DATA ANALYTICS (CSE-4027) PROJECT REPORT

TITLE – CUSTOMER SEGMENTATION

Guided By

Dr. Gopikrishnan S

TEAM MEMBERS

MANEESH MADALA
(19BCE7192)
PATRI LALITHYA MANASA
(19BCD7013)
AKELLA SIVA SAI ATCHYUT
(19BCE7513)
JATHIN KOLLA
(19BCD7192)

TABLE OF CONTENTS

INTRODUCTION

- 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.
- 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.

AIM

- 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.

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.

LITERATURE REVIEW

Ggplot2 is now over 10 years old and is used by 100's of 1000's of people to make millions of plots. It is an R package dedicated to data visualization. It can greatly enhance the quality and aesthetics of your graphics, and will make you much more efficient in creating them. Ggplot2 allows building almost any type of chart. It is a system for declarative creating graphics, based on the grammar of graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

DATASET

You are owing a supermarket mall and through membership cards , you have some basic data about your customers like Customer ID, age, gender, annual income and spending score.

Spending Score is something you assign to the customer based on your defined parameters like customer behavior and purchasing data.

DATA PREPROCESSING AND DATA ANALYSIS

Code:
customer_data=
read.csv("C:/Users/LENOVO/OneDrive/Documents/customer-segmentation-dataset/Mall_Customers.csv")
str(customer_data)
names(customer_data)

OUTPUT:

```
> customer_data=read.csv("C:/Users/LENOVO/OneDrive/Documents/customer-segmentation-dataset/Mall_Customers.csv")
> str(customer_data)
'data.frame':   1500 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"                    "Annual.Income..k.."
[5] "Spending.Score..1.100."
```

Code:
head(customer_data)
summary(customer_data$Age)

OUTPUT:

```
> 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   25.00   26.00   27.73   28.00   70.00 
> 
```

Code:
sd(customer_data$Age)
summary(customer_data$Annual.Income..k..)
sd(customer_data$Annual.Income..k..)

summary(customer_data$Age)

OUTPUT:

```
> summary(customer_data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  18.00   25.00   26.00   27.73   28.00   70.00 
> sd(customer_data$Age)
[1] 6.827445
> summary(customer_data$Annual.Income..k..)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   15.0   375.8   737.5   738.4  1122.2  1500.0 
> sd(customer_data$Annual.Income..k..)
[1] 445.4754
> summary(customer_data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  18.00   25.00   26.00   27.73   28.00   70.00 
```

Code:
sd(customer_data$Spending.Score..1.100.)

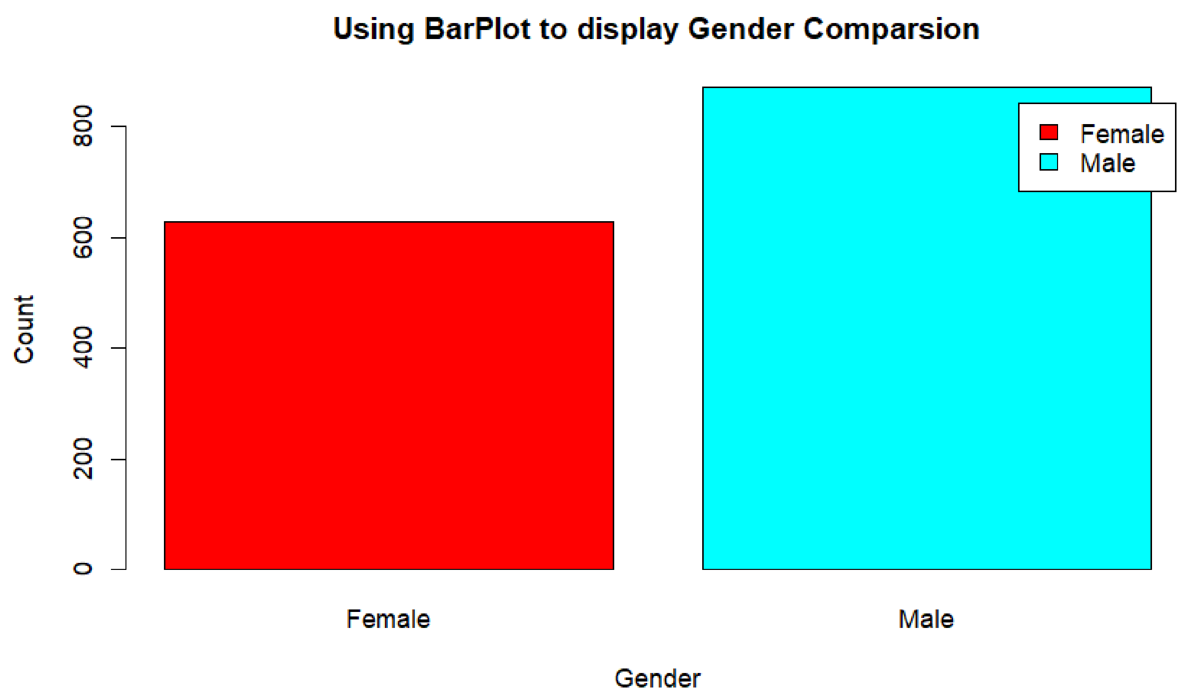
OUTPUT:

```
> sd(customer_data$Spending.Score..1.100.)
[1] 12.60675
> 
```

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

OUTPUT:

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

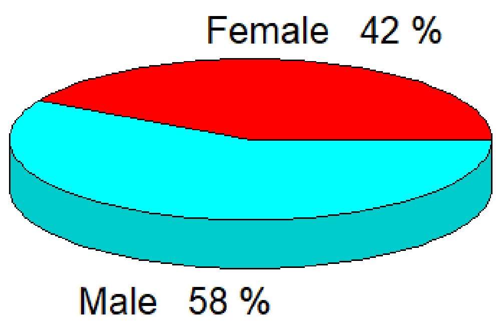


pct=round(a/sum(a)*100)
lbs=paste(c("Female","Male")," ",pct,"%",sep=" ")
library(plotrix)
pie3D(a,labels=lbs,main="Pie Chart Depicting Ratio of Female and Male")

OUTPUT:

```
> 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



Code:
summary(customer_data$Age)

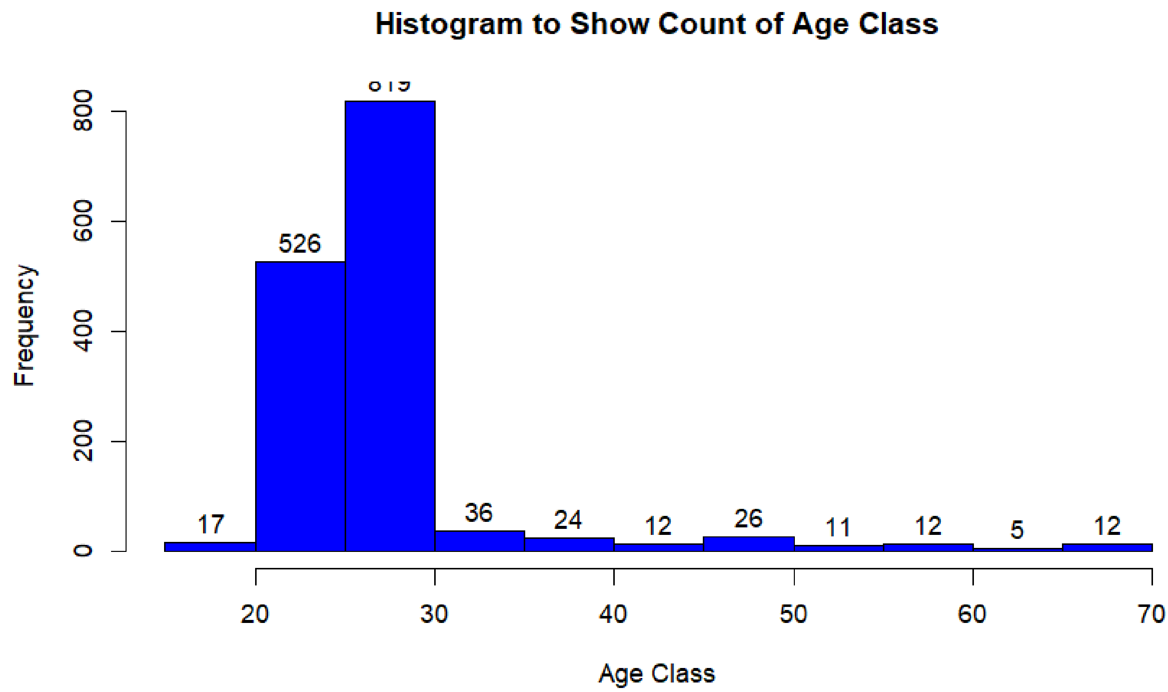
OUTPUT:

```
> summary(customer_data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  18.00   25.00   26.00   27.73   28.00   70.00 
> 
```

CODE:
hist(customer_data$Age,col="blue",main="Histogram to Show Count of Age Class",xlab="Age Class",ylab="Frequency",labels=TRUE)

OUTPUT:

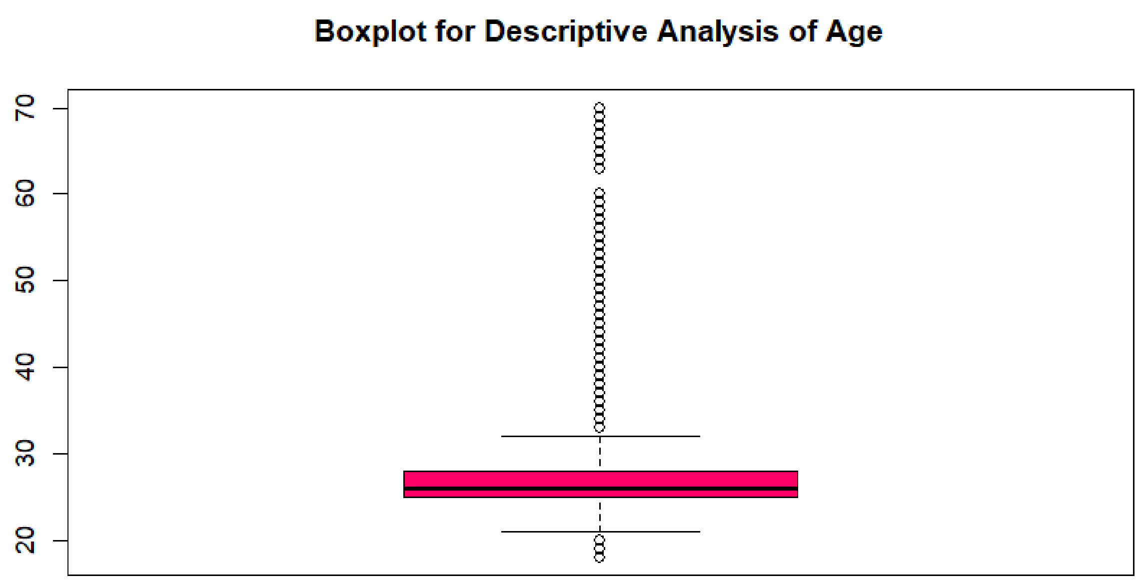
```
> hist(customer_data$Age,col="blue",main="Histogram to Show Count of Age Class",xlab="Age Class",ylab="Frequenc
y",labels=TRUE)
> 
```



CODE:

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

OUTPUT:

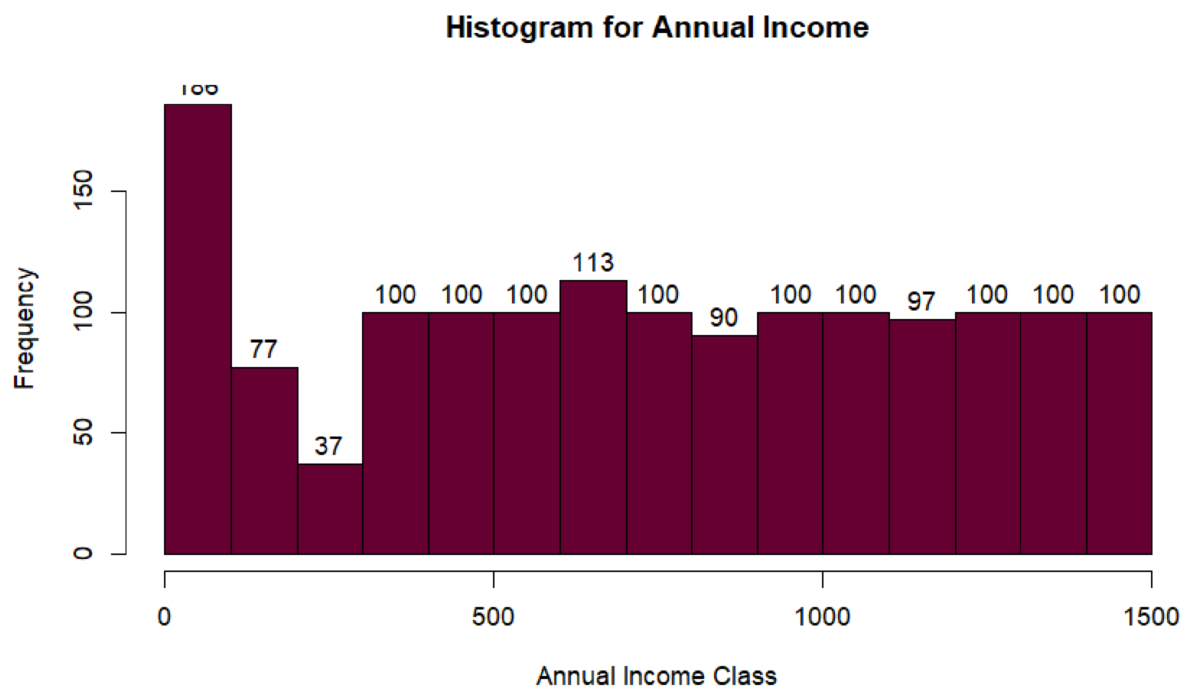


CODE:

```
summary(customer_data$Annual.Income..k..)
hist(customer_data$Annual.Income..k..,
    col="#660033",
    main="Histogram for Annual Income",
    xlab="Annual Income Class",
    ylab="Frequency",
    labels=TRUE)
```

OUTPUT:

```
> summary(customer_data$Annual.Income..k..)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   15.0   375.8   737.5   738.4  1122.2  1500.0 
> hist(customer_data$Annual.Income..k..,
+      col="#660033",
+      main="Histogram for Annual Income",
+      xlab="Annual Income Class",
+      ylab="Frequency",
+      labels=TRUE)
```



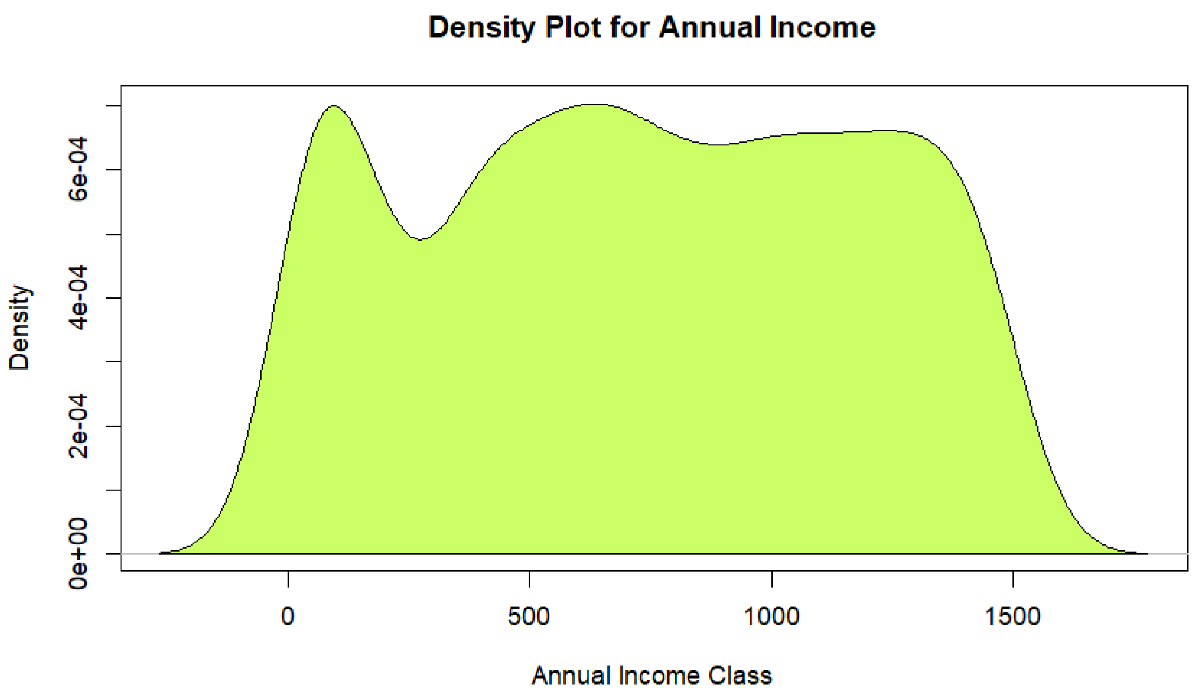
CODE:

```
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")

OUTPUT:

```
> 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")
> |
```



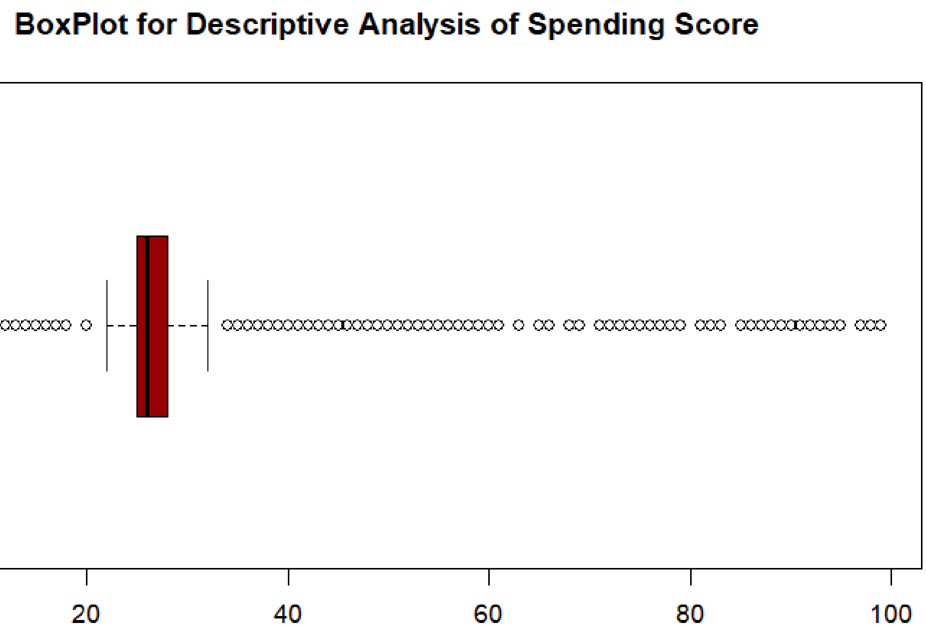
CODE:

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

```
boxplot(customer_data$Spending.Score..1.100.,
        horizontal=TRUE,
        col="#990000",
        main="BoxPlot for Descriptive Analysis of Spending Score")
```

OUTPUT:

```
> summary(customer_data$Spending.Score..1.100.)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00   25.00   26.00   29.57   28.00   99.00 
> 
> boxplot(customer_data$Spending.Score..1.100.,
+         horizontal=TRUE,
+         col="#990000",
+         main="BoxPlot for Descriptive Analysis of Spending Score")
> 
```

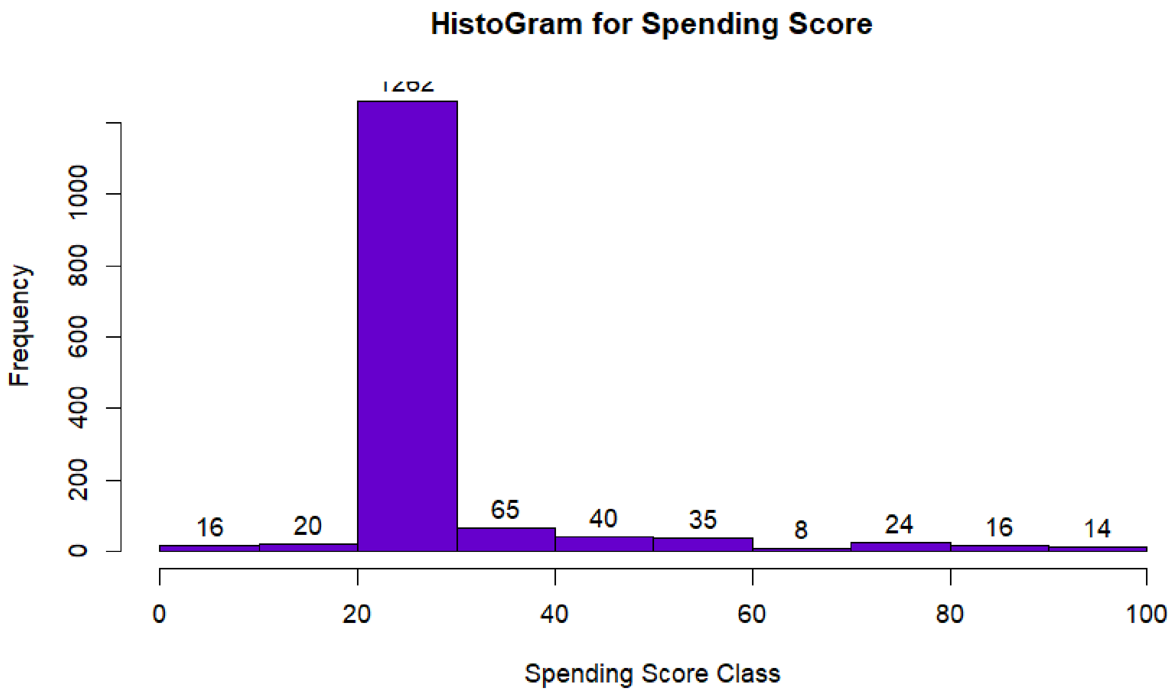


CODE:

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

OUTPUT:

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

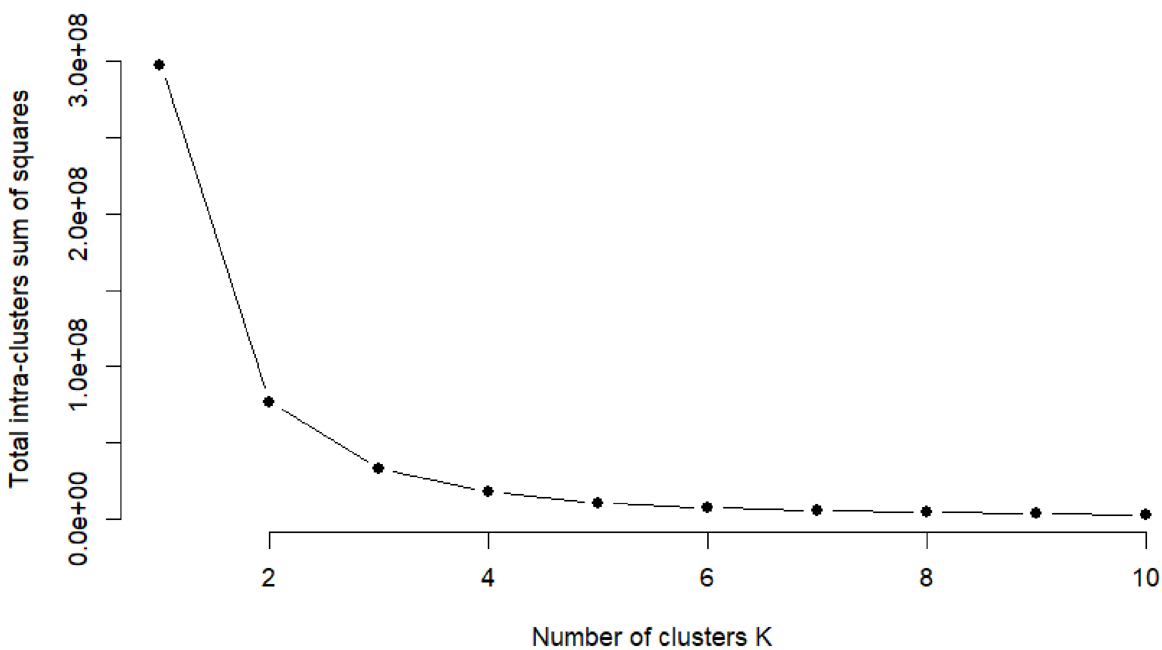


COde:

```
##Elbow Method
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")
```

OUTPUT:

```
> ##Elbow Method
> 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)
There were 29 warnings (use warnings() to see them)
>
> plot(k.values, iss_values,
+      type="b", pch = 19, frame = FALSE,
+      xlab="Number of clusters K",
+      ylab="Total intra-clusters sum of squares")
>
```

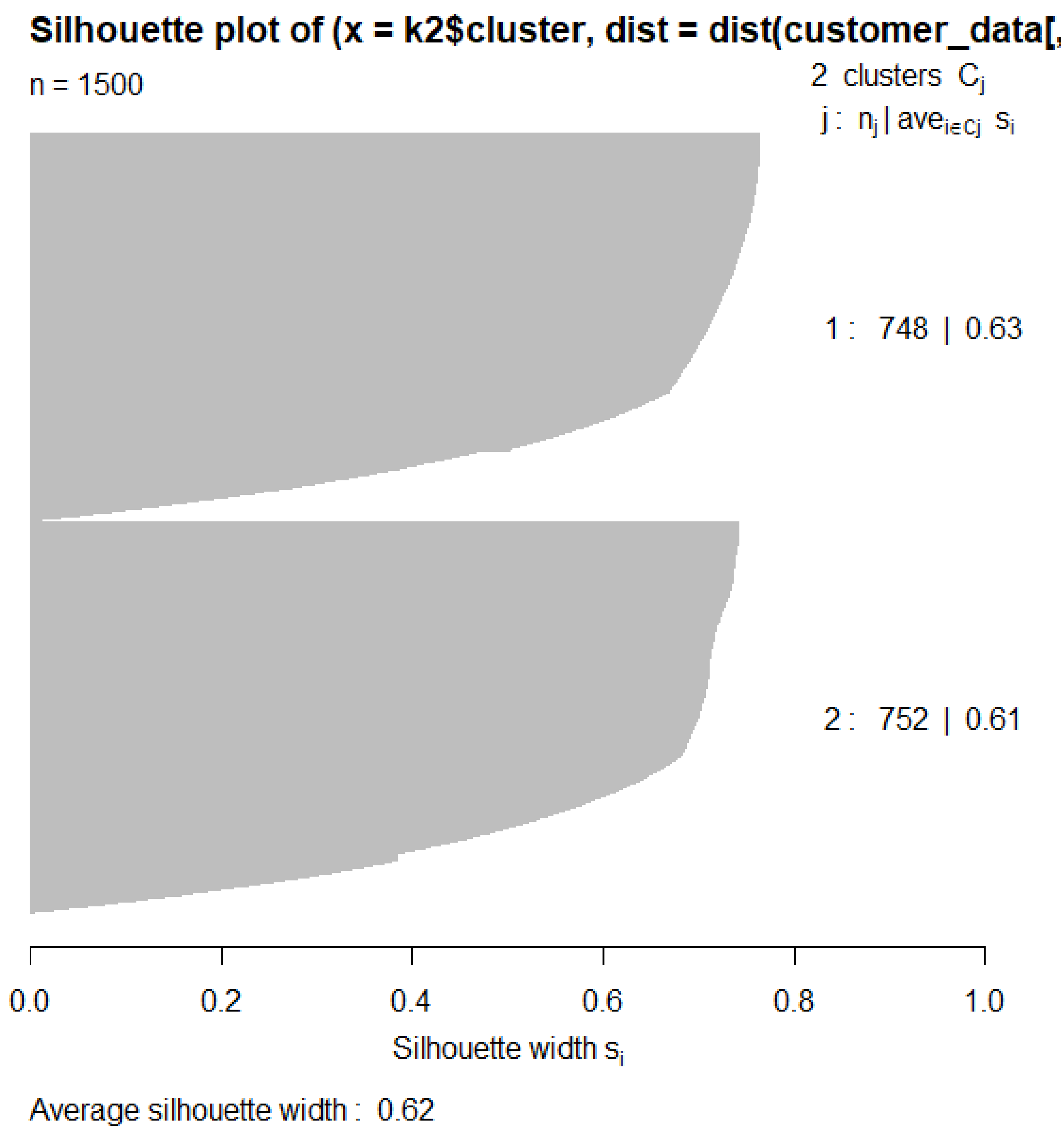


CODE:
##Average Silhouette Method
library(cluster)
library(gridExtra)
library(grid)

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")))

OUTPUT:

```
> library(cluster)
> library(gridExtra)
> 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")))
>
>
>
>
```

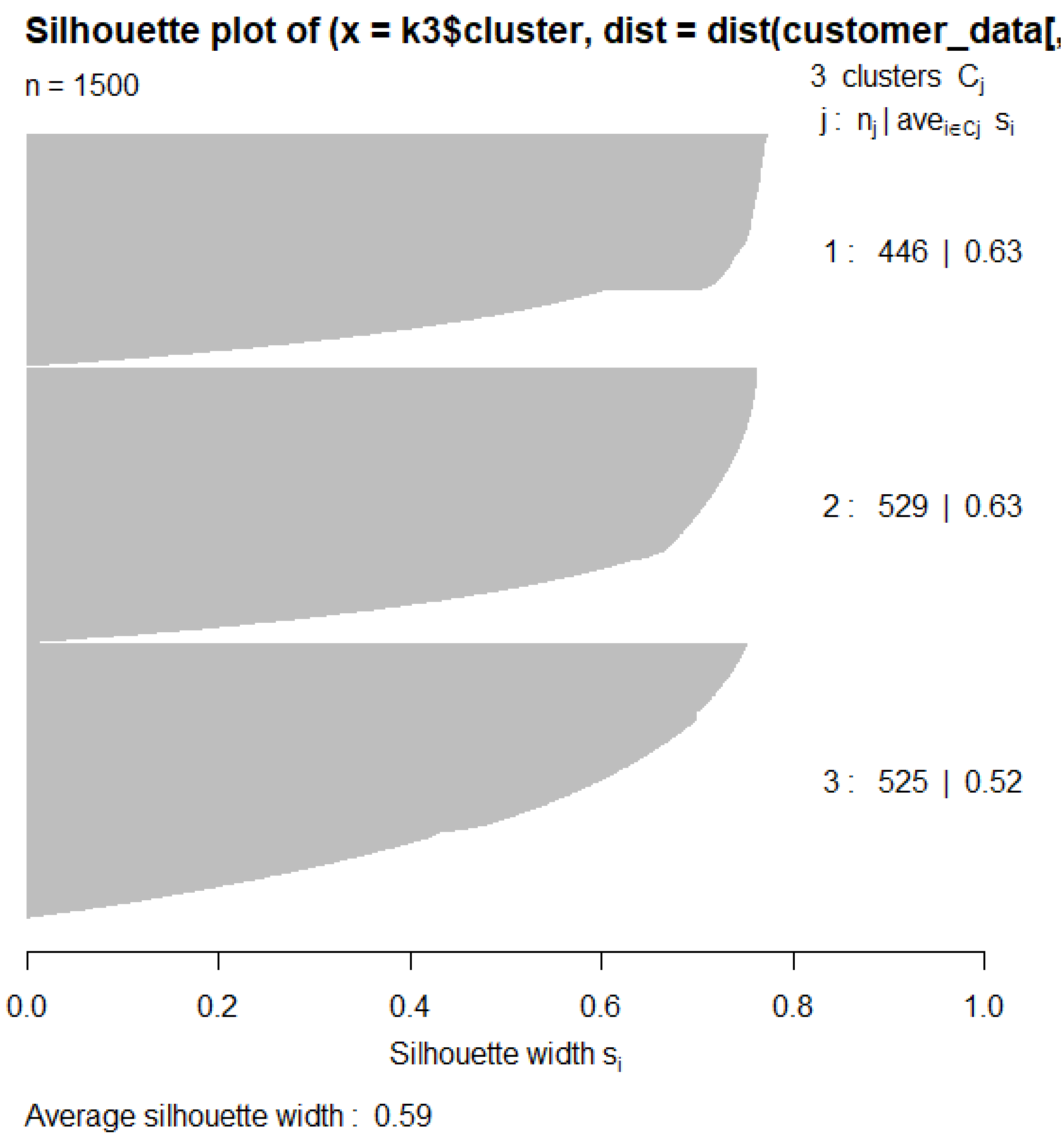


CODE:

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")))

OUTPUT:

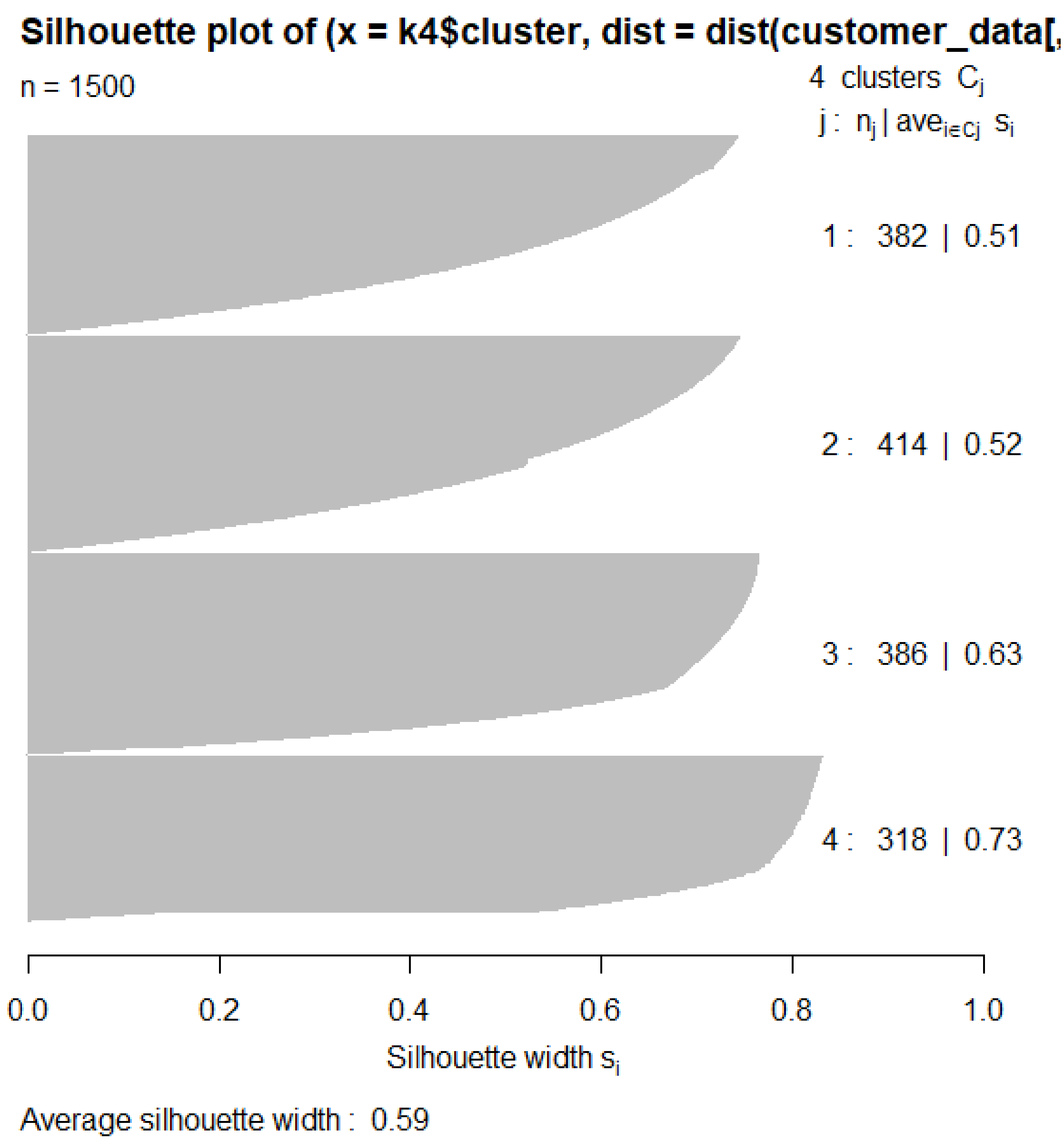
```
>
> k3<-kmeans(customer_data[,3:5],3,iter.max=100,nstart=50,algorithm="Lloyd")
> s3<-plot(silhouette(k3$cluster,dist(customer_data[,3:5],"euclidean")))
> |
```



Code:
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")))

OUTPUT:

```
>
> 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")))
```

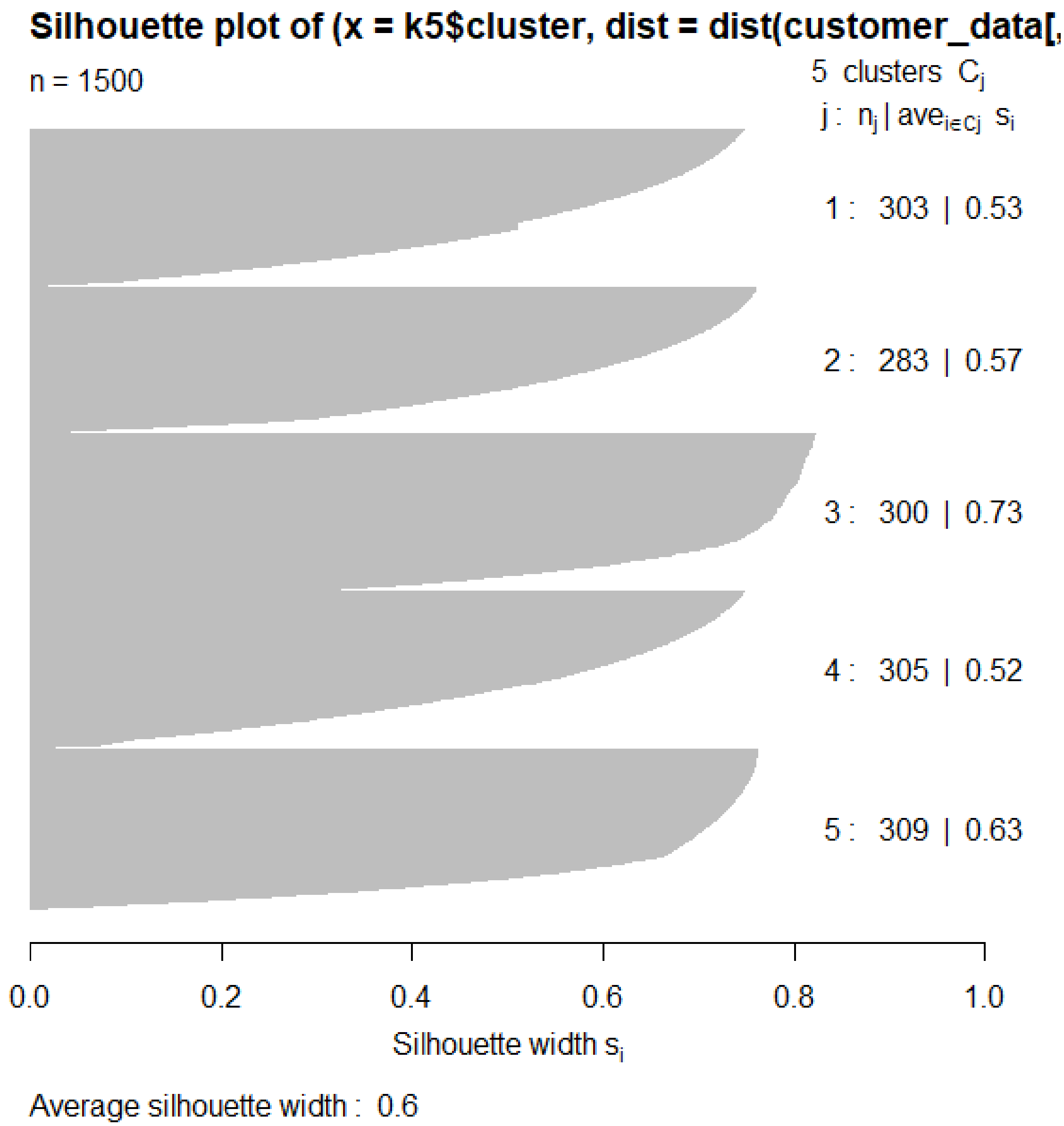


CODE:

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")))

OUTPUT:

```
> 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")))
```



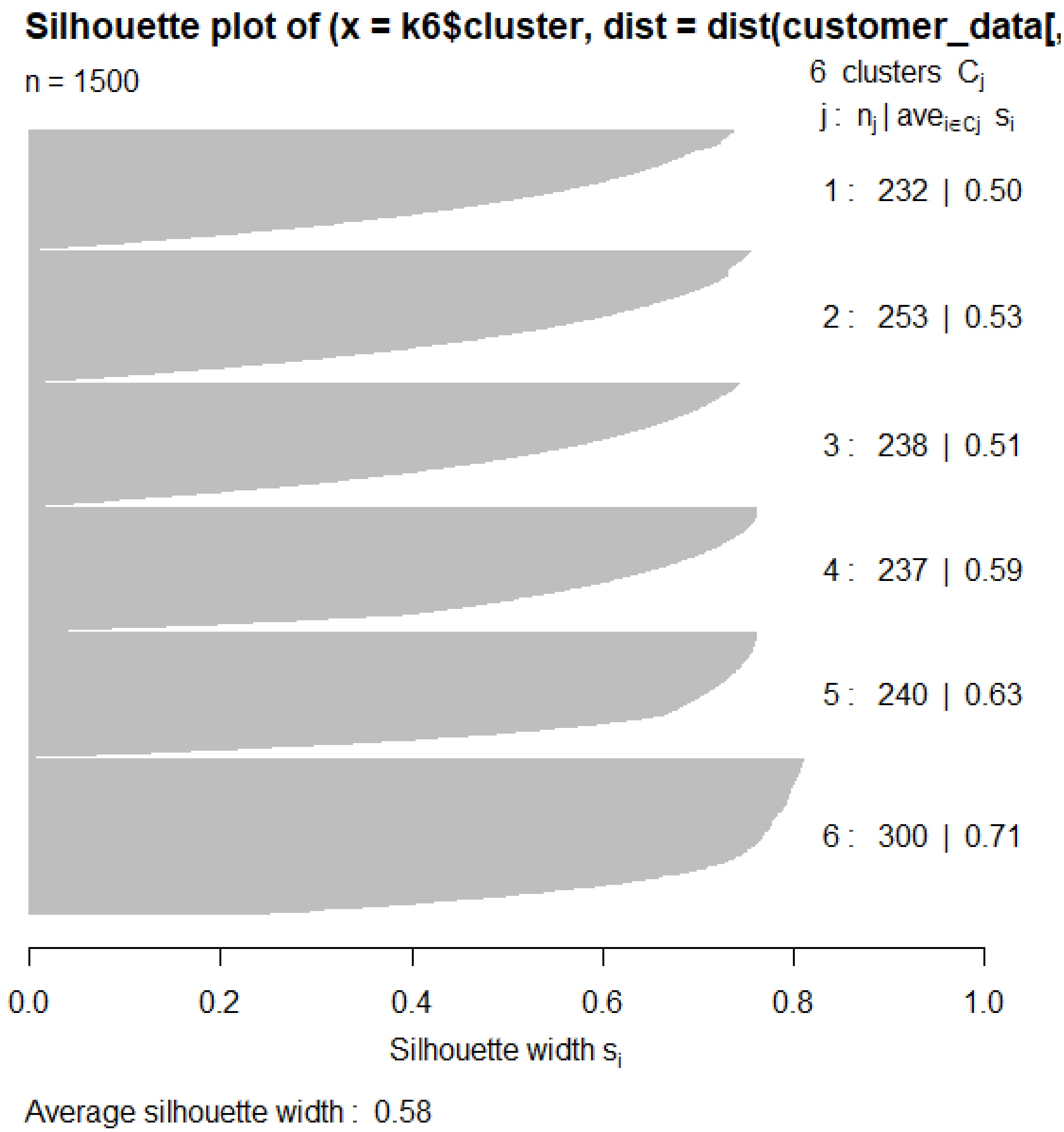
CODE:

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")))

OUTPUT:

```
>
> 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")))
>
```

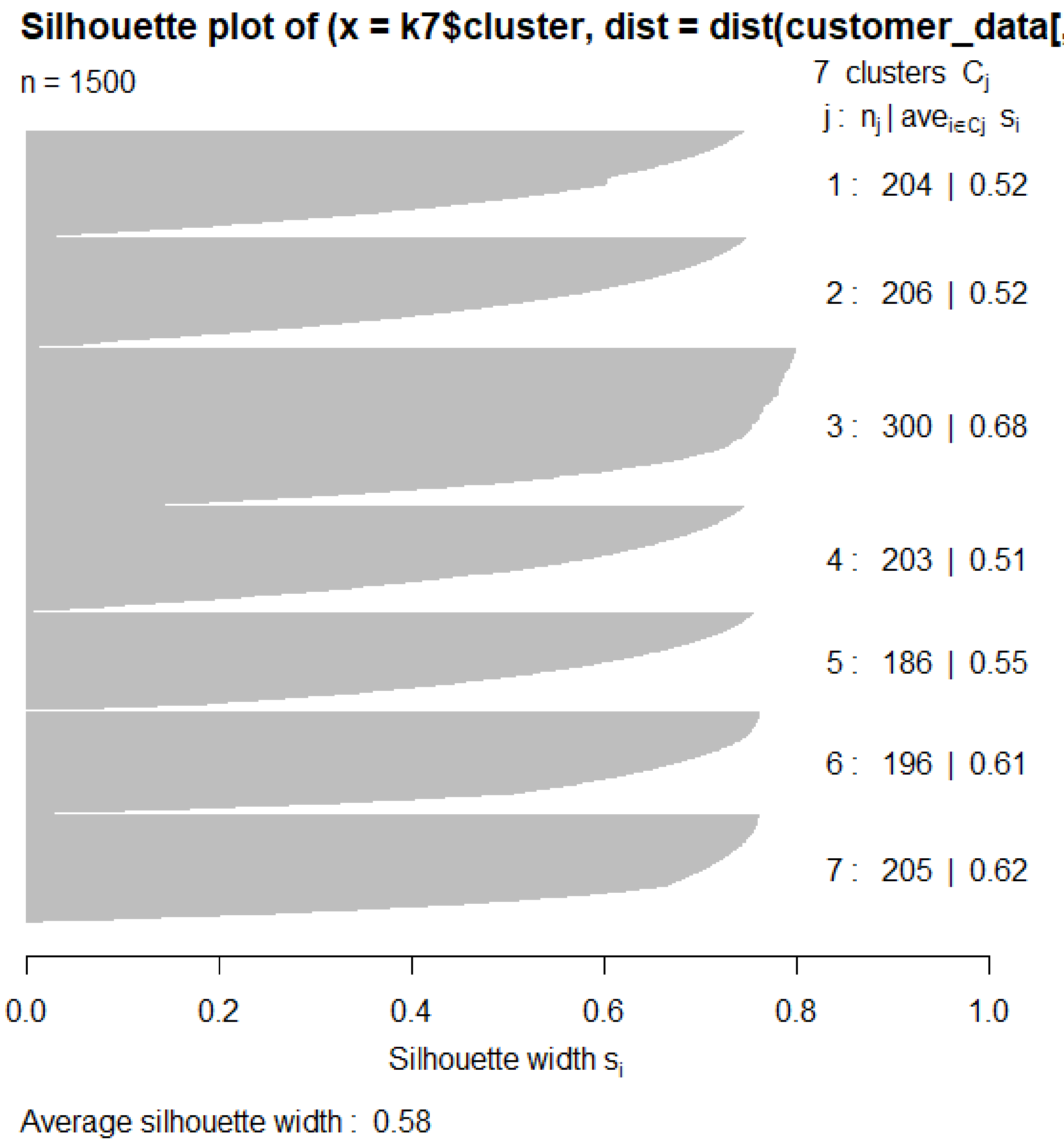


CODE:

```
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")))
```

OUTPUT:

```
> 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")))
> 
```

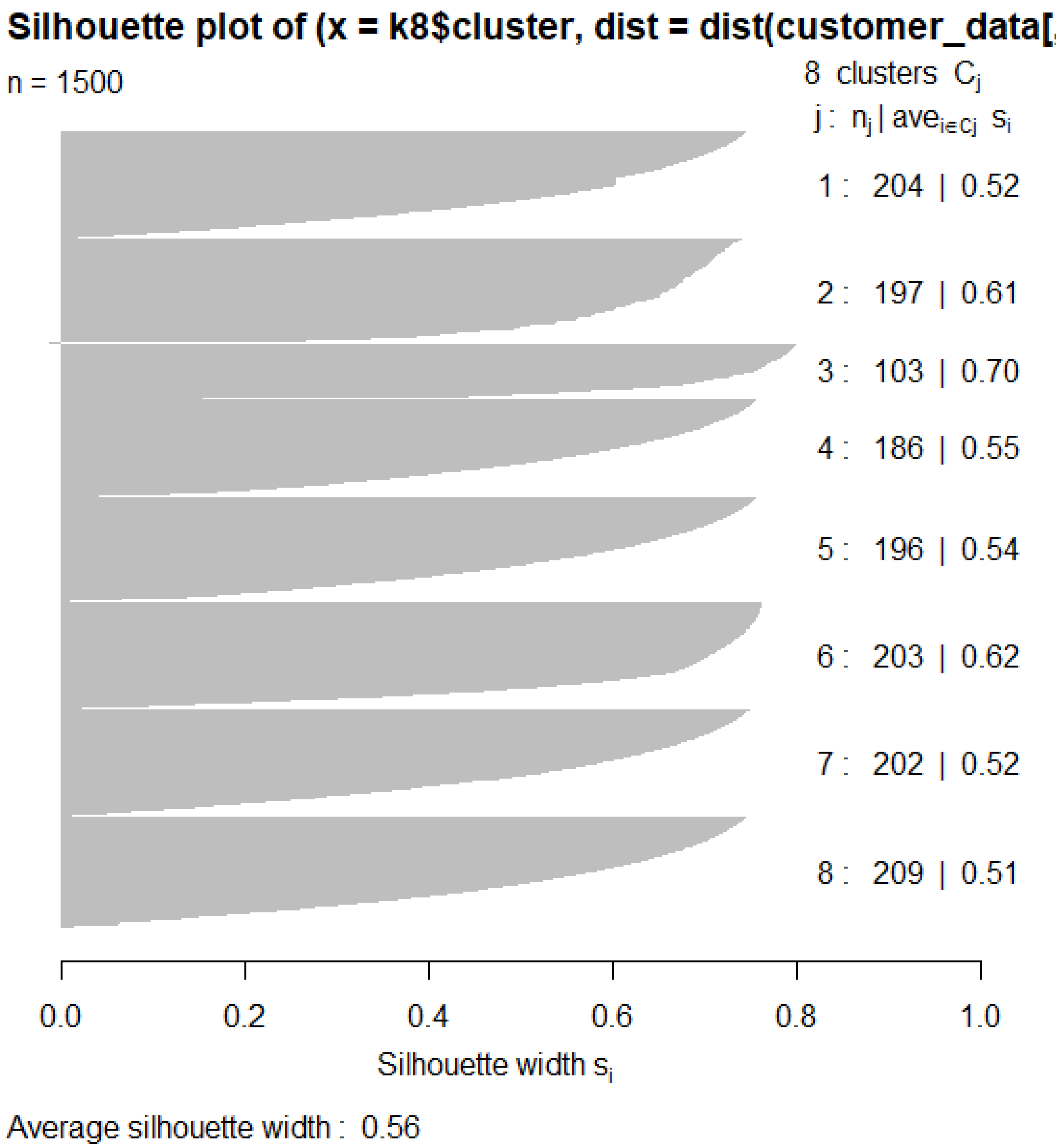


CODE:

```
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")))
```

OUTPUT:

```
> 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")))
```

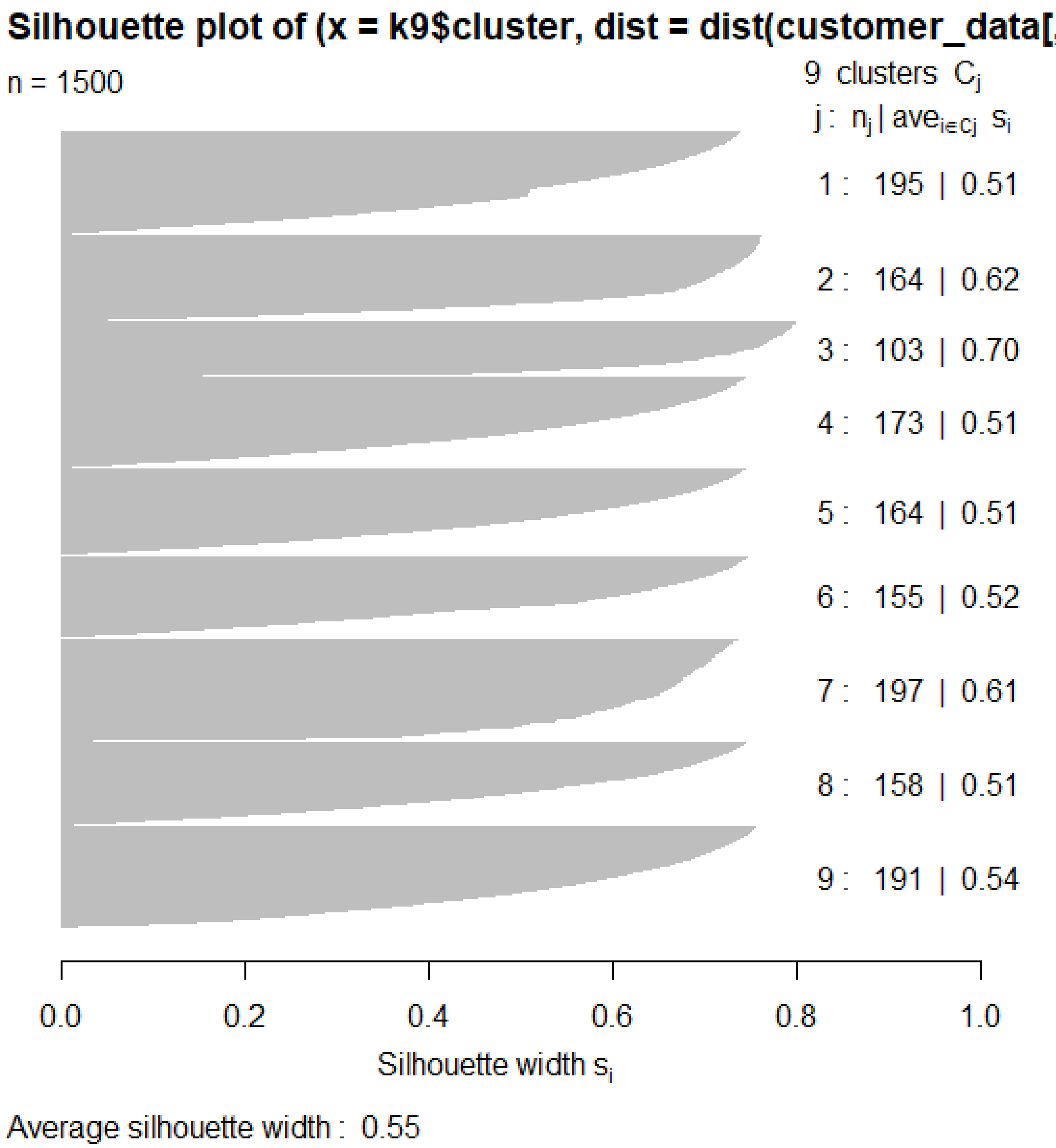


CODE:

```
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")))
```

OUTPUT:

```
> 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")))
```

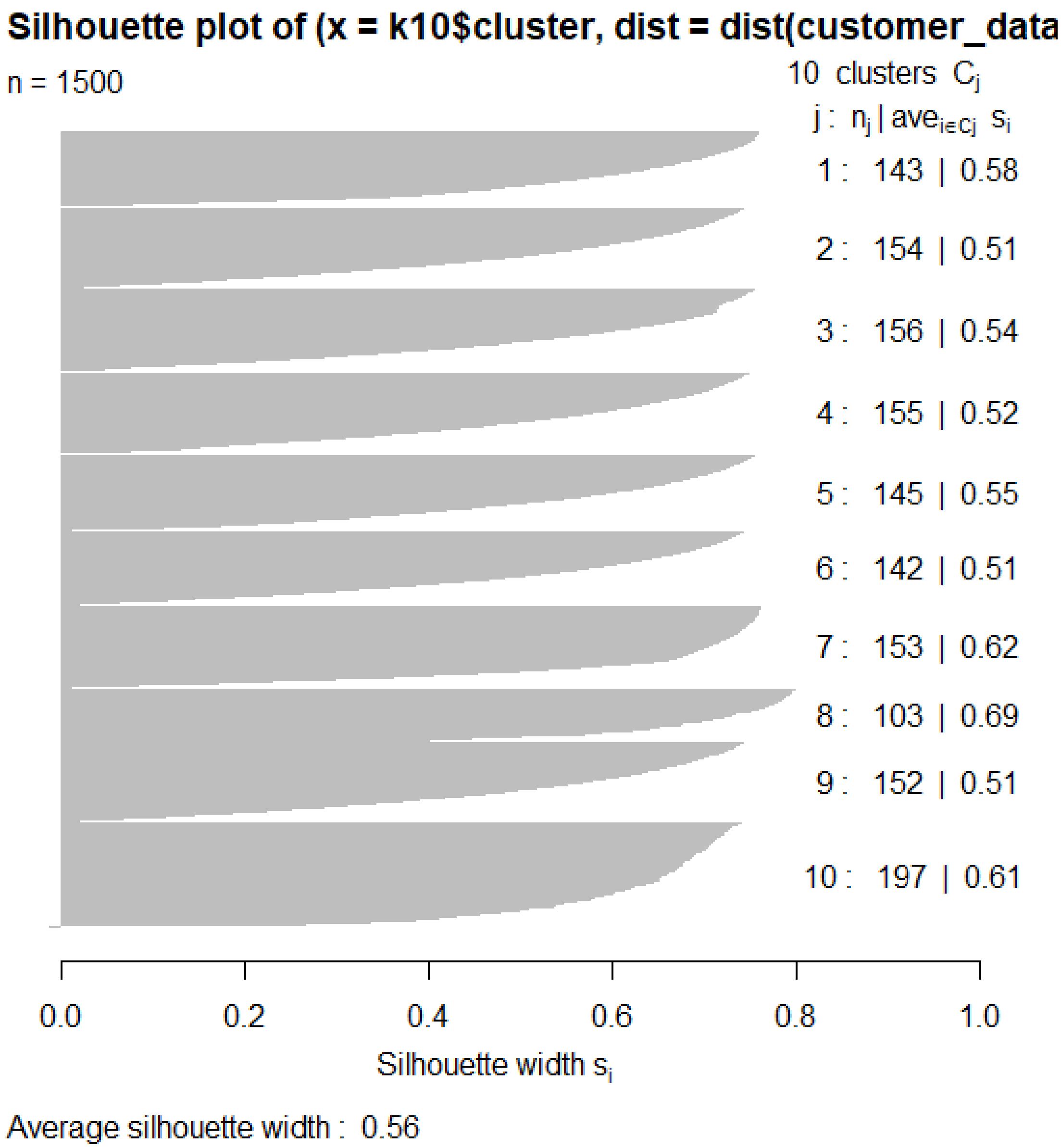


CODE:

```
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")))
```

OUTPUT:

```
> 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")))
```

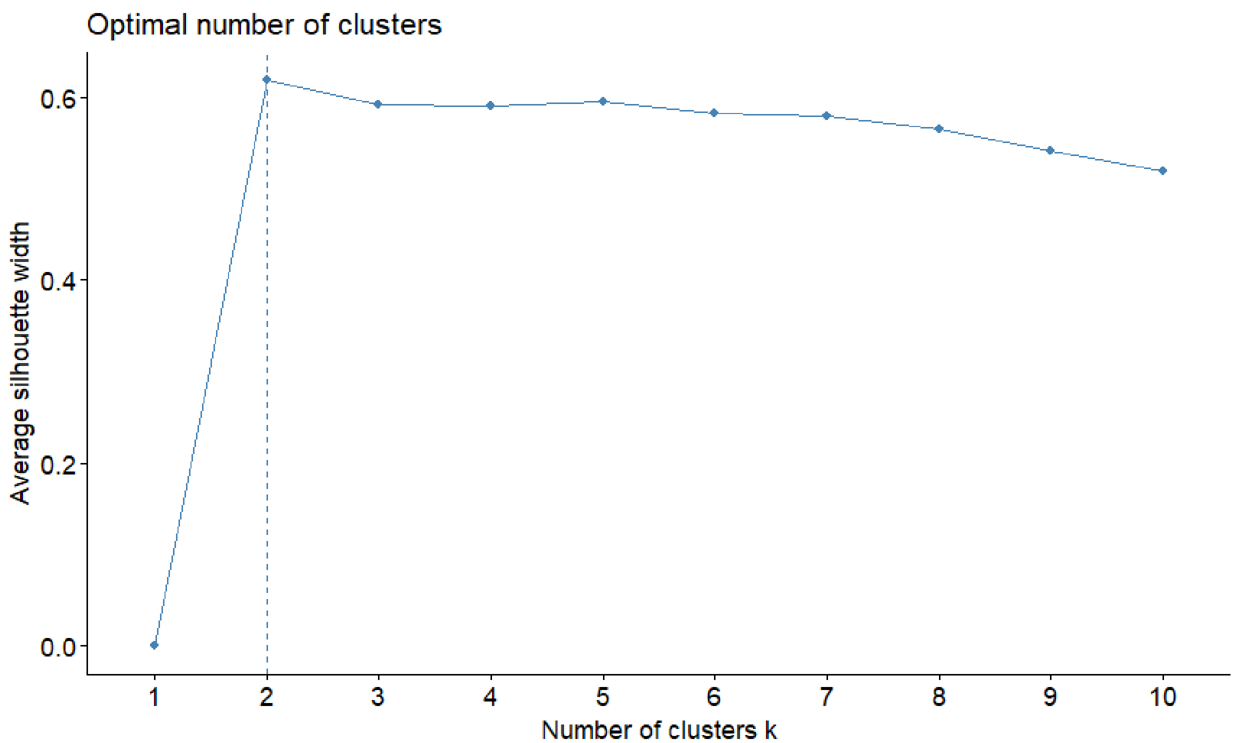


CODE:

```
##Determine and visualize the optimal number of clusters
library(NbClust)
library(factoextra)

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

OUTPUT:

```
> ##Determine and visualize the optimal number of clusters
> library(NbClust)
> library(factoextra)
Loading required package: ggplot2
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
>
> fviz_nbclust(customer_data[,3:5], kmeans, method = "silhouette")
```



CODE:

#Gap Statistic Method

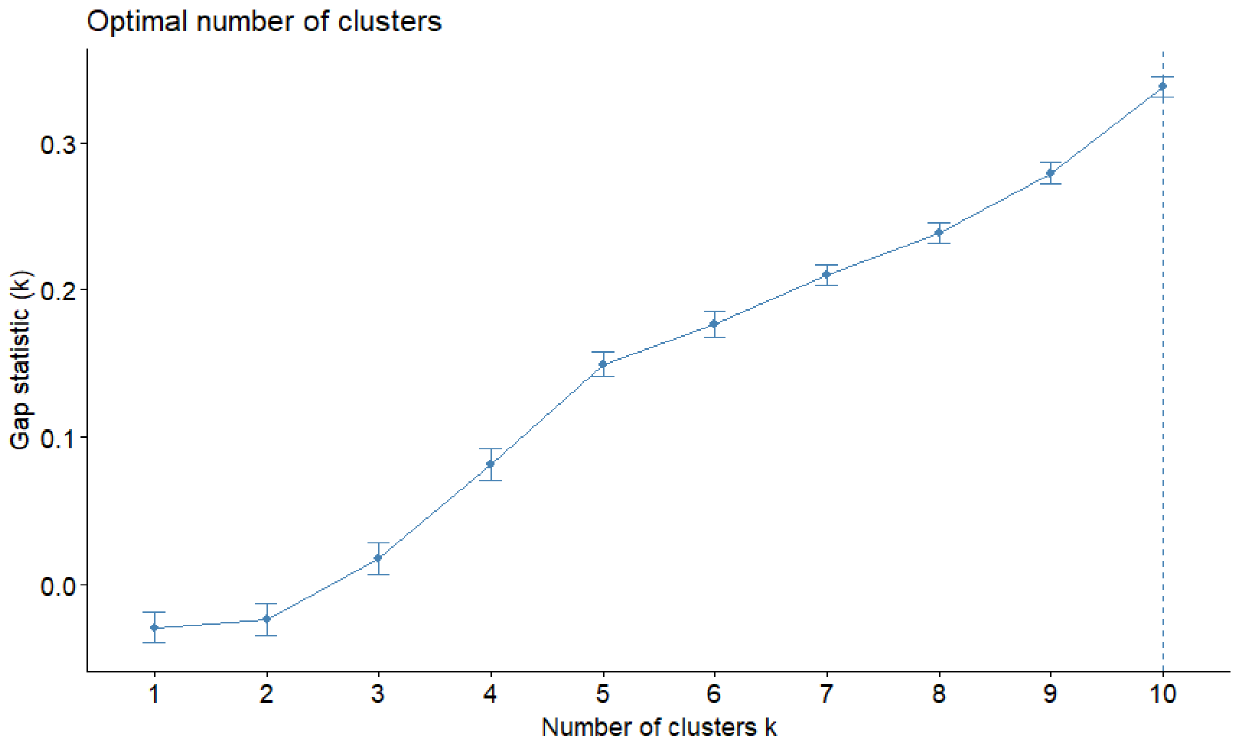
set.seed(125)

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

fviz_gap_stat(stat_gap)

OUTPUT:

```
> #Gap Statistic Method
> set.seed(125)
> stat_gap <- clusGap(customer_data[,3:5], FUN = kmeans, nstart = 25,K.max = 10, B = 50)
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
.................................................. 50
> fviz_gap_stat(stat_gap)
```



CODE:

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

OUTPUT:

```
> k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
> k6
K-means clustering with 6 clusters of sizes 255, 238, 241, 300, 231, 235

Cluster means:
       Age Annual.Income..k.. Spending.Score..1.100.
1 26.06275           661.3020               25.98824
2 26.03361          1381.5000               26.06723
3 25.91286           421.0000               26.11203
4 34.61000           102.8733               43.70333
5 25.97403           905.1429               25.96104
6 26.09362          1143.6979               26.03830

Clustering vector:
   [1] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
  [52] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 [103] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 [154] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 [205] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 [256] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3
 [307] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [358] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [409] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [460] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [511] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [562] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [613] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [664] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [715] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [766] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 [817] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 [868] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 [919] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 [970] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 [ reached getOption("max.print") -- omitted 500 entries ]

Within cluster sum of squares by cluster:
[1] 1188774 1124370 1167401 1505605 1160133 1123628
 (between_SS / total_SS =  97.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
>
```

CODE:

##Visualizing the Clustering Results using the First Two Principle Components
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)

pcclust$rotation[,1:2]

OUTPUT:

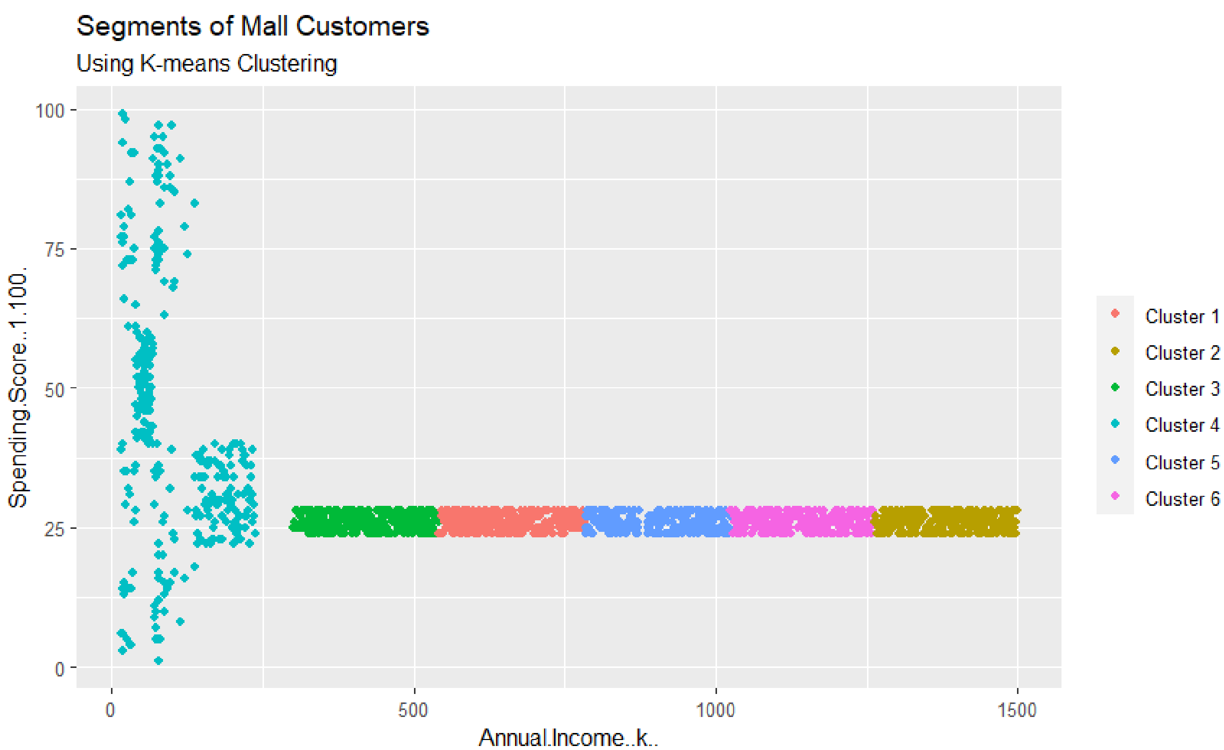
```
> ##Visualizing the Clustering Results using the First Two Principle Components
> pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
> summary(pcclust)
Importance of components:
                            PC1      PC2    PC3
Standard deviation     445.5145 11.45433 6.2846
Proportion of Variance   0.9991  0.00066 0.0002
Cumulative Proportion    0.9991  0.99980 1.0000
>
> pcclust$rotation[,1:2]
                              PC1         PC2
Age                    -0.00582055 -0.06556685
Annual.Income..k..      0.99991216 -0.01226356
Spending.Score..1.100. -0.01190737 -0.99777282
```

CODE:

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

OUTPUT:

```
> 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", "Clust
er 3", "Cluster 4", "Cluster 5","Cluster 6")) +
+   ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
> |
```

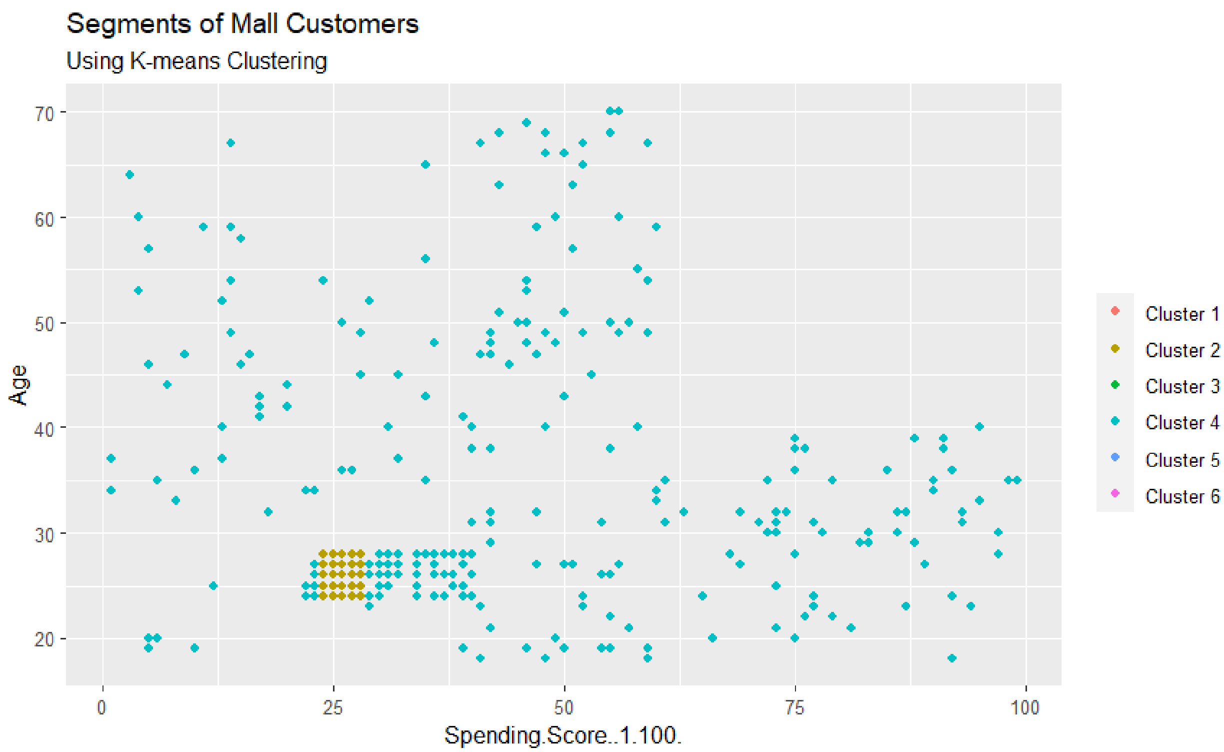


CODE:

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

OUTPUT:

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



CODE:

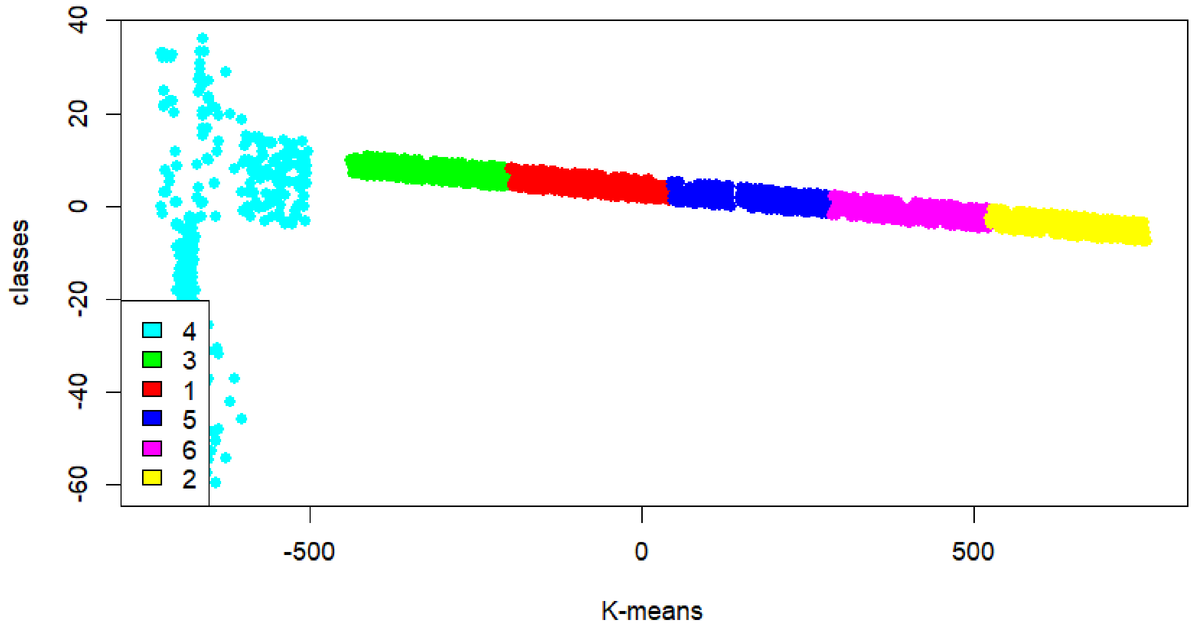
kCols=function(vec){cols=rainbow (length (unique (vec)))
return (cols[as.numeric(as.factor(vec))])}

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)))

OUTPUT:

```
> 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)))
>
```



FUTURE SCOPE

Customer segmentation is the process of grouping customers together based on common characteristics. These customer groups are beneficial in marketing campaigns, in identifying potentially profitable customers, and in developing customer loyalty. Common types of customer segmentation include: Demographic segmentation

CONCLUSION

To be effective, we must prepare and plan for the various challenges and hurdles that each step may present, and always make sure to adapt your process to any new information or feedback that might change its output. Additionally, we cannot force feed this process on your business. If the key stakeholders that will be impacted by the best current customers segmentation process do not fully buy-in, then the outputs produced from it will be relatively meaningless. If you properly manage the best current customer segmentation process, however, the impact it can have on every part of your organization — sales, marketing, product development, customer service, etc. — is immense. Your business will possess stronger customer focus and market clarity, allowing it to scale in a far more predictable and efficient manner. Ultimately, that means no longer needing to take on every customer that is willing to pay for your product or service, which will allow you to instead hone in on a specific subset of customers that present the most profitable opportunities and efficient use of resources. That is critical for every business, of course, but at the expansion stage, it can often be the difference between incredible success and certain failure.

REFERENCES

1) https://data-flair.training/blogs/r-data-science-project-customer-segmentation/
2) https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python