Customer Segmentation Using Machine Learning

Shruti Jana

2024-03-24

Customer segmentation divides a customer base into groups with similar characteristics, such as gender, age, interests, and spending habits. This helps companies tailor marketing efforts to specific customer groups, understanding individual needs and preferences. Analyzing demographic, geographic, economic, and behavioral data helps identify valuable customer segments for targeted marketing strategies, optimizing techniques, and minimizing risks.

```
customer_data=read.csv("E:/Mall_Customers.csv")
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."
```

```
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] 13.96901
```

```
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
```

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

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

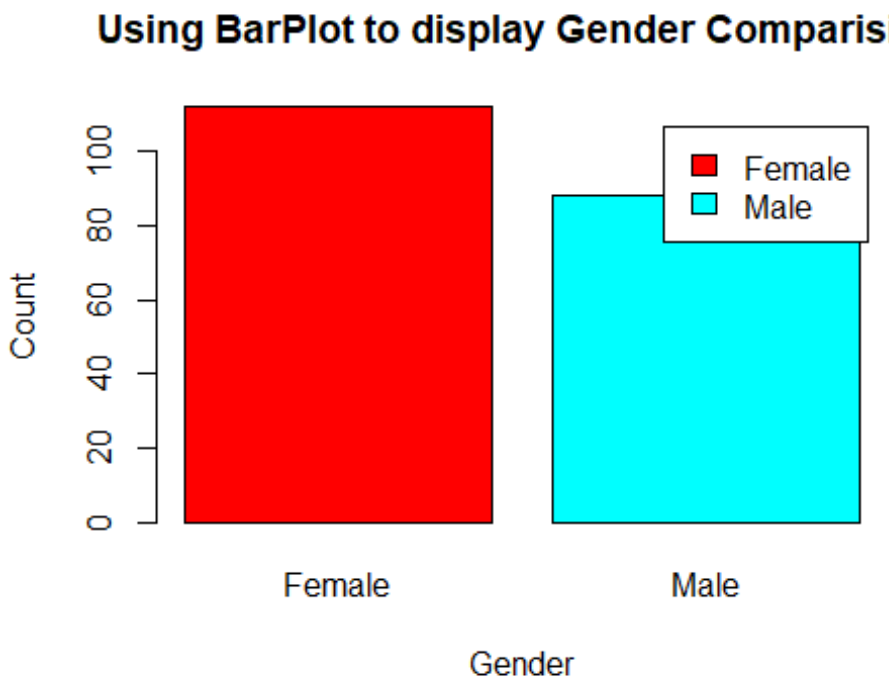
```
sd(customer_data$Spending.Score..1.100.)
```

```
## [1] 25.82352
```

Customer Gender Visualization

Customer gender distribution will be visualized through a bar plot and a pie chart.

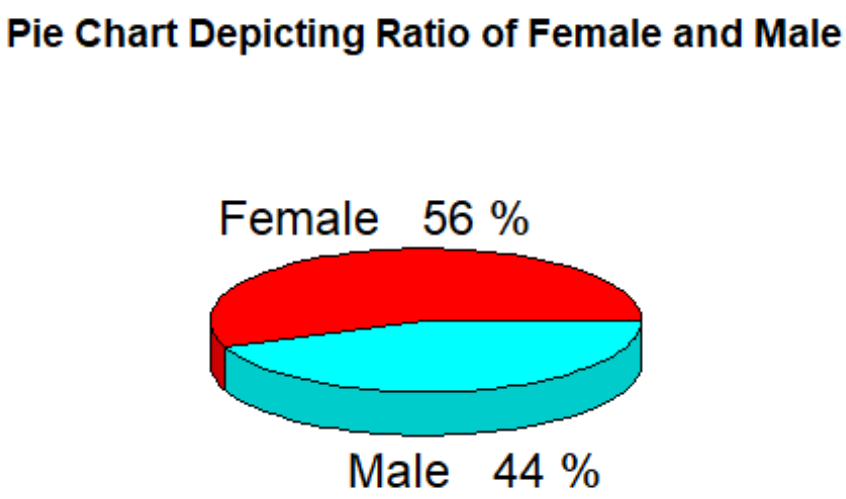
```
a=table(customer_data$Gender)
barplot(a,main="Using BarPlot to display Gender Comparision",
        ylab="Count",
        xlab="Gender",
        col=rainbow(2),
        legend=rownames(a))
```



From the bar plot, it is observed that the number of females exceeds that of males.

Now, a pie chart will be visualized to observe the distribution ratio of males and females.

```
pct=round(a/sum(a)*100)
lbs=paste(c("Female","Male")," ",pct,"%",sep=" ")
library(plotrix)

## Warning: package 'plotrix' was built under R version 4.3.2

pie3D(a,labels=lbs,
   main="Pie Chart Depicting Ratio of Female and Male")
```



From the above graph, it is concluded that the percentage of females is 56%, whereas the percentage of males in the customer dataset is 44%.

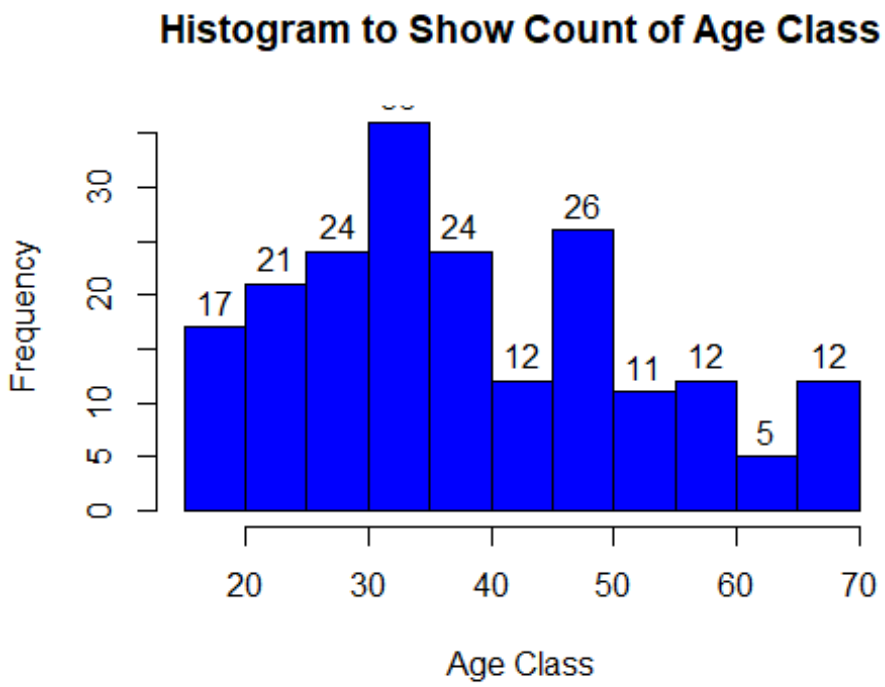
Customer Age Visualization

A histogram will be plotted to display the frequency of customer ages. First, a summary of the Age variable will be taken.

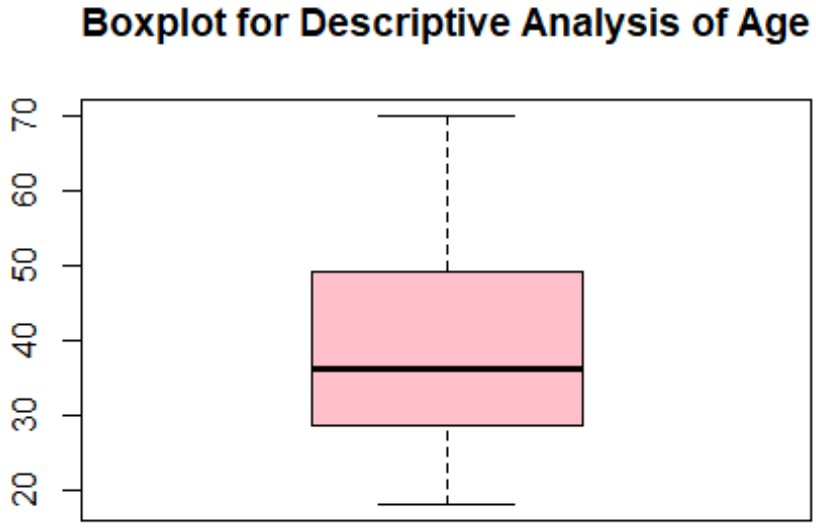
```
summary(customer_data$Age)
```

```
##    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)
```




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

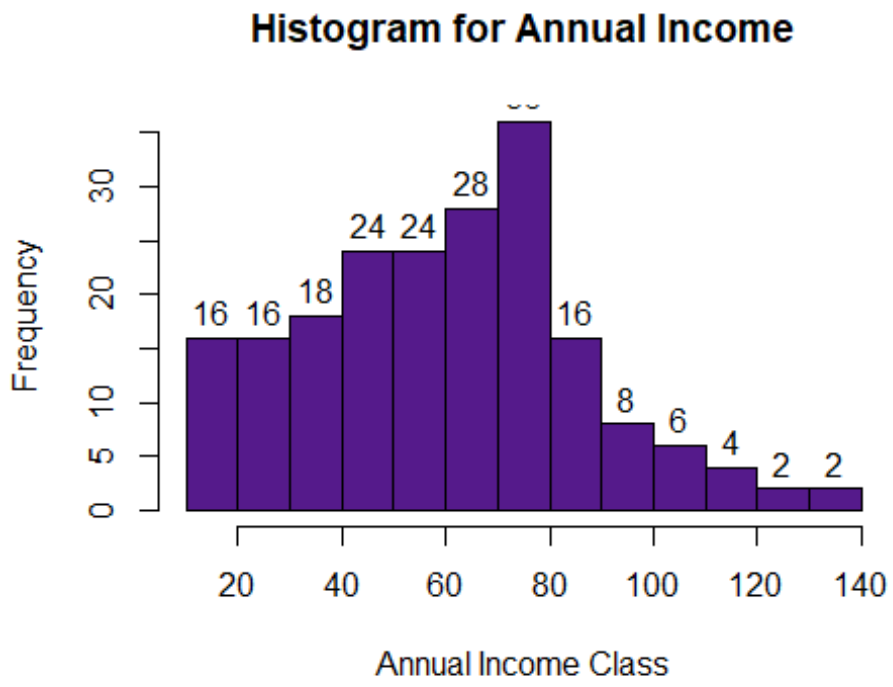


From the above two visualizations, it is concluded that the maximum customer ages are between 30 and 35. The minimum age of customers is 18, while the maximum age is 70.

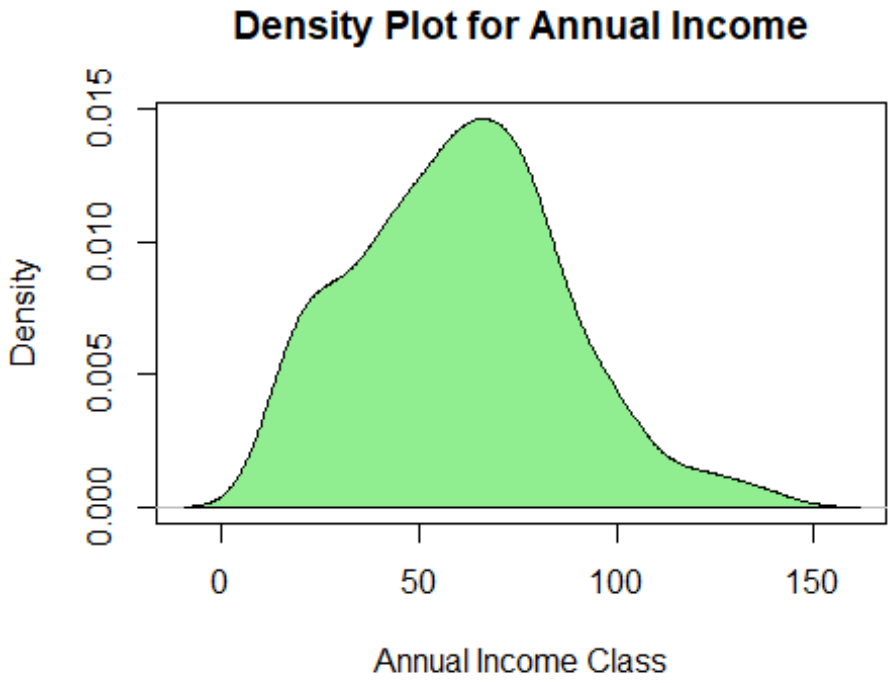
Analysis of the Annual Income of the Customers:

Visualizations will be created to analyze the annual income of the customers. A histogram will be plotted, followed by an examination of the 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..,
  col="purple4",
  main="Histogram for Annual Income",
  xlab="Annual Income Class",
  ylab="Frequency",
  labels=TRUE)
```



```r
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="lightgreen")
```

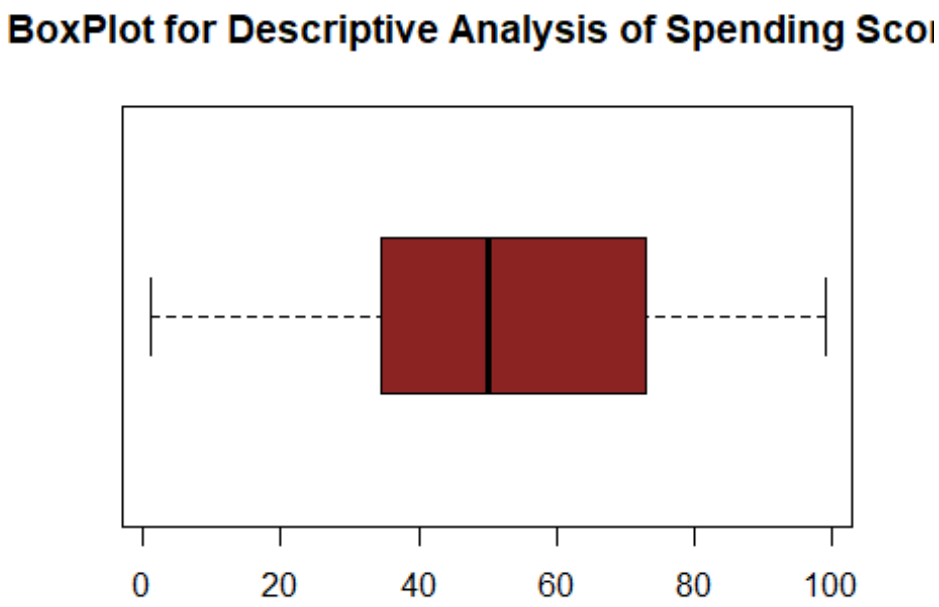


From the above descriptive analysis, it is concluded that the minimum annual income of the customers is 15, and the maximum income is 137. Individuals with an average income of 70 have the highest frequency count in our histogram distribution. The average salary of all the customers is 60.56. In the Kernel Density Plot displayed above, it is observed that the annual income follows a normal distribution.

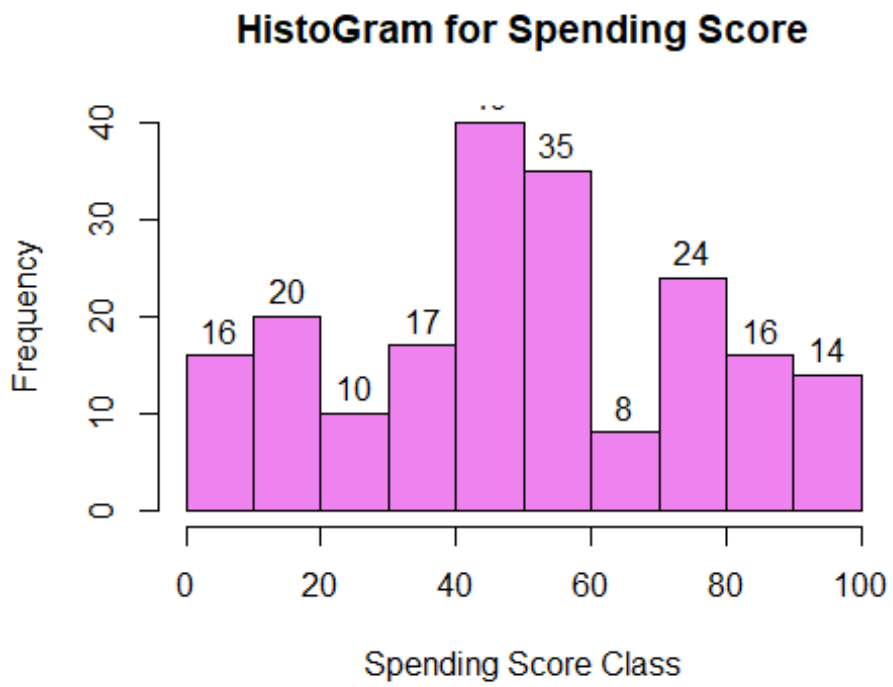
```r
summary(customer_data$Spending.Score..1.100.)
```

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

```r
boxplot(customer_data$Spending.Score..1.100.,
   horizontal=TRUE,
   col="brown4",
   main="BoxPlot for Descriptive Analysis of Spending Score")
```




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

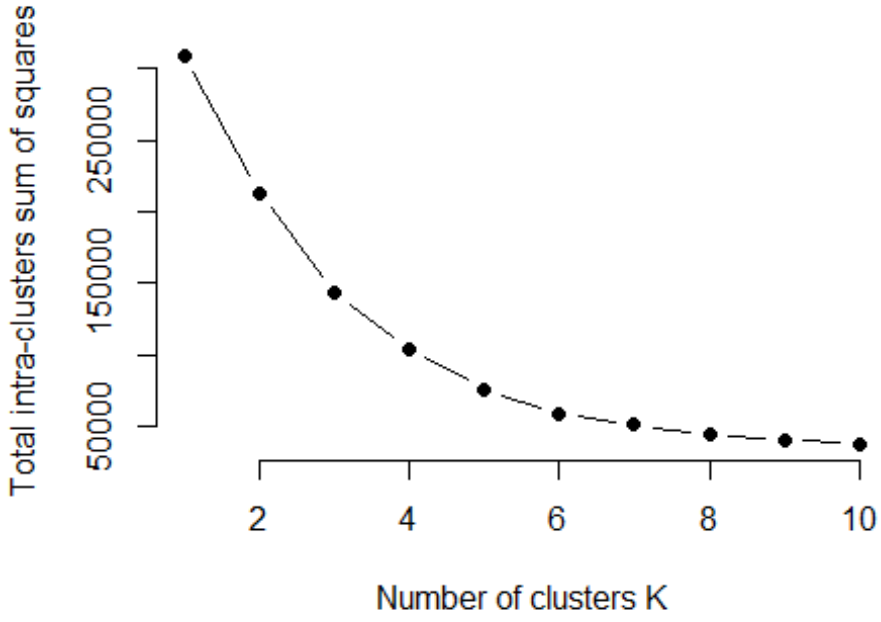


The descriptive analysis reveals that the spending score ranges from 1 to 99, with an average of 50.20. The histogram indicates that customers in the 40-50 spending score range are the most frequent.

- In the K-means algorithm, the number of desired clusters (k) is specified, initial centroids are randomly selected, and objects are assigned to the closest centroid based on Euclidean distance. Centroids are then updated iteratively until cluster assignments stabilize.
- For determining the optimal number of clusters, the elbow method involves plotting the total intra-cluster sum of squares (ISS) against the number of clusters. The point where the plot exhibits a bend or "elbow" signifies the optimal number of clusters.

```
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" )$to
t.withinss
```

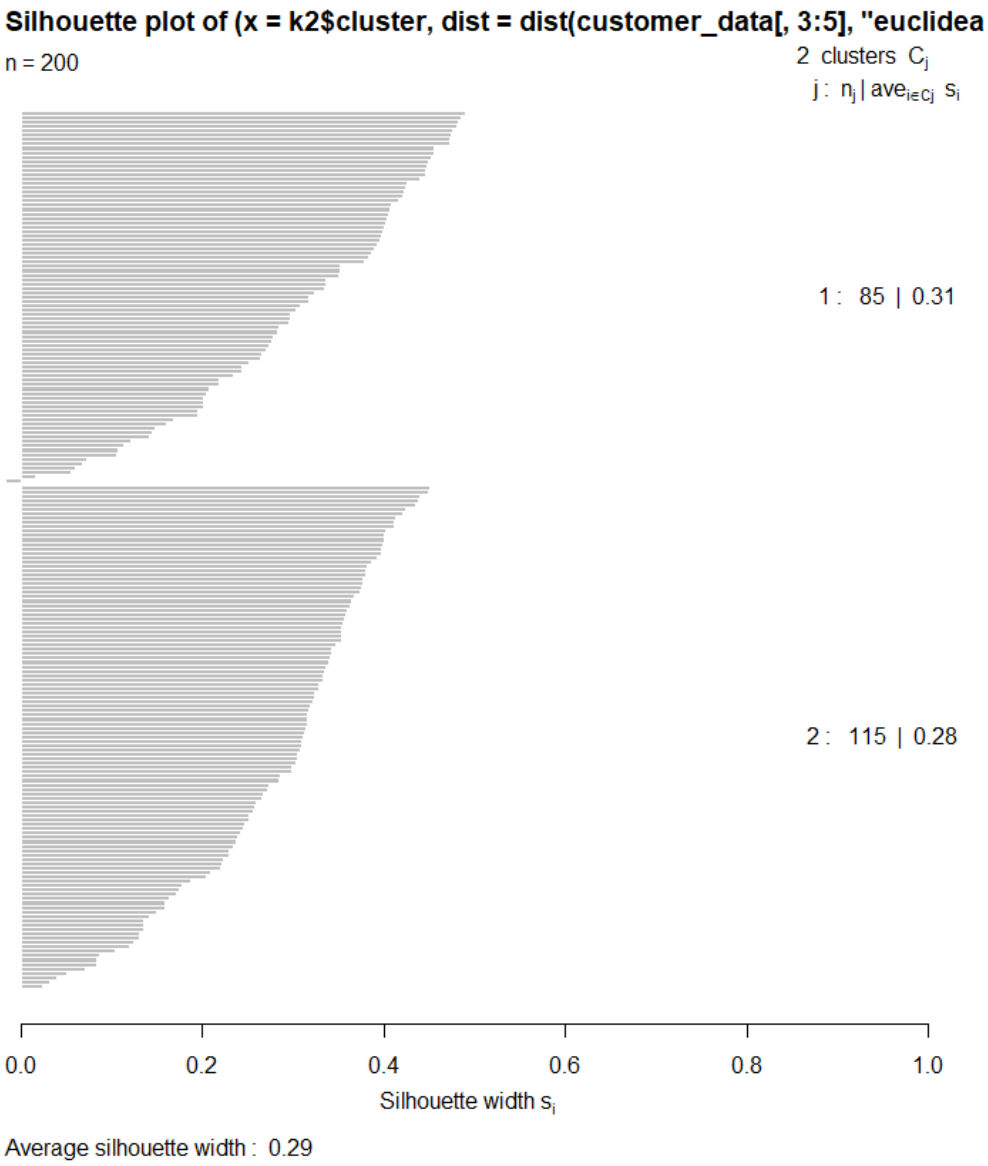
```
}

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")
```



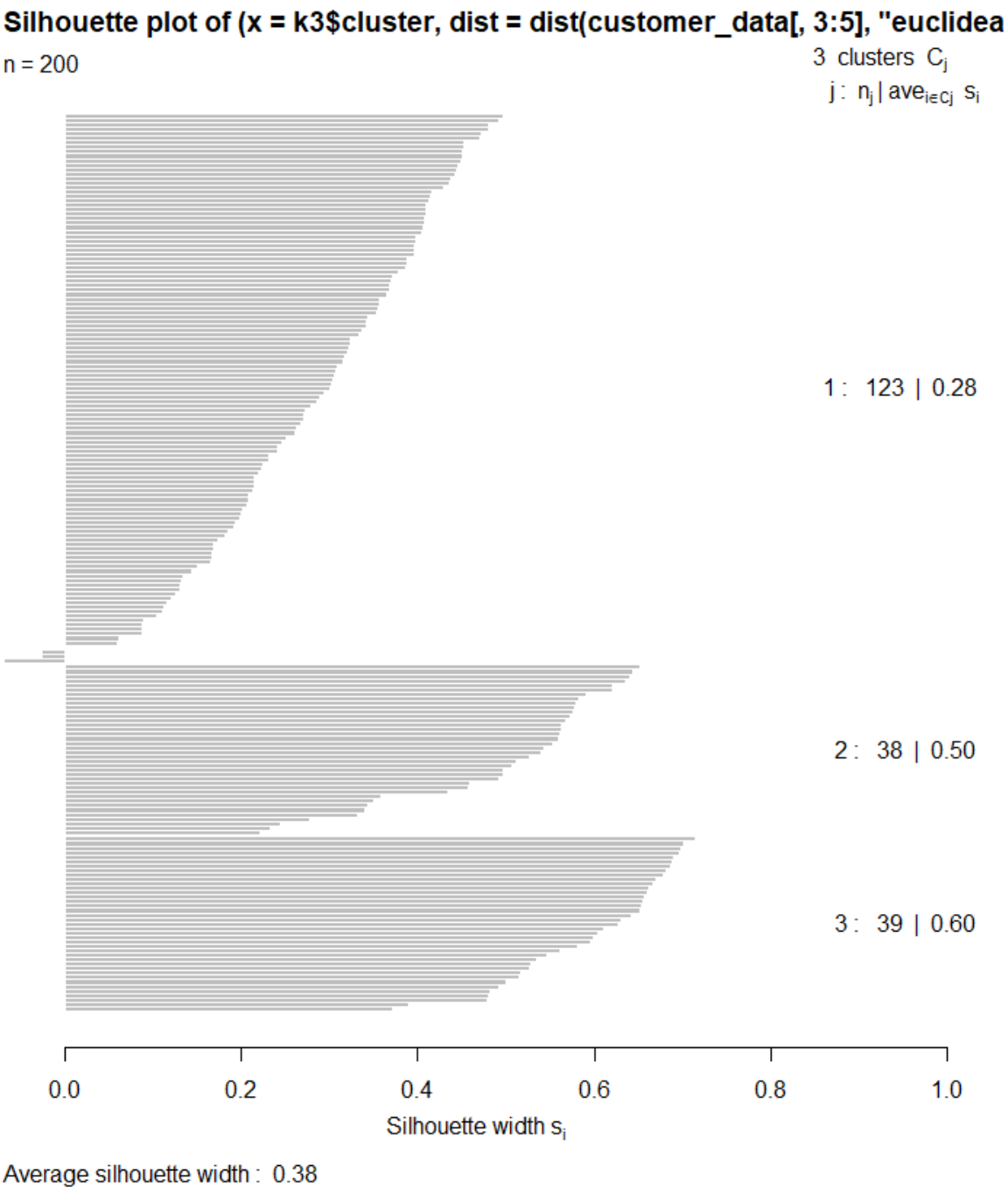
The graph suggests that 4 is the optimal number of clusters based on the elbow method.

- The average silhouette method measures clustering quality by evaluating how well data objects fit within clusters. It computes the mean silhouette width for different k values, aiming for higher averages. The silhouette function in the cluster package is used to calculate this, with the optimal cluster having the highest average silhouette width.

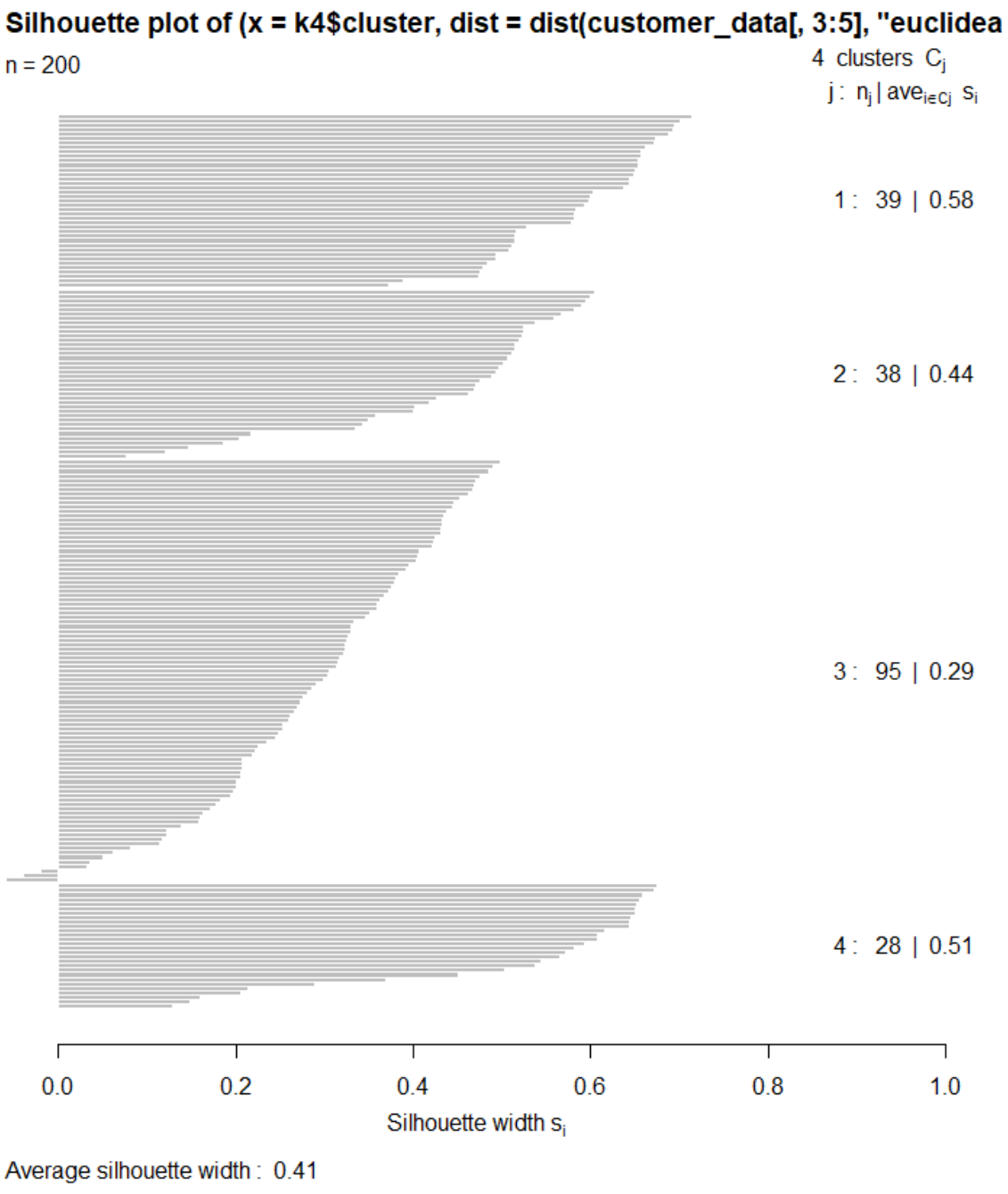
```r
library(cluster)
library(gridExtra)

## Warning: package 'gridExtra' was built under R version 4.3.2

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")))
```



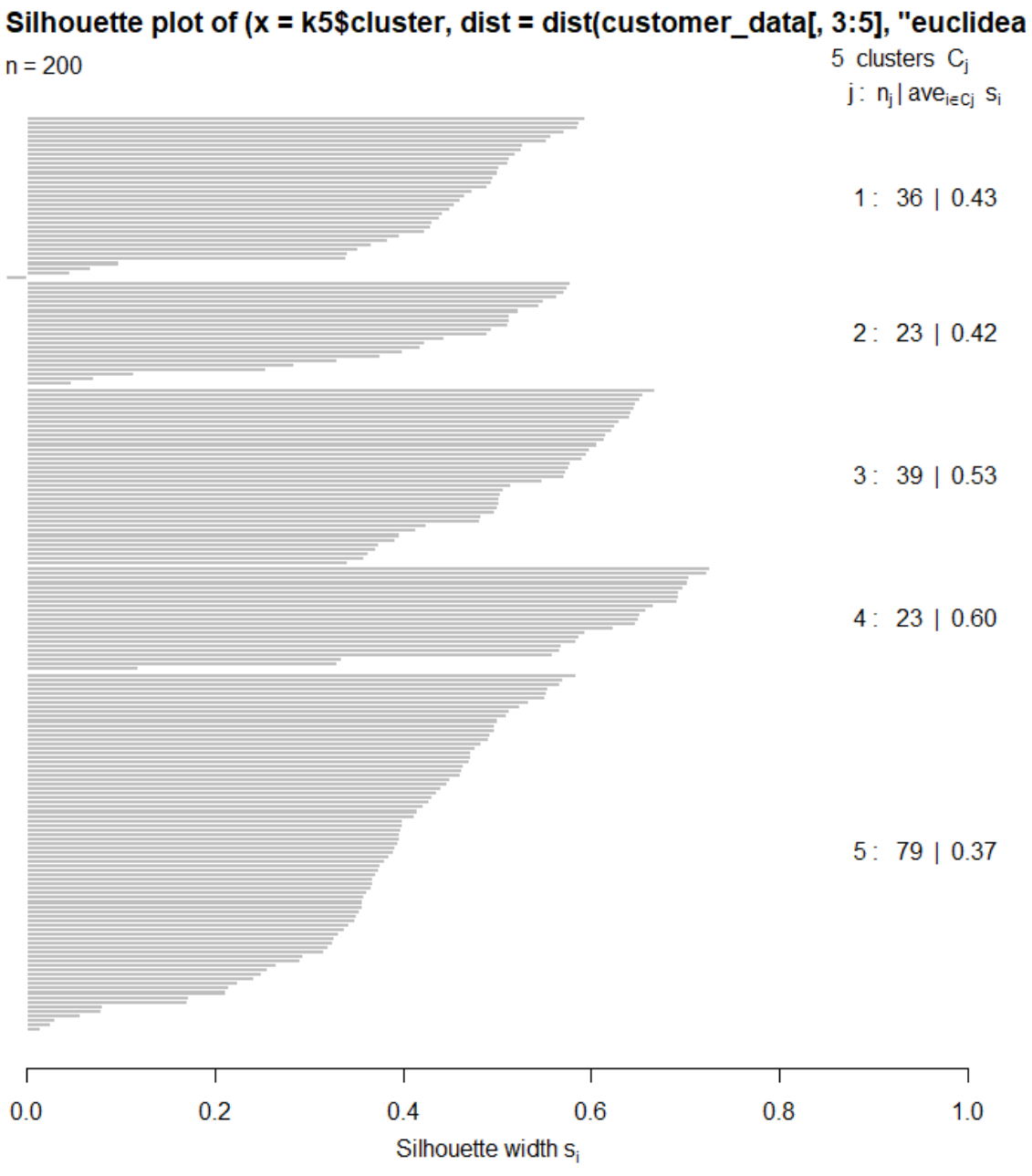
```r
k3<-kmeans(customer_data[,3:5],3,iter.max=100,nstart=50,algorithm="Lloyd")
s3<-plot(silhouette(k3$cluster,dist(customer_data[,3:5],"euclidean")))
```



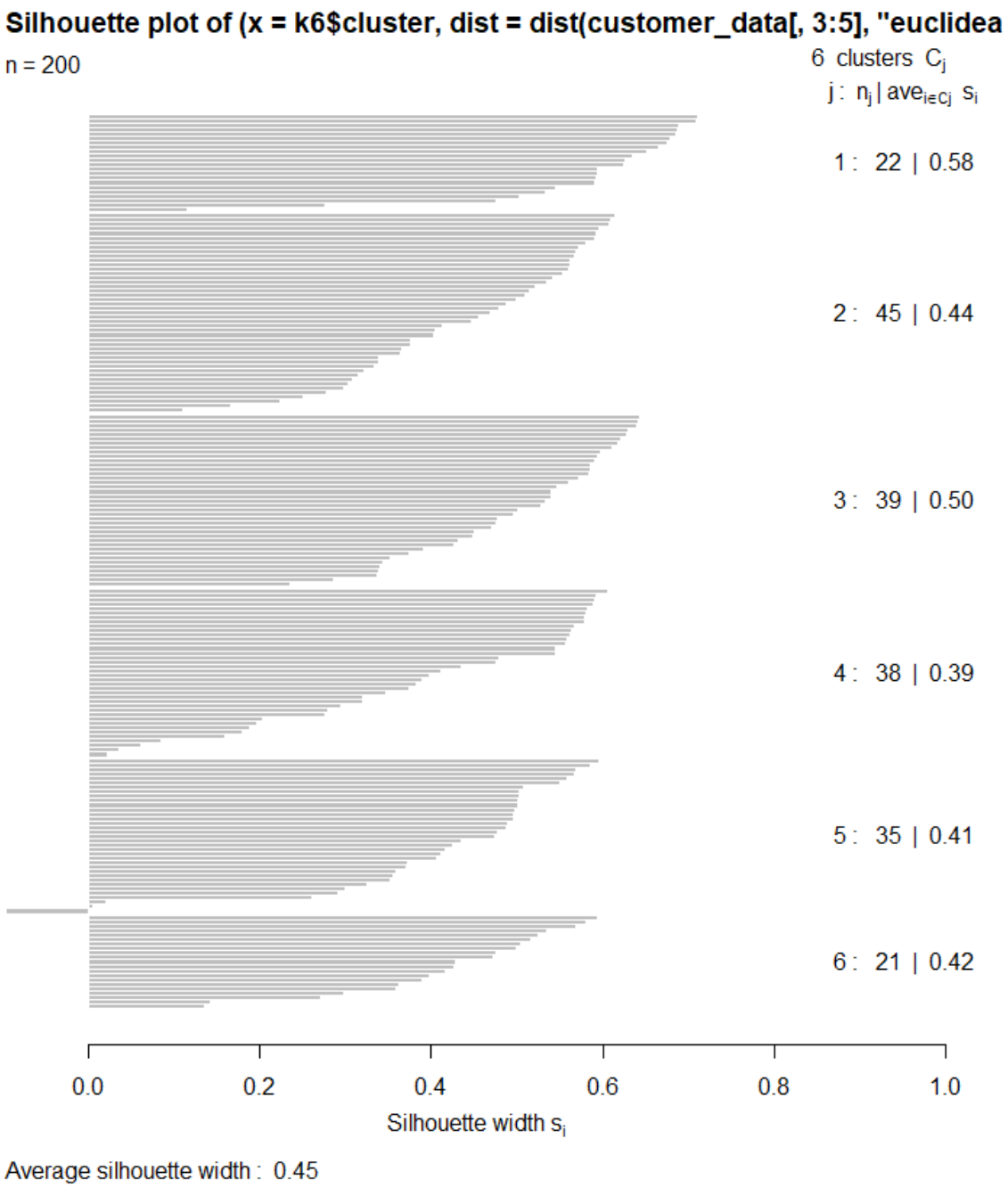
```
k4<-kmeans(customer_data[,3:5],4,iter.max=100,nstart=50,algorithm="Lloyd")
s4<-plot(silhouette(k4$cluster,dist(customer_data[,3:5],"euclidean")))
```



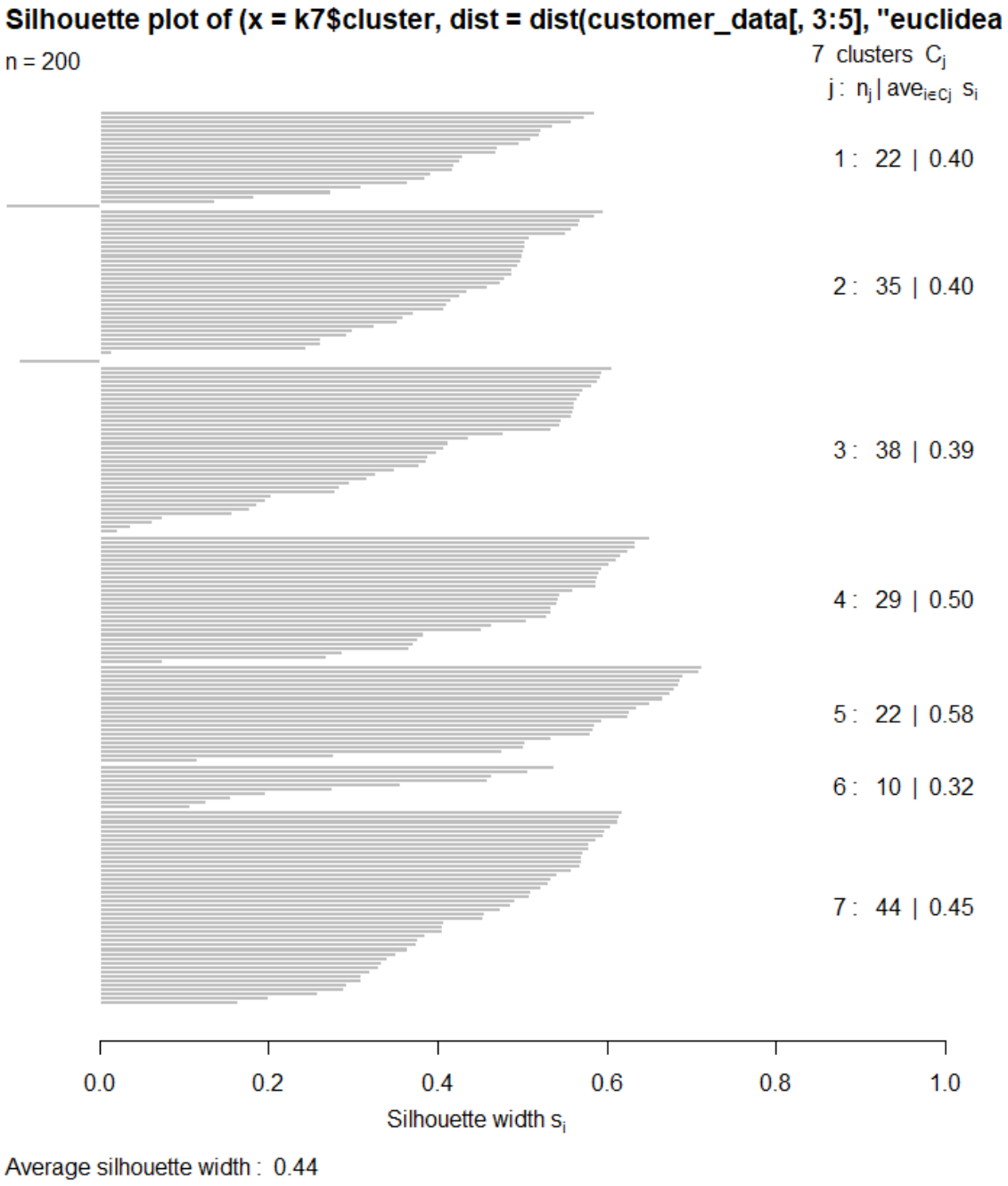
```r
k5<-kmeans(customer_data[,3:5],5,iter.max=100,nstart=50,algorithm="Lloyd")
s5<-plot(silhouette(k5$cluster,dist(customer_data[,3:5],"euclidean")))
```



```r
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")))
```

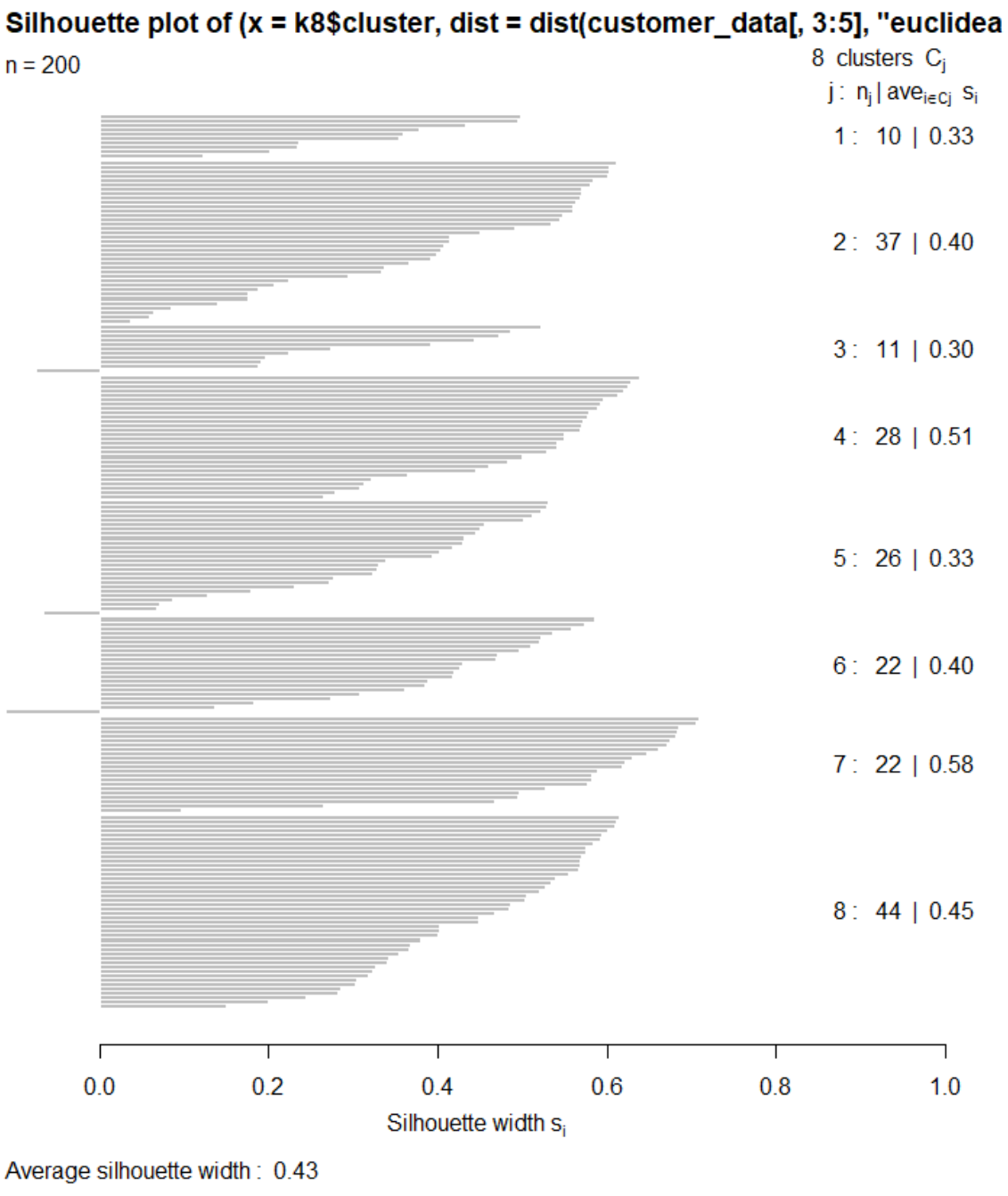


```
k7<-kmeans(customer_data[,3:5],7,iter.max=100,nstart=50,algorithm="Lloyd")
s7<-plot(silhouette(k7$cluster,dist(customer_data[,3:5],"euclidean")))
```

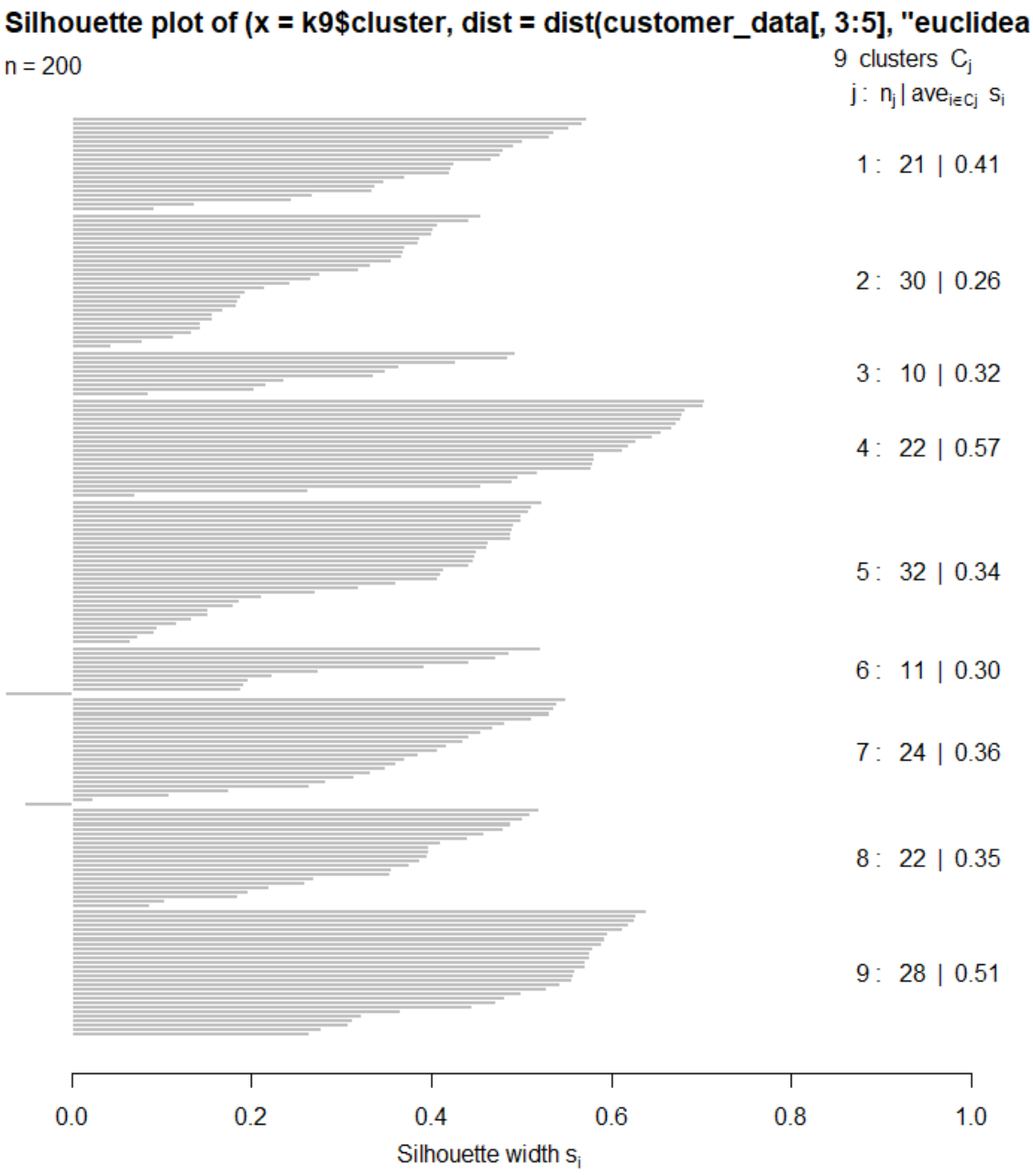


Average silhouette width : 0.44

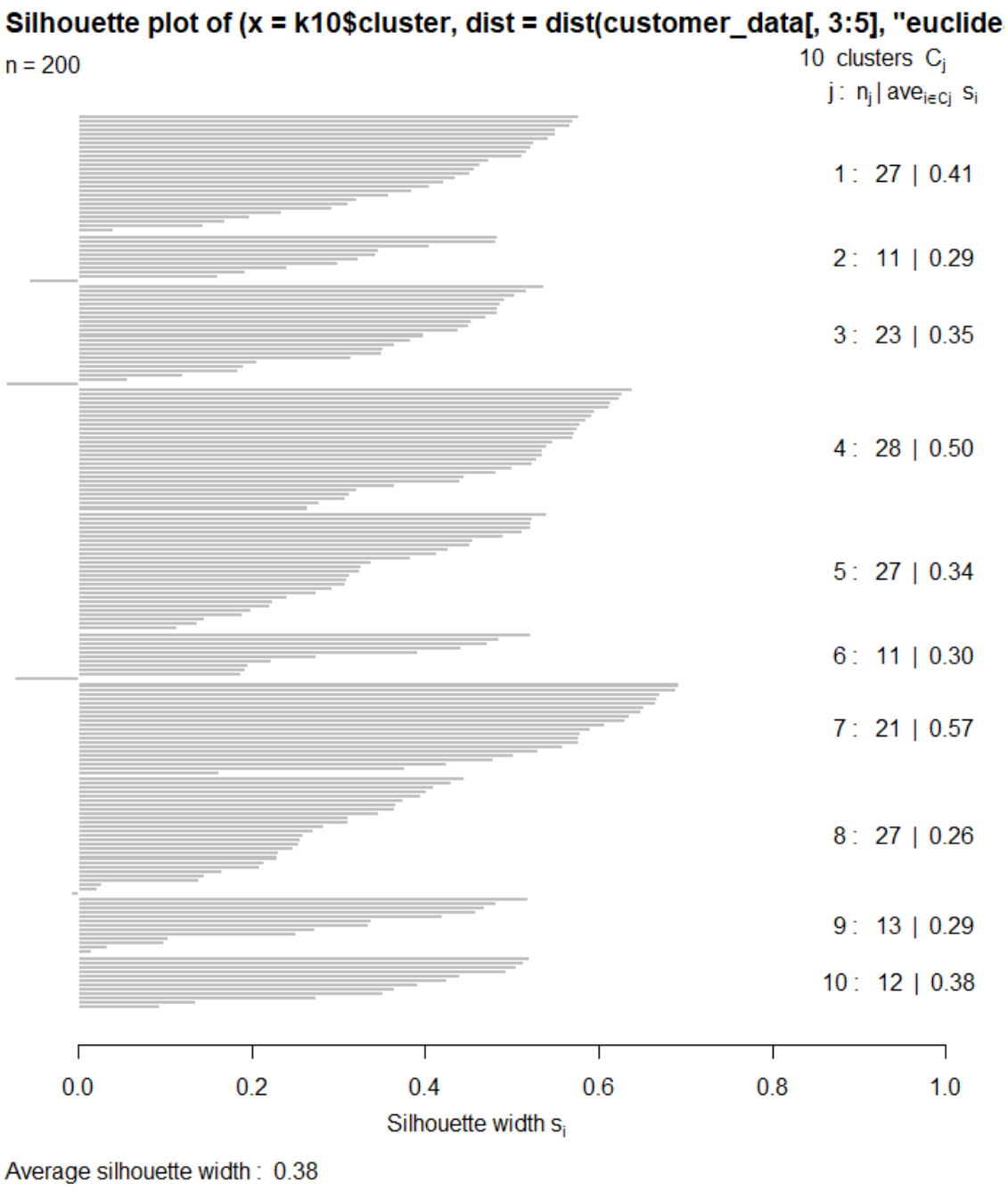
```
k8<-kmeans(customer_data[,3:5],8,iter.max=100,nstart=50,algorithm="Lloyd")
s8<-plot(silhouette(k8$cluster,dist(customer_data[,3:5],"euclidean")))
```



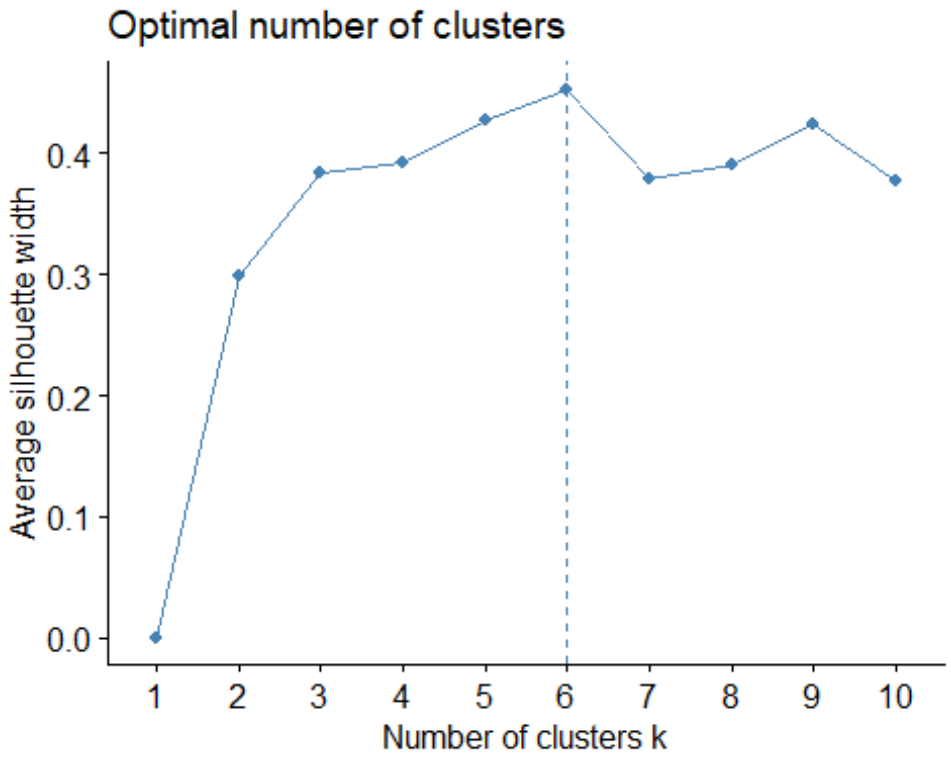
```r
k9<-kmeans(customer_data[,3:5],9,iter.max=100,nstart=50,algorithm="Lloyd")
s9<-plot(silhouette(k9$cluster,dist(customer_data[,3:5],"euclidean")))
```



```
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")))
```

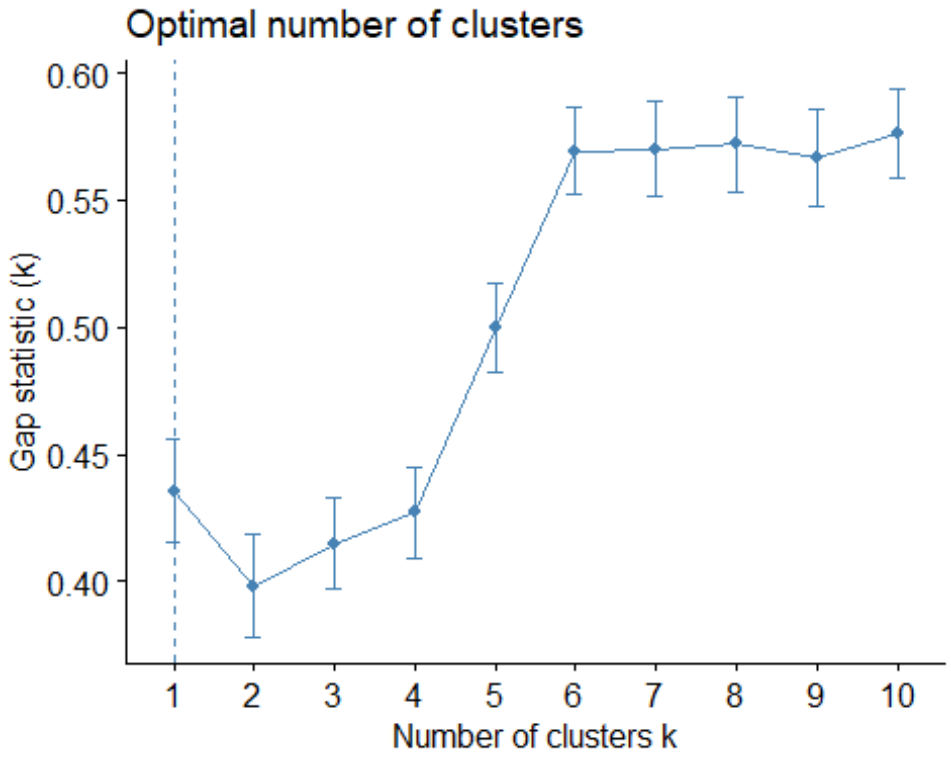


```r
library(NbClust)
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

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

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



- The Gap Statistic Method compares total intracluster variation for different k values with expected values under a null reference distribution. It applies to various clustering methods like K-means and hierarchical clustering, utilizing Monte Carlo simulations to generate sample datasets and calculating ranges between minimum and maximum values for each variable. The clusGap function computes the gap statistic and standard error for a given output.

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



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

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

## K-means clustering with 6 clusters of sizes 35, 22, 38, 44, 22, 39
##
## Cluster means:
##        Age Annual.Income..k.. Spending.Score..1.100.
## 1 41.68571           88.22857                17.28571
## 2 44.31818           25.77273                20.27273
## 3 27.00000           56.65789                49.13158
## 4 56.34091           53.70455                49.38636
## 5 25.27273           25.72727                79.36364
## 6 32.69231           86.53846                82.12821
##
```

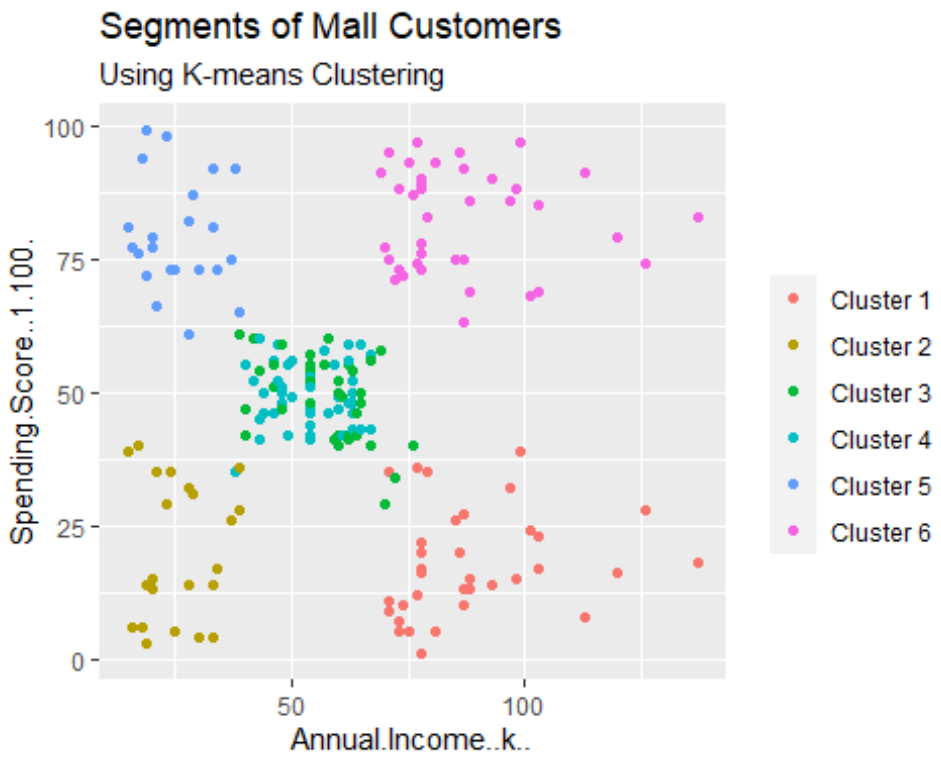
```
## Clustering vector:
##   [1] 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2
2 5 2
##  [38] 5 2 5 4 5 2 3 2 5 4 3 3 3 4 3 3 4 4 4 4 4 3 4 4 3 4 4 4 3 4 4 3 3 4
4 4 4
##  [75] 4 3 4 3 3 4 4 3 4 4 3 4 4 3 3 4 4 3 4 3 3 3 4 3 4 3 3 4 4 3 4 3 4 4
4 4 4
## [112] 3 3 3 3 3 4 4 4 4 3 3 3 6 3 6 1 6 1 6 1 6 3 6 1 6 1 6 1 6 1 6 3 6 1
6 1 6
## [149] 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6
1 6 1
## [186] 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6
##
## Within cluster sum of squares by cluster:
## [1] 16690.857  8189.000  7742.895  7607.477  4099.818 13972.359
##  (between_SS / total_SS =  81.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withi
nss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

In the output of our kmeans operation, we observe key information including:

- cluster: A vector indicating the allocation of each point to a cluster.
- totss: Total sum of squares.
- centers: Matrix of cluster centers.
- withinss: Vector representing intra-cluster sum of squares per cluster.
- tot.withinss: Total intra-cluster sum of squares.
- betweenss: Sum of between-cluster squares.
- size: Total number of points in each cluster.

The clustering results will be visualized using the first two principal components.

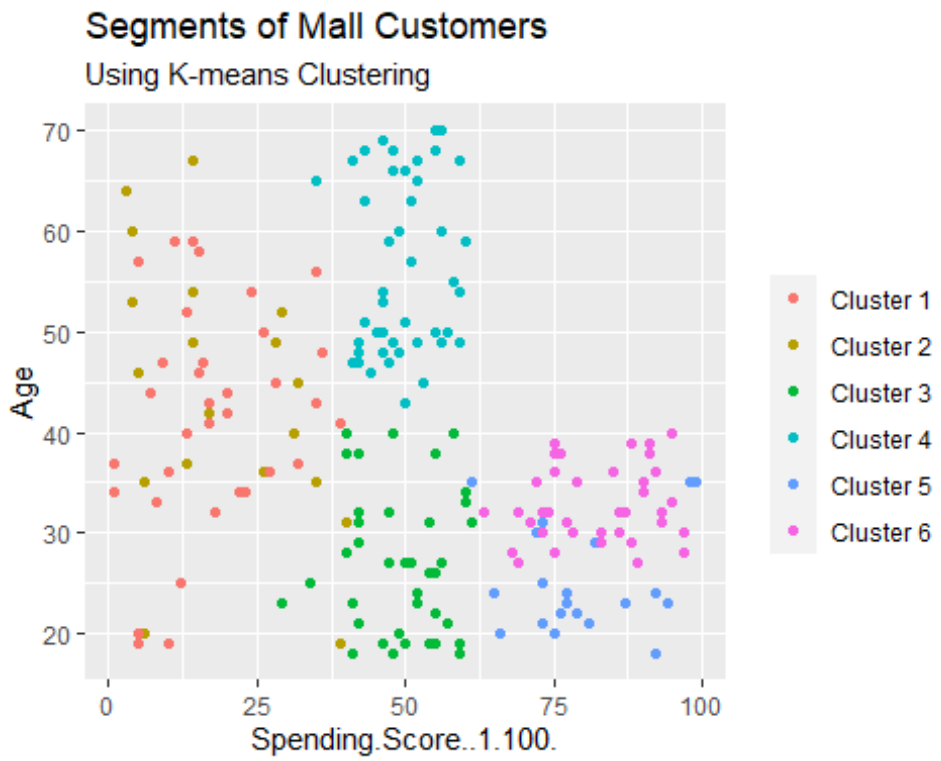
```
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)

## 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

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", "C
luster 5","Cluster 6")) +
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering"
)
```



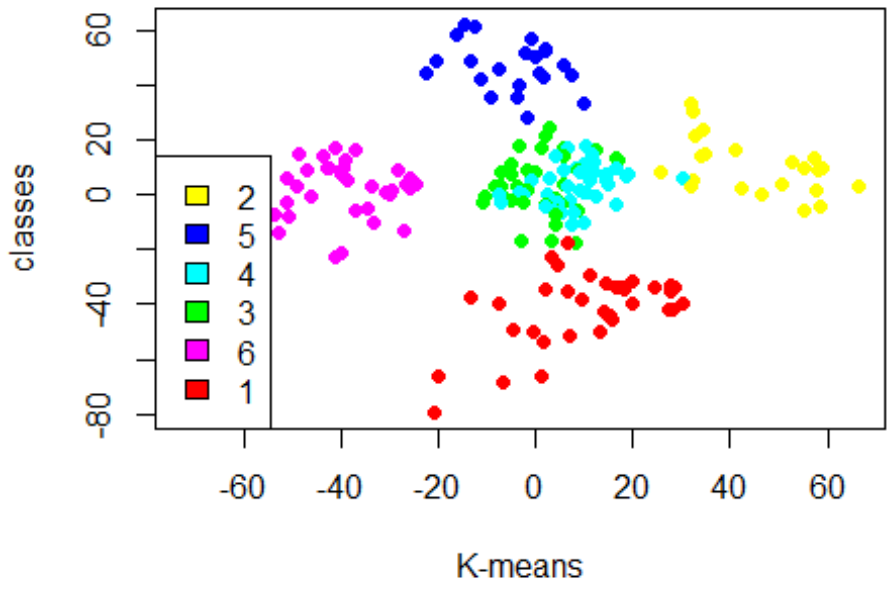
From the above visualization, it is observed that there is a distribution of 6 clusters as follows:

- Clusters 6 and 4: Represent customer_data with medium income salary and medium annual spend.
- Cluster 1: Represents customer data with high annual income and high annual spend.
- Cluster 3: Denotes customer data with low annual income and low yearly spend.
- Cluster 2: Indicates high annual income and low yearly spend.
- Cluster 5: Represents low annual income but high yearly expenditure.

```
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", "Cluste
r 4", "Cluster 5","Cluster 6")) +
  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="cl
asses")
legend("bottomleft",unique(dignm),fill=unique(kCols(digCluster)))
```



- Cluster 4 and 1 consist of customers with medium PCA1 and medium PCA2 scores.
- Cluster 6 represents customers with high PCA2 and low PCA1.
- In Cluster 5, customers have medium PCA1 and low PCA2 scores.
- Cluster 3 comprises customers with high PCA1 and high PCA2.
- Cluster 2 consists of customers with high PCA2 and medium annual spending.

Summary:

This data science project focuses on customer segmentation using unsupervised learning, specifically K-means clustering. Data analysis and visualization were performed before implementing the clustering algorithm.