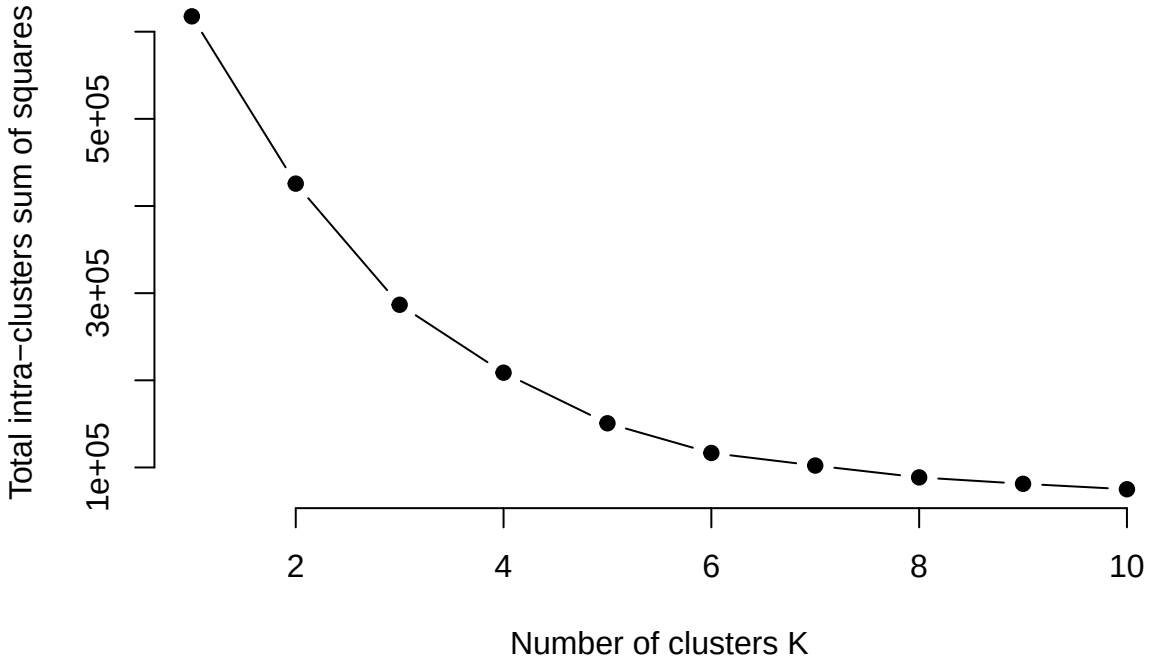
Through the implementation of the K-means algorithm and analysis of the ISS plot, I gained valuable insights into grouping data points into clusters based on their similarities and determined the optimal number of clusters for the dataset.

```
library(purrr)
set.seed(123)
# function to calculate total intra-cluster sum of square
iss <- function(k) {
  kmeans(customer_data[,3:5],k,iter.max=100,nstart=100,algorithm="Lloyd" )$tot.withinss
}

k.values <- 1:10


iss_values <- map_dbl(k.values, iss)

plot(k.values, iss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total intra-clusters sum of squares")


#Average Silhouette Method

library(cluster)
```

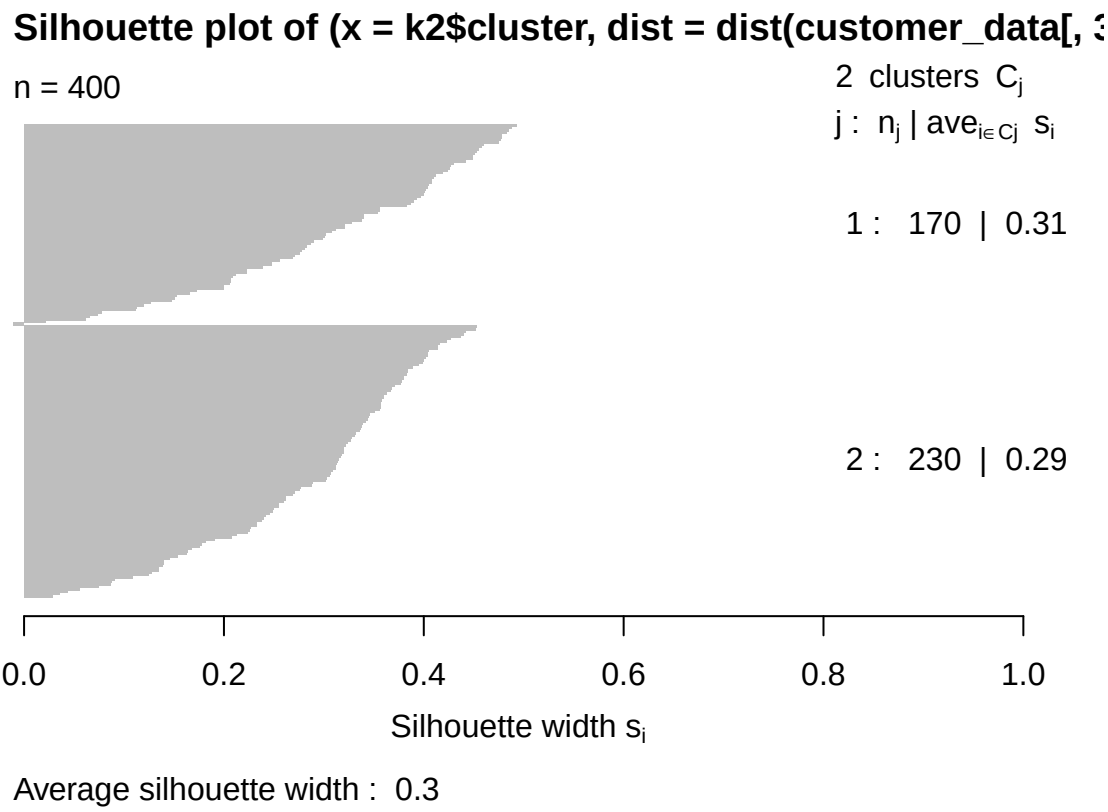
```
## Warning: package 'cluster' was built under R version 4.3.1
```

```
library(gridExtra)
```

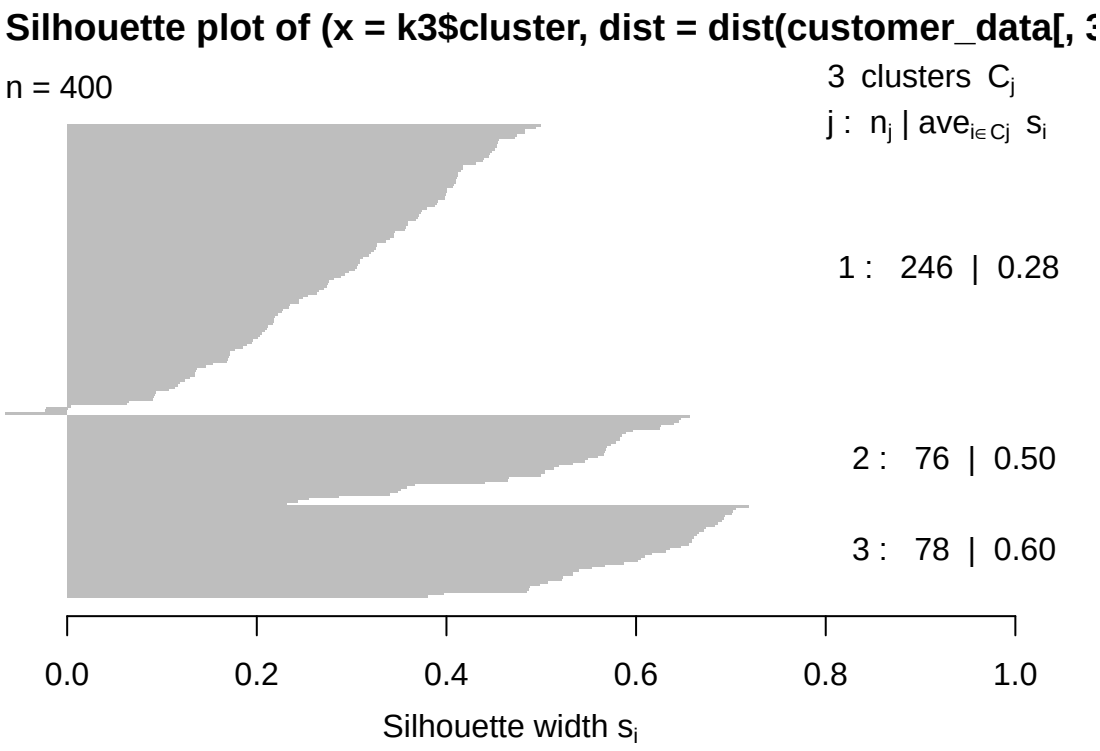
```
## Warning: package 'gridExtra' was built under R version 4.3.1
```




```
library(grid)


k2<-kmeans(customer_data[,3:5],2,iter.max=100,nstart=50,algorithm="Lloyd")
s2<-plot(silhouette(k2$cluster,dist(customer_data[,3:5],"euclidean")))
```

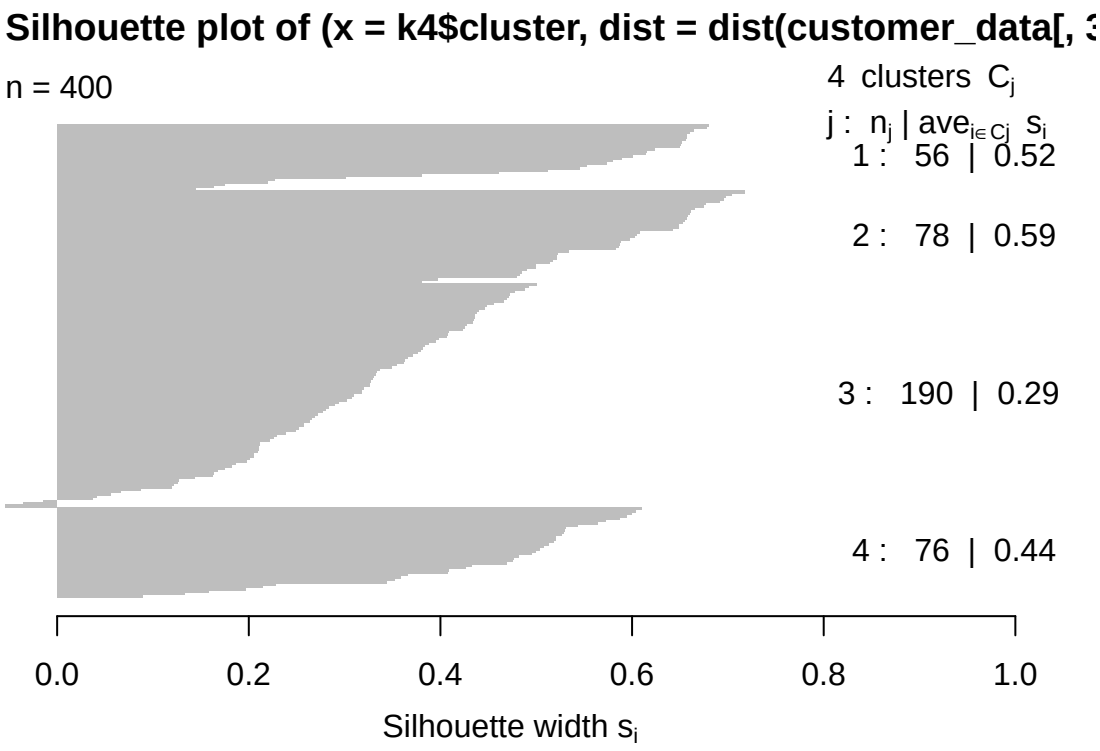



```
k3<-kmeans(customer_data[,3:5],3,iter.max=100,nstart=50,algorithm="Lloyd")
s3<-plot(silhouette(k3$cluster,dist(customer_data[,3:5],"euclidean")))
```



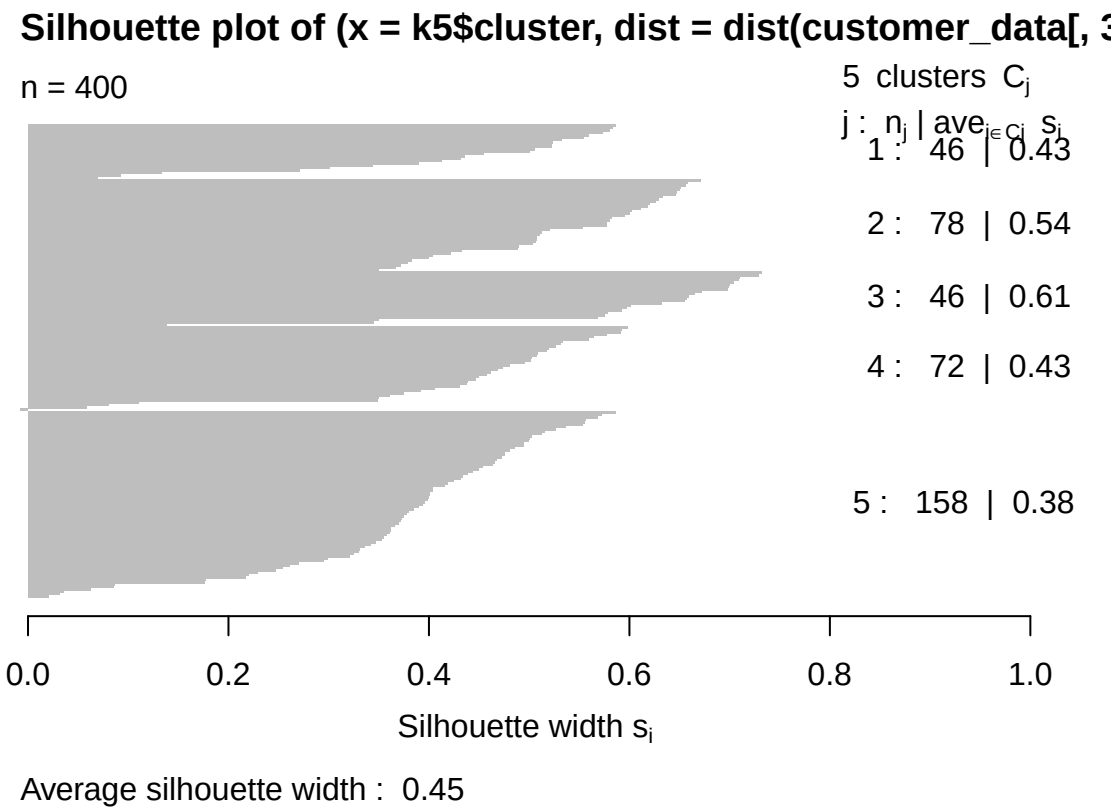
Average silhouette width : 0.39

```
k4<-kmeans(customer_data[,3:5],4,iter.max=100,nstart=50,algorithm="Lloyd")
s4<-plot(silhouette(k4$cluster,dist(customer_data[,3:5],"euclidean")))
```

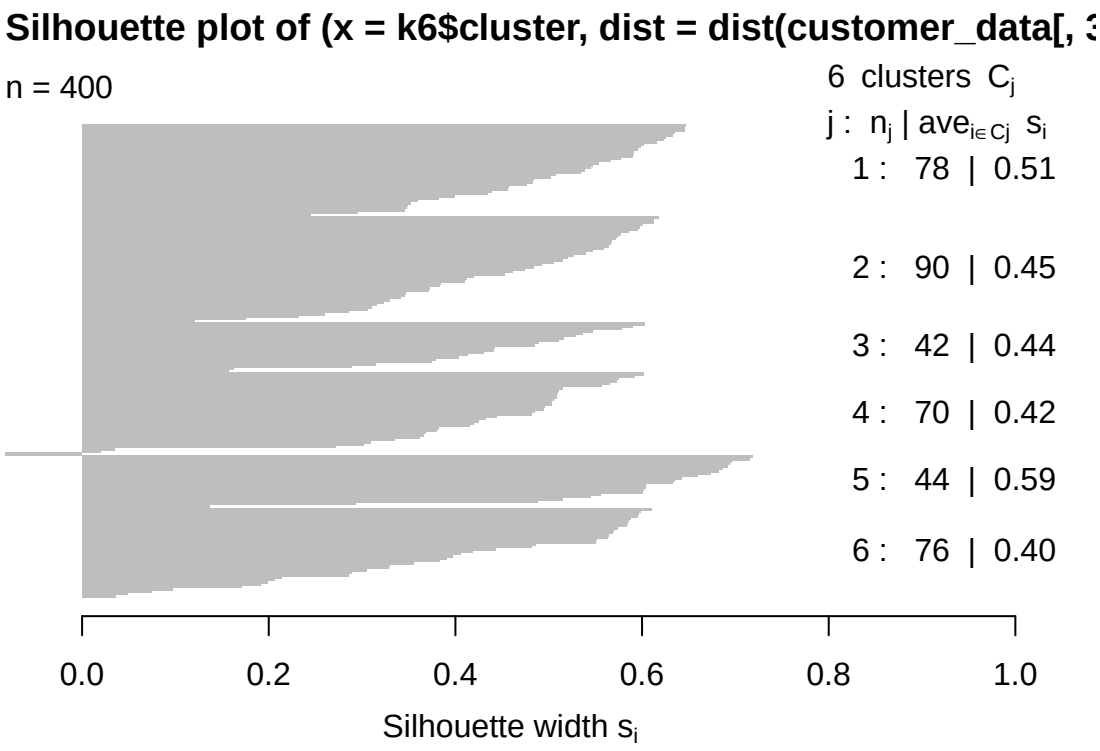


Average silhouette width : 0.41

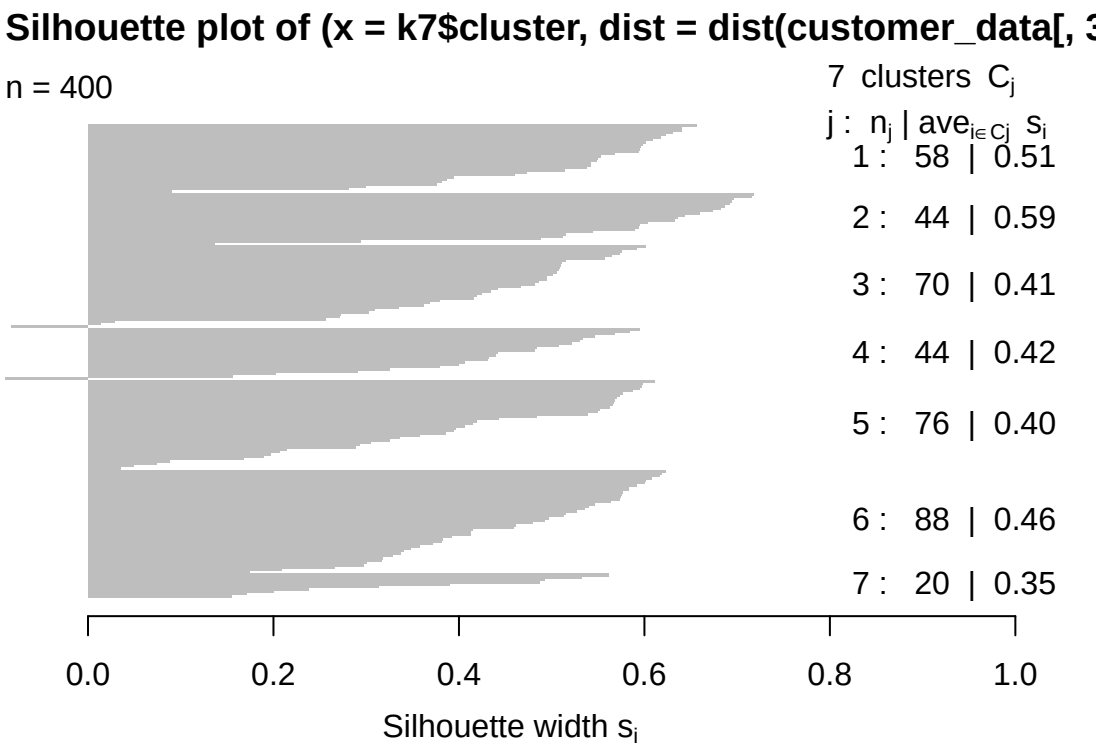
```
k5<-kmeans(customer_data[,3:5],5,iter.max=100,nstart=50,algorithm="Lloyd")
s5<-plot(silhouette(k5$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
s6<-plot(silhouette(k6$cluster,dist(customer_data[,3:5],"euclidean")))
```

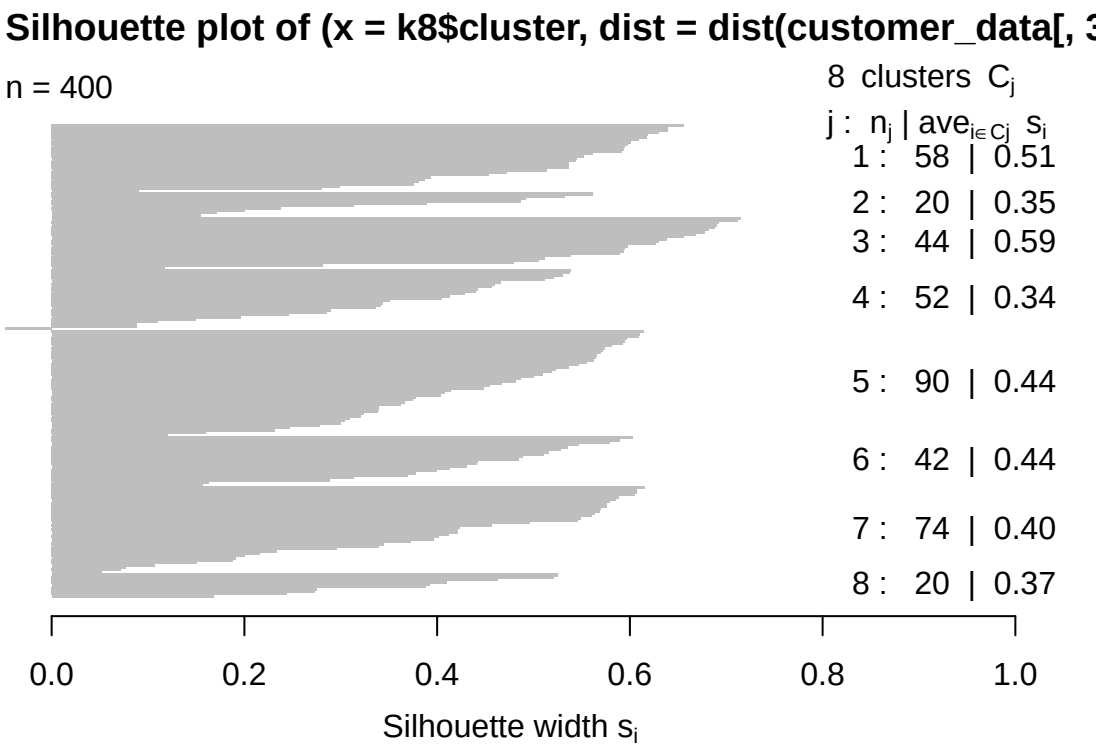



```
k7<-kmeans(customer_data[,3:5],7,iter.max=100,nstart=50,algorithm="Lloyd")
s7<-plot(silhouette(k7$cluster,dist(customer_data[,3:5],"euclidean")))
```

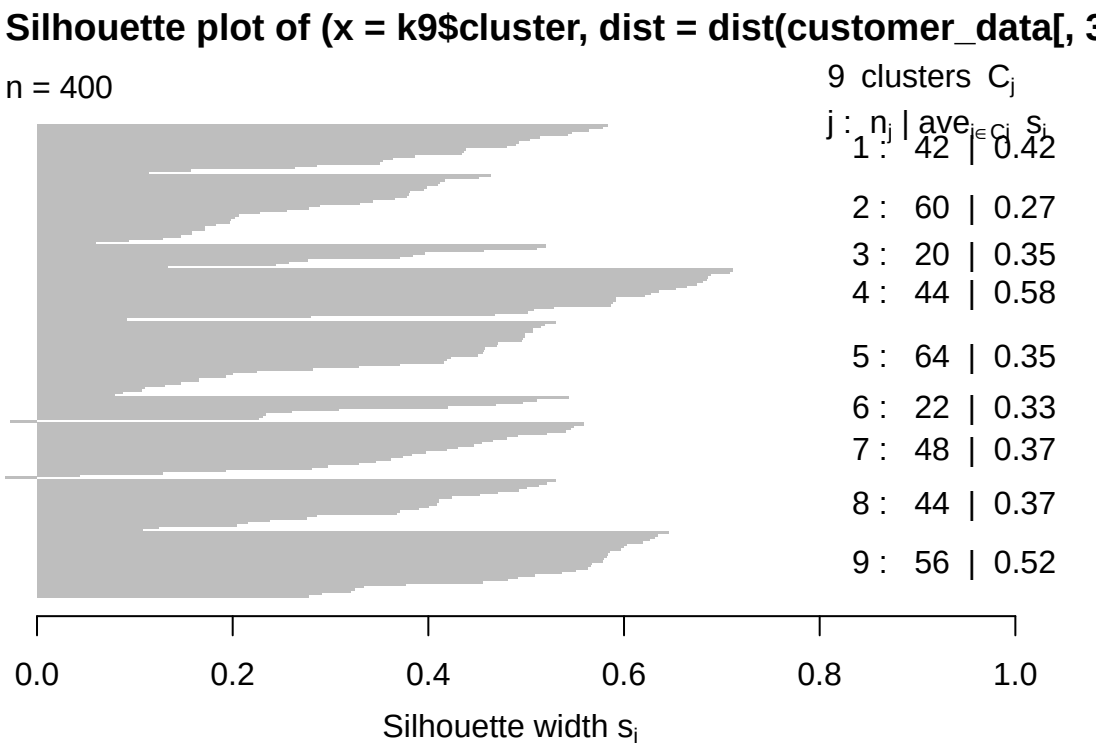


Average silhouette width : 0.45

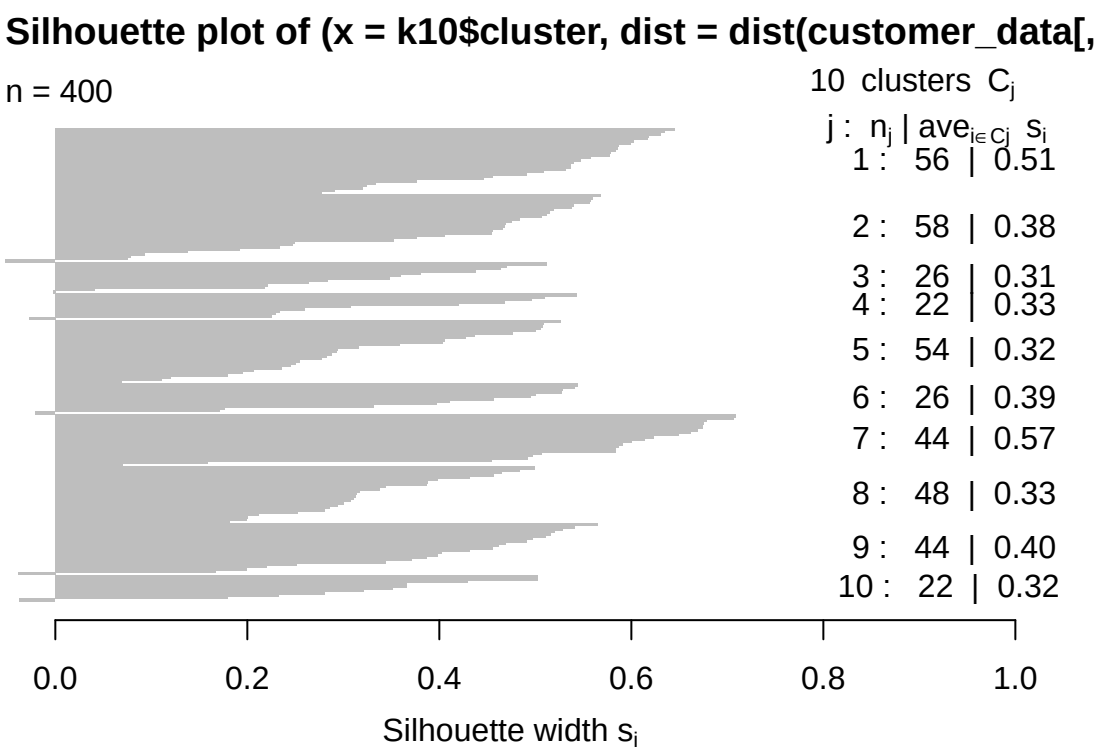
```
k8<-kmeans(customer_data[,3:5],8,iter.max=100,nstart=50,algorithm="Lloyd")
s8<-plot(silhouette(k8$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
k9<-kmeans(customer_data[,3:5],9,iter.max=100,nstart=50,algorithm="Lloyd")
s9<-plot(silhouette(k9$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
k10<-kmeans(customer_data[,3:5],10,iter.max=100,nstart=50,algorithm="Lloyd")
s10<-plot(silhouette(k10$cluster,dist(customer_data[,3:5],"euclidean")))
```




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

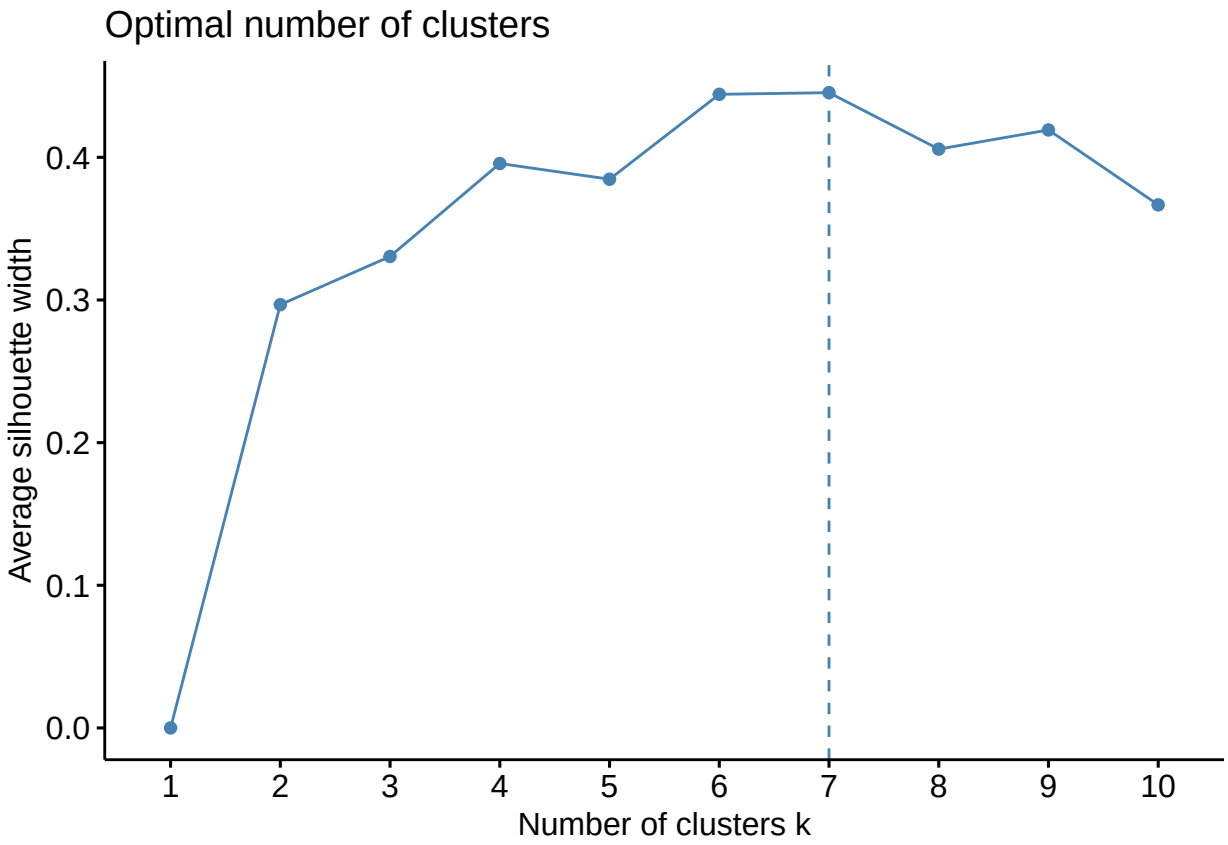
```
## Warning: package 'factoextra' was built under R version 4.3.1
```

```
## Loading required package: ggplot2
```

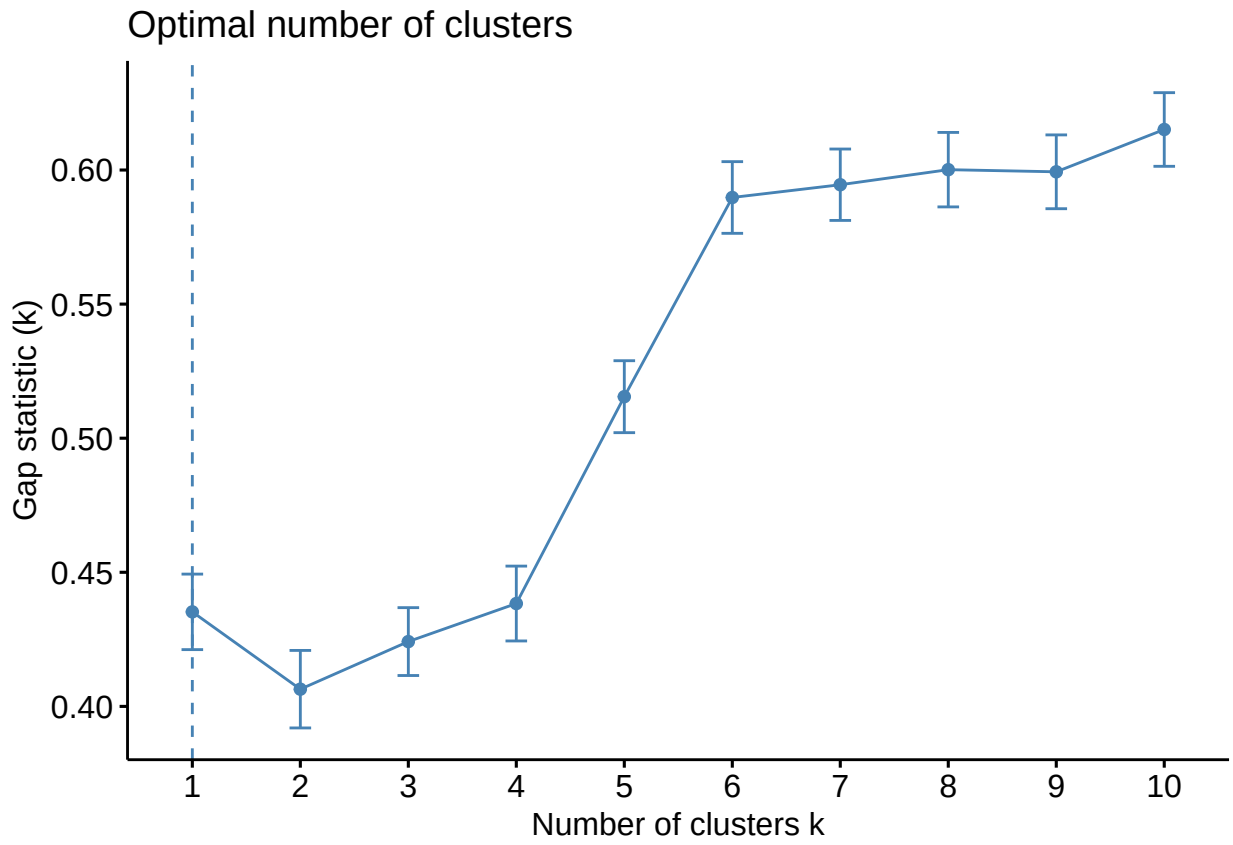
```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

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

```
fviz_nbclust(customer_data[,3:5], kmeans, method = "silhouette")
```




```
set.seed(125)
stat_gap <- clusGap(customer_data[,3:5], FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
fviz_gap_stat(stat_gap)
```




```
k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
k6
```

```
## K-means clustering with 6 clusters of sizes 90, 76, 44, 78, 70, 42
##
## Cluster means:
##        Age Annual.Income..k.. Spending.Score..1.100.
## 1 56.15556           53.37778                49.08889
## 2 27.00000           56.65789                49.13158
## 3 25.27273           25.72727                79.36364
## 4 32.69231           86.53846                82.12821
## 5 41.68571           88.22857                17.28571
## 6 44.14286           25.14286                19.52381
##
## Clustering vector:
##   [1] 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3
##  [38] 3 6 3 1 3 1 2 6 3 1 2 2 2 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 1 2 2 1 1 1 1
##  [75] 1 2 1 2 2 1 1 2 1 1 2 1 1 2 2 1 1 2 1 2 2 2 1 2 1 2 2 1 1 2 1 2 1 1 1 1 1
## [112] 2 2 2 2 2 1 1 1 1 2 2 2 4 2 4 5 4 5 4 5 4 2 4 5 4 5 4 5 4 5 4 2 4 5 4 5 4
## [149] 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5
## [186] 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3
## [223] 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 6 3 1 3 1 2 6 3 1 2 2 2 1 2 2 1 1 1 1 1 2
## [260] 1 1 2 1 1 1 2 1 1 2 2 1 1 1 1 1 2 1 2 2 1 1 2 1 1 2 1 1 2 2 1 1 2 1 2 2 2
## [297] 1 2 1 2 2 1 1 2 1 2 1 1 1 1 1 2 2 2 2 2 1 1 1 1 2 2 2 4 2 4 5 4 5 4 5 4 2
## [334] 4 5 4 5 4 5 4 5 4 5 4 2 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4
## [371] 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4
```

```
##
## Within cluster sum of squares by cluster:
## [1] 16124.267 15485.789  8199.636 27944.718 33381.714 15464.762
##  (between_SS / total_SS =  81.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Visualizing the Results Using the first two Principle Components

I visualized the clustering results using the first two principal components through a line plot. This graph displayed a series of data points connected by straight line segments.

From the visualization, it is observed the presence of 6 distinct clusters, each with unique characteristics:

Cluster 6 and 4: These clusters represent customers with medium income and medium annual spending.

Cluster 1: This cluster comprises customers with high annual income and high annual spending.

Cluster 3: Customers in this cluster have low annual income and low annual spending.

Cluster 2: This cluster includes customers with high annual income but low annual spending.

Cluster 5: Customers in this cluster have low annual income but high annual spending.

Furthermore, when examining the clustering results based on the first two principal components (PCA), the clusters exhibited the following patterns:

Cluster 4 and 1: These clusters consist of customers with medium PCA1 and PCA2 scores.

Cluster 6: This cluster represents customers with high PCA2 and low PCA1 scores.

Cluster 5: Customers in this cluster have medium PCA1 and low PCA2 scores.

Cluster 3: This cluster comprises customers with high PCA1 scores and high PCA2 scores.

Cluster 2: Customers in this cluster have high PCA2 scores and medium annual spending.

Through clustering, there is a comprehensive understanding of the variables, allowing for more informed decision-making. By identifying specific customer segments based on income, age, spending patterns, and other parameters, companies can better target their products and services. Additionally, the consideration of more complex patterns like product reviews contributes to improved segmentation and personalized offerings.

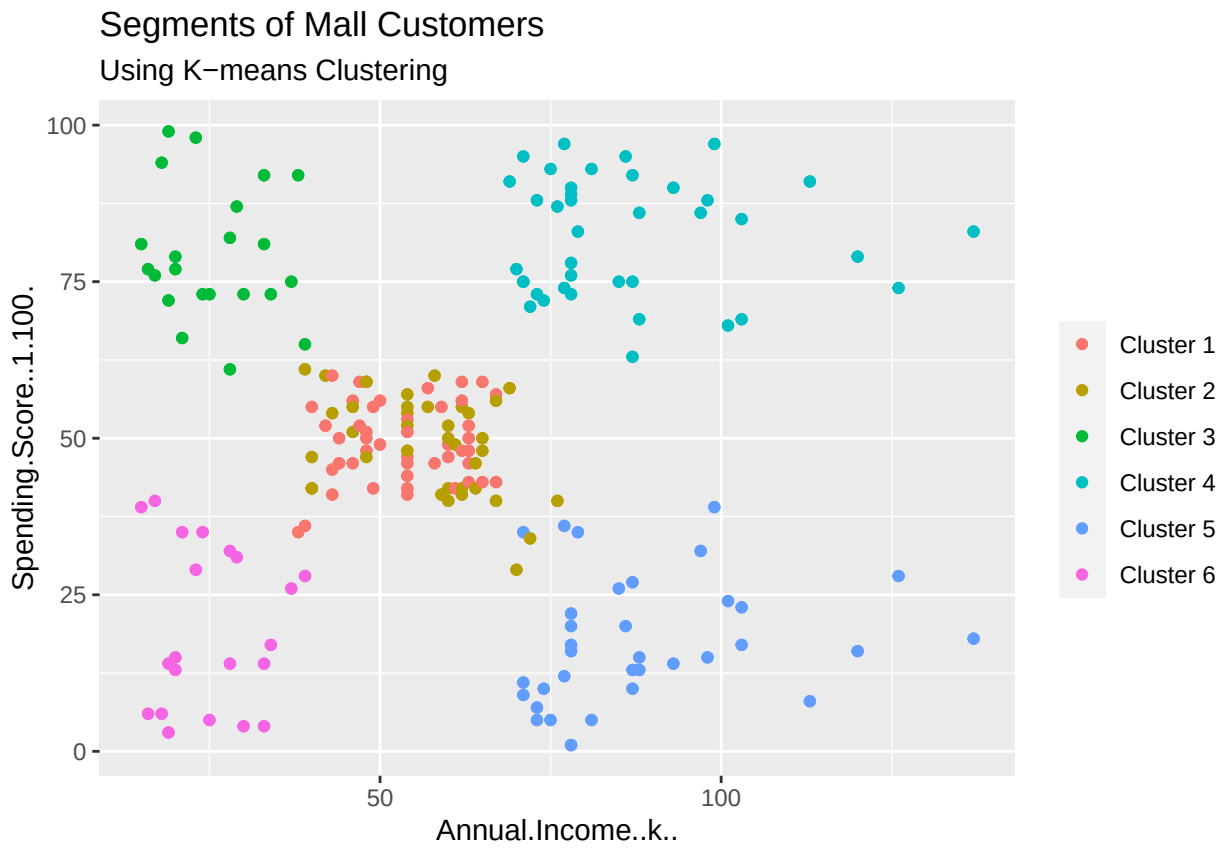
```
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)
```

```
## Importance of components:
##                             PC1     PC2     PC3
## Standard deviation     26.4293 26.1269 12.9155
## Proportion of Variance  0.4512  0.4410  0.1078
## Cumulative Proportion   0.4512  0.8922  1.0000
```

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

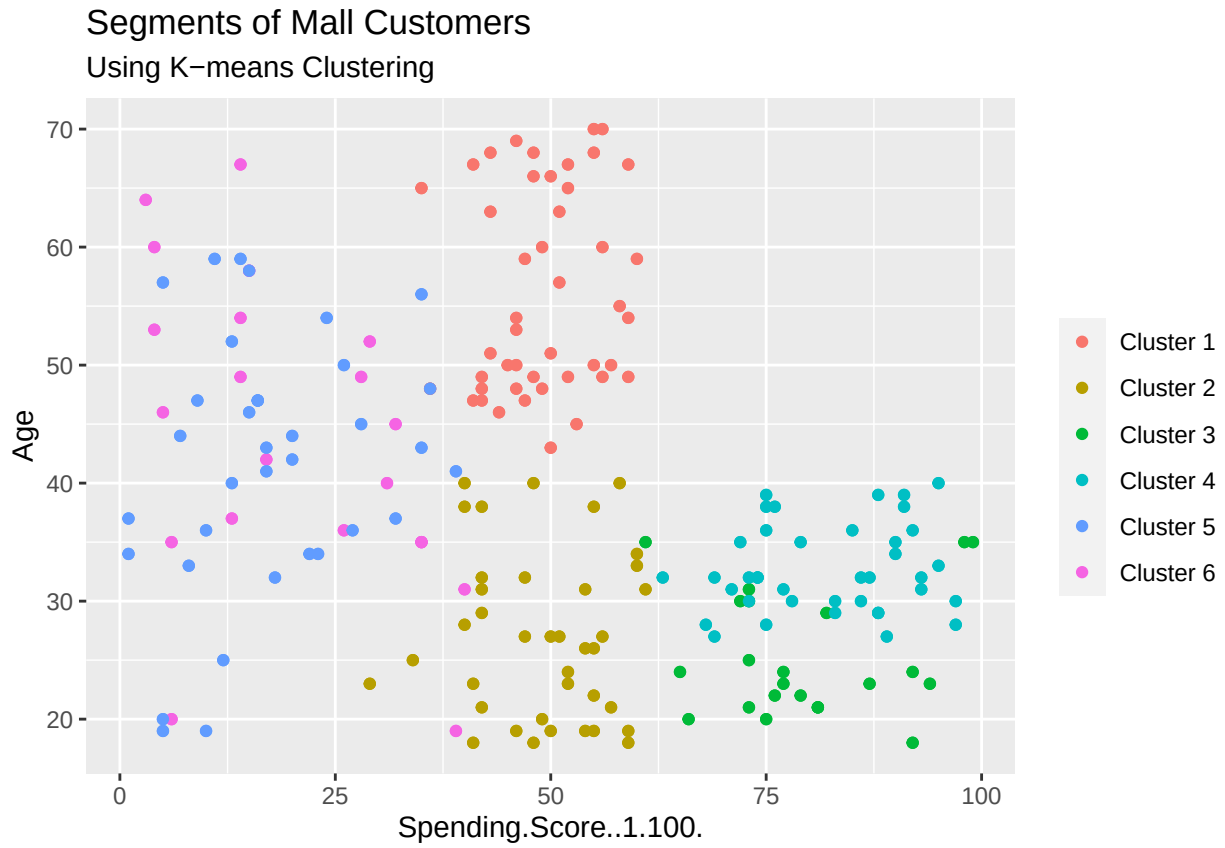
```
##                                PC1        PC2
## Age                      0.1889742 -0.1309652
## Annual.Income..k..      -0.5886410 -0.8083757
## Spending.Score..1.100. -0.7859965  0.5739136
```

```
set.seed(1)
ggplot(customer_data, aes(x =Annual.Income..k.., y = 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","Cluste
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
```

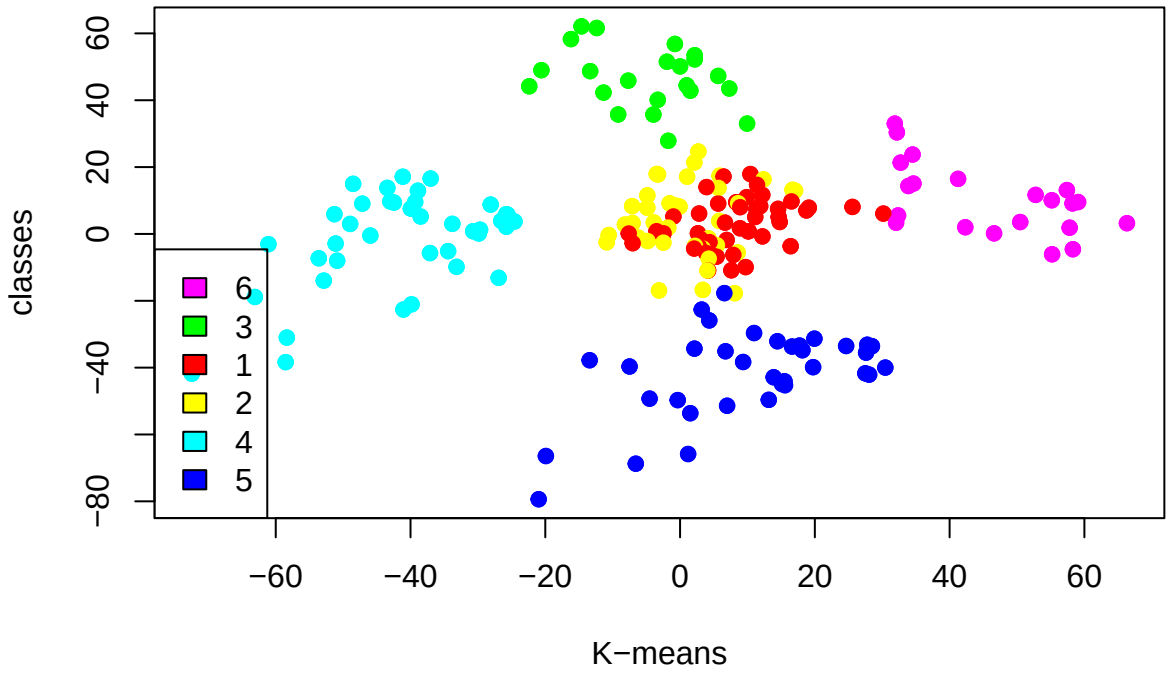



```
ggplot(customer_data, aes(x =Spending.Score..1.100., y =Age)) +
  geom_point(stat = "identity", aes(color = as.factor(k6$cluster))) +
  scale_color_discrete(name=" ",
```

```
                      breaks=c("1", "2", "3", "4", "5","6"),
                      labels=c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5","Cluste
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
```




```
kCols=function(vec){cols=rainbow (length (unique (vec)))
return (cols[as.numeric(as.factor(vec))])}

digCluster<-k6$cluster; dignm<-as.character(digCluster); # K-means clusters

plot(pcclust$x[,1:2], col =kCols(digCluster),pch =19,xlab ="K-means",ylab="classes")
legend("bottomleft",unique(dignm),fill=unique(kCols(digCluster)))
```



SUMMARY

The customer segmentation project successfully implemented the K-means clustering algorithm to divide the customer base into distinct groups based on income and spending patterns. This approach allowed for targeted marketing strategies and a deeper understanding of customer preferences. The project started with data exploration and visualization of customer characteristics such as gender, age distribution, and annual income. The insights gained from these visualizations provided valuable information about the customer base. The K-means algorithm was then employed to create clusters based on customer income and spending patterns, enabling businesses to develop tailored marketing strategies. Overall, the customer segmentation project demonstrated the benefits of understanding customer segments and highlighted the potential for optimizing marketing efforts and improving decision-making processes.

LIMITATIONS

Limited Variables: The project focused on a specific set of variables such as gender, age, and income. This limited scope may not capture all relevant factors that influence customer behavior, potentially leading to oversimplification and incomplete segmentation.

CONCLUSION

In conclusion, the customer segmentation project using the K-means clustering algorithm has successfully divided the customer base into distinct groups based on income and spending patterns. This segmentation approach can allow businesses to implement targeted marketing strategies and gain a deeper understanding of customer preferences. By tailoring marketing efforts to specific customer segments, businesses can optimize their marketing campaigns and drive better results. The project highlights the benefits of customer segmentation in enhancing marketing effectiveness and overall business success. Going forward, the insights gained from this project can serve as a valuable foundation for further research and implementation of customer segmentation strategies to drive customer satisfaction and business growth.

Pius Mithika
Data Scientist