Customer Segmentation using R

Data Exploration

Importing data from csv format

```
customer_data <- read.csv("Mall_Customers.csv")
```

Have a look at the dataset

```
head(customer_data)
```

	CustomerID <int>	Gender <chr>	A… <int>	Annual.Income..k.. <int>	Spending.Score..1.100. <int>
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

6 rows

Structure of the dataset

```
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 ...
```

Columns of the dataset

```
names(customer_data)
```

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

Summary of the dataset

```
summary(customer_data)
```

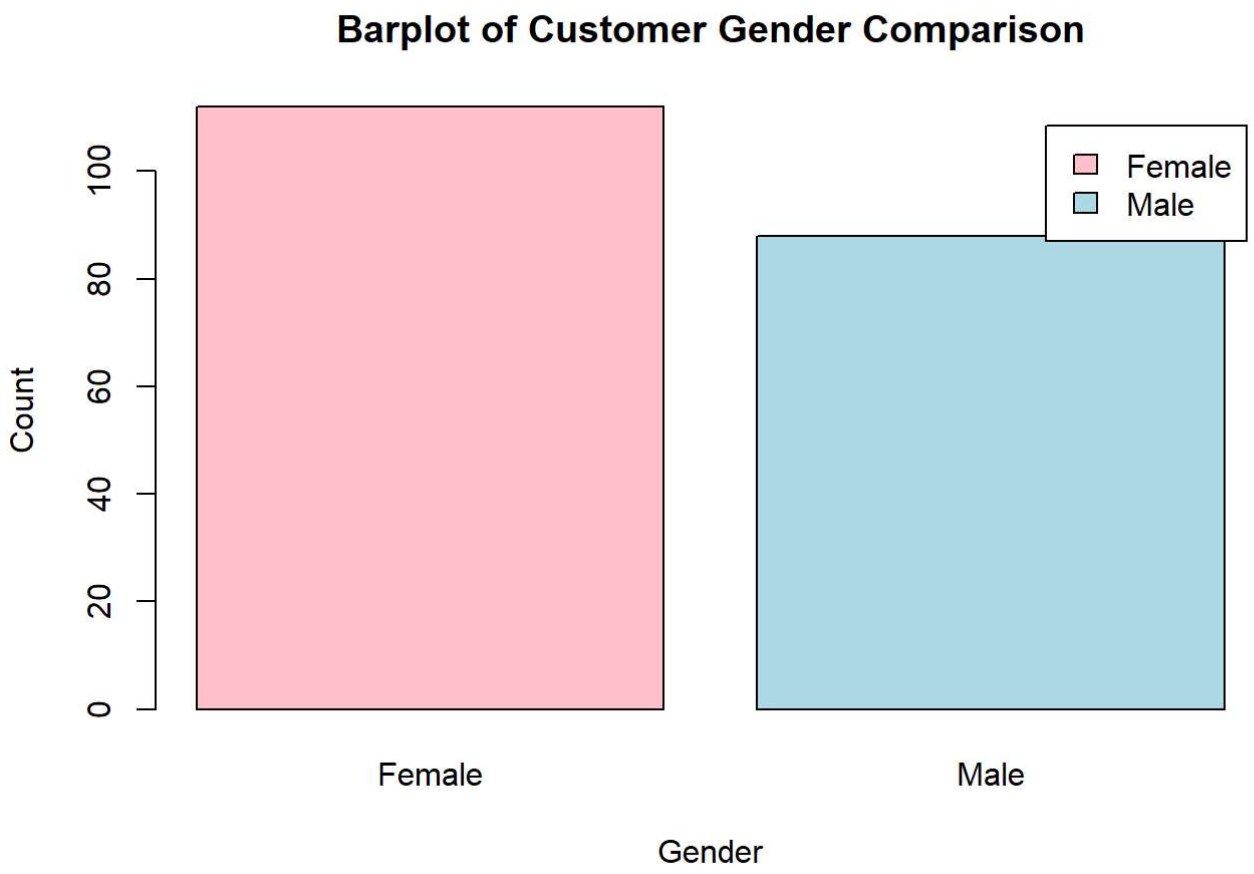
```
##    CustomerID        Gender               Age        Annual.Income..k..
##  Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00
##  1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50
##  Median :100.50   Mode  :character   Median :36.00   Median : 61.50
##  Mean   :100.50                      Mean   :38.85   Mean   : 60.56
##  3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00
##  Max.   :200.00                      Max.   :70.00   Max.   :137.00
##  Spending.Score..1.100.
##  Min.   : 1.00
##  1st Qu.:34.75
##  Median :50.00
##  Mean   :50.20
##  3rd Qu.:73.00
##  Max.   :99.00
```

Data Visualization and Analyzing Variables

Customer Gender visualization

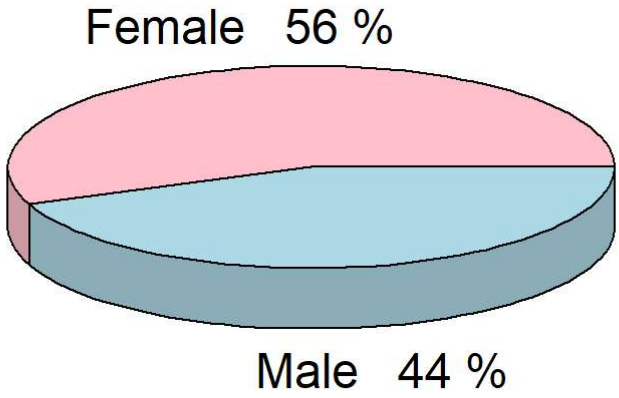
Creating a barplot and a piechart to show the gender distribution across the customer data

```
gender_data <- table(customer_data$Gender)
barplot(gender_data, main = "Barplot of Customer Gender Comparison", ylab="Count", xlab="Gend
er", col=c("pink", "lightblue"), legend=rownames(gender_data))
```




```
pct_gender=round(gender_data/sum(gender_data)*100)
lbs=paste(c("Female","Male")," ",pct_gender,"%")

library(plotrix)
pie3D(pct_gender,labels=lbs,main="Pie Chart Depicting the Ratio of Female and Male", col=c("p
ink", "lightblue"))
```



The barplot and pie chart concludes the customer dataset has more female customers at 56% ratio as compared to male customers at 44%

Customer Age Distribution visualization

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

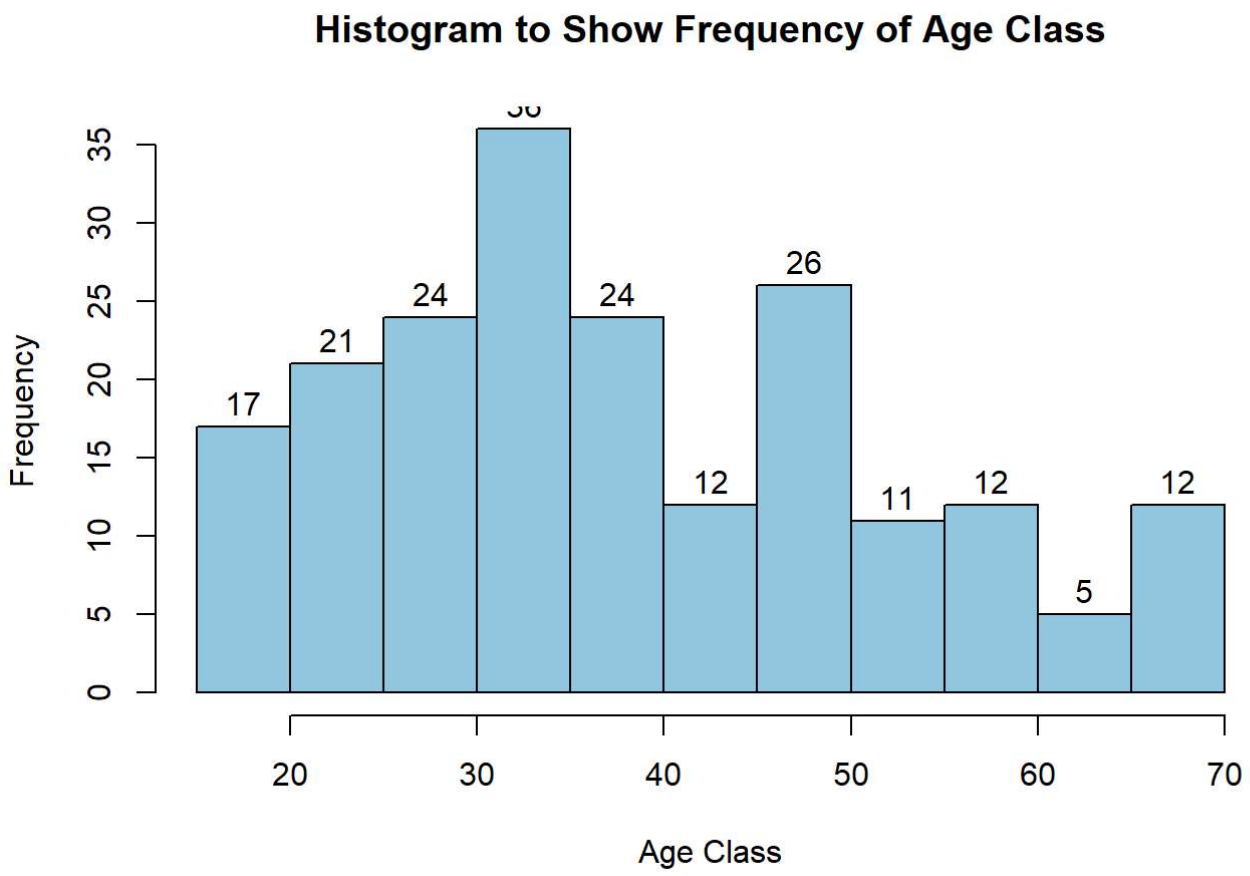
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.75   36.00   38.85   49.00   70.00
```

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

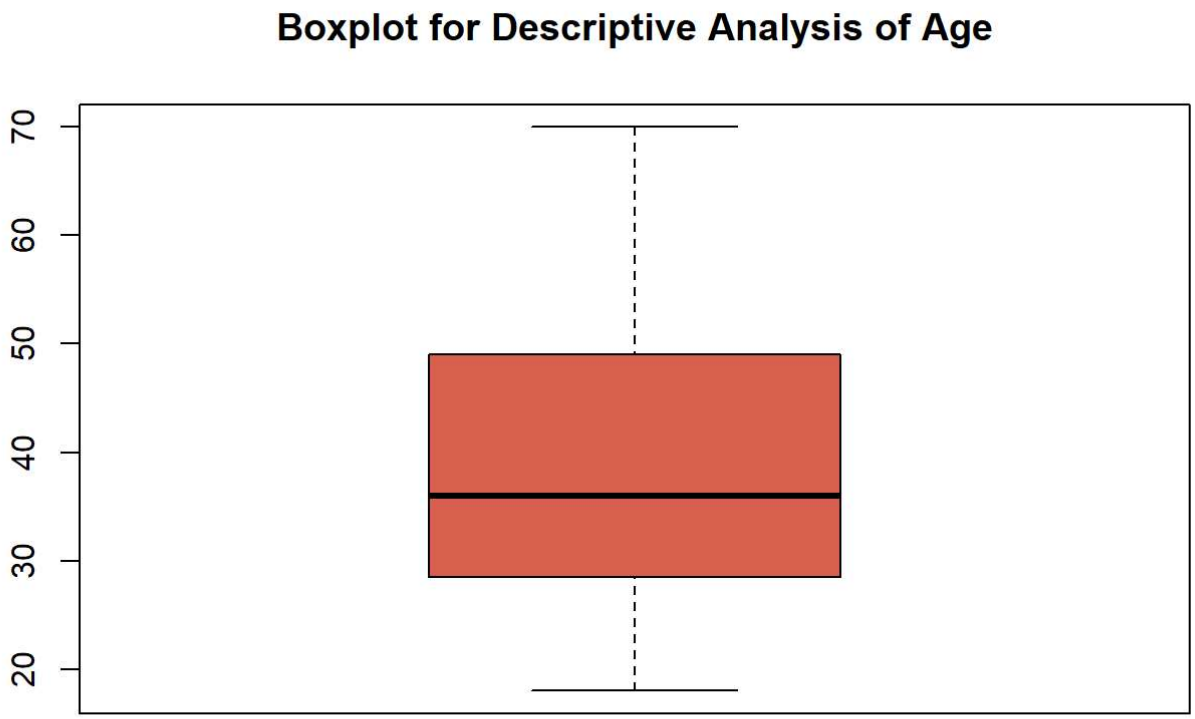
```
## [1] 13.96901
```

Creating histogram to view the frequency of customer ages

```
hist(customer_data$Age,
     col="#92C5DE",
     main="Histogram to Show Frequency of Age Class",
     xlab="Age Class", ylab="Frequency",
     labels=TRUE)
```




```
boxplot(customer_data$Age,
        col="#D6604D",
        main="Boxplot for Descriptive Analysis of Age")
```



The above two visualizations conclude that the maximum customers are aged between 30 and 35, also the minimun and maximum age of the customers are 18 and 70 respectively.

Analysis and Visualization of Annual Income of the Customers

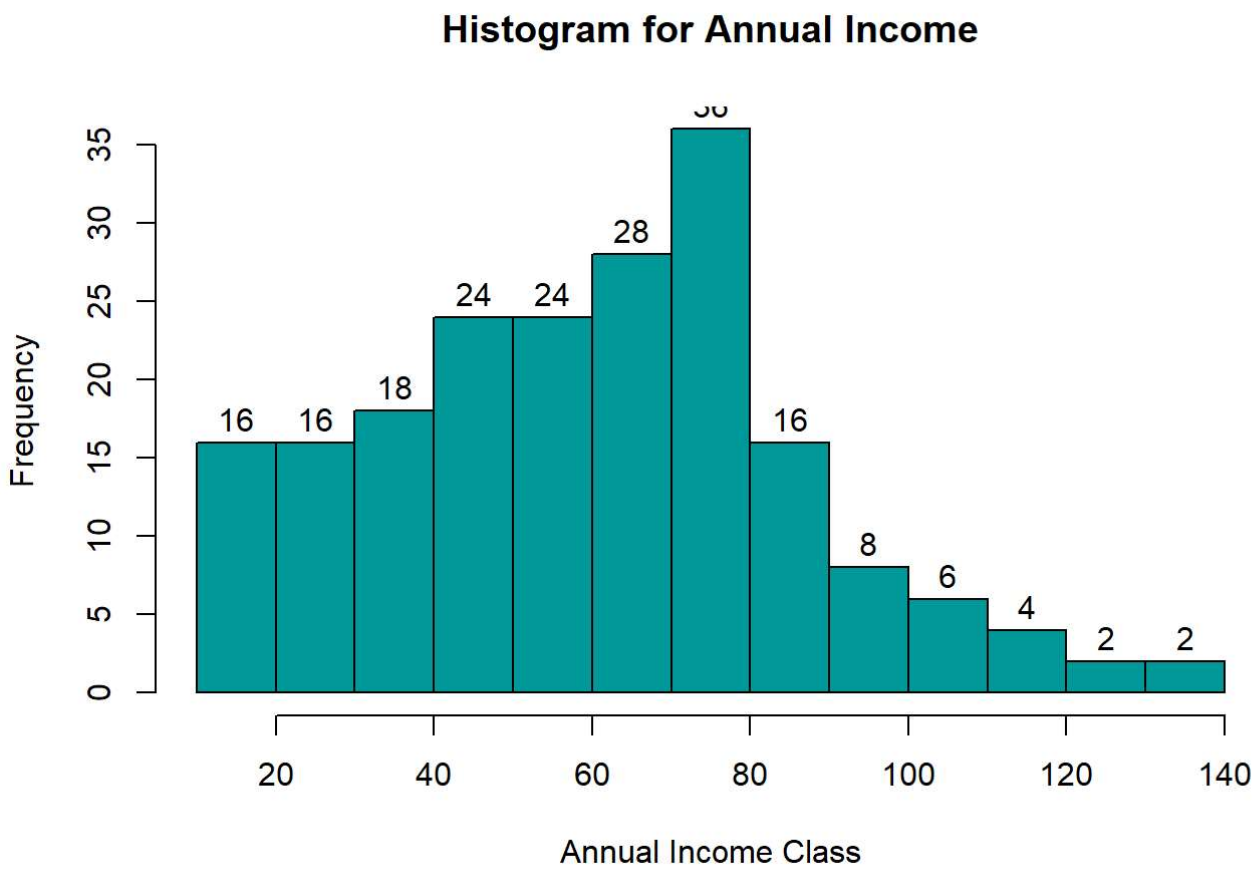
```
summary(customer_data$Annual.Income..k..)
```

```
##    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..)
```

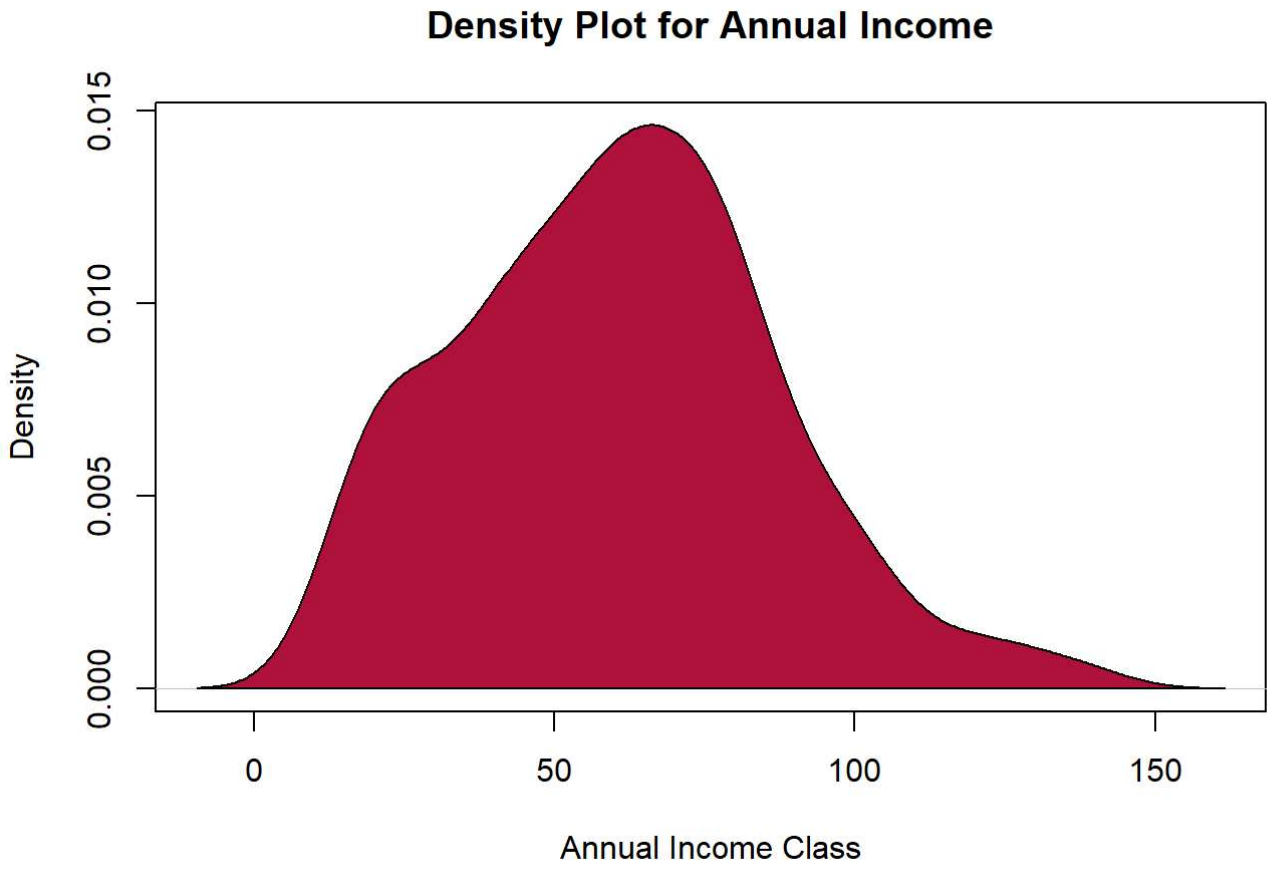
```
## [1] 26.26472
```

```
hist(customer_data$Annual.Income..k..,
     col="#009999",
     main="Histogram for Annual Income",
     xlab="Annual Income Class",
     ylab="Frequency",
     labels=TRUE)
```




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

polygon(density(customer_data$Annual.Income..k..),
        col="#AE123A")
```



The above two visualizations show that minimum and maximum annual incomes of our customers ranging from 15 and 137. The average income is 60.56. The Kernel Density Plot draws the normal distribution for the variable.

Analysis and Visualization of Spending Score of the Customers

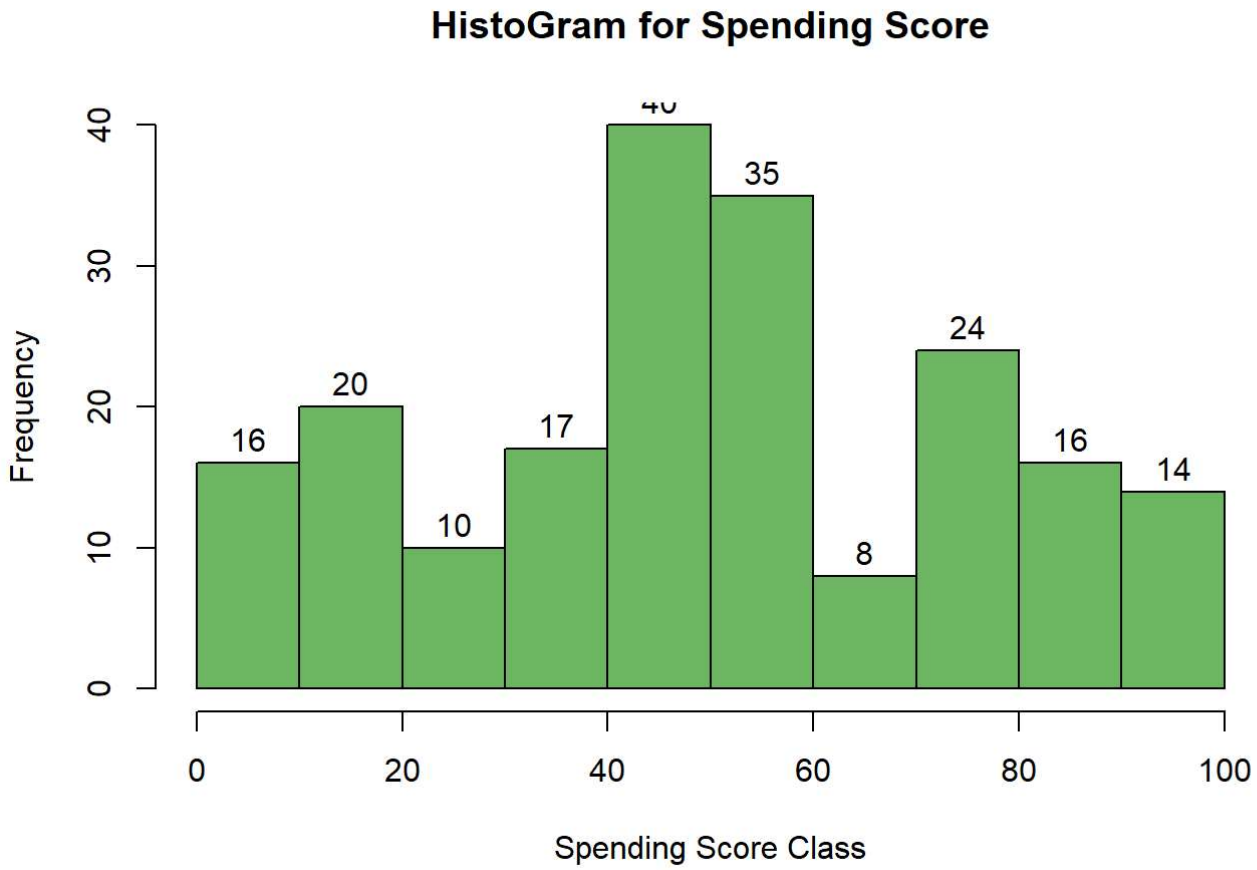
```
summary(customer_data$Spending.Score..1.100.)
```

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

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

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

```
hist(customer_data$Spending.Score..1.100.,
     main="HistoGram for Spending Score",
     xlab="Spending Score Class",
     ylab="Frequency",
     col="#6DB562",
     labels=TRUE)
```



Minimum and maximum spending score of our customers are 1 and 99 respectively, as well the average score is 50.20. The above histogram shows the highest number of customers are having spending score between 40 and 50.

K-means Algorithm

K-means algorithm is used for clustering, that computes the centroids and iterates until it finds the optimal centroid. The initial step is to randomly select k objects that are the means of the clusters. The remaining objects fall in the same cluster as their closest cluster mean, by finding Euclidean Distance between the objects. After each data object is assigned to a cluster, new mean value is calculated for each cluster in the data. New centers are recalculated and the observations are checked if they are closer to different clusters, if so the object is reassigned to a new cluster. This process is iterated several times until no new alterations in cluster assignments, that is the clusters in the new iteration are the same as obtained in the previous iteration.

Optimal Cluster Selection

Below methods are popular while determining optimal number of clusters

1. Elbow method
2. Silhouette method
3. Gap statistic

Elbow method

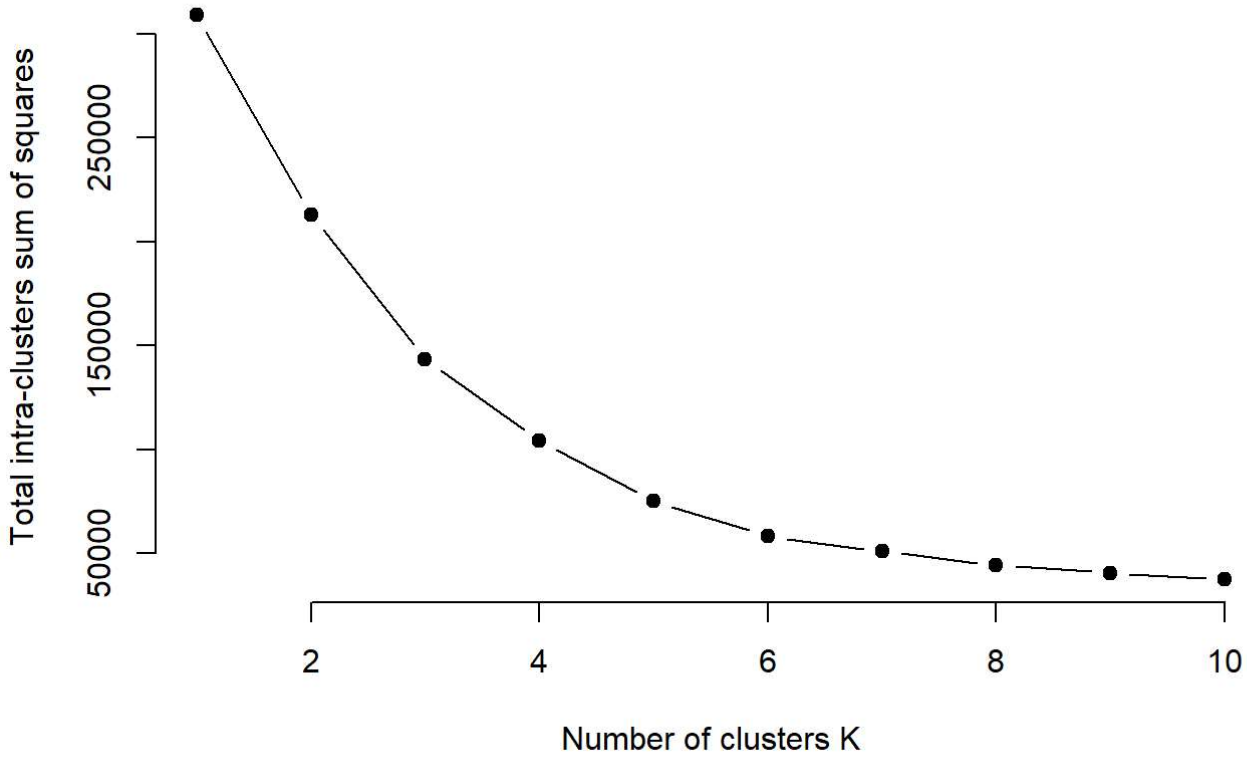
```
library(purrr)
```

```
set.seed(123)
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")
```



The elbow plot seems be appearing at the bend at 4, hence the appropriate number of clusters is 4.

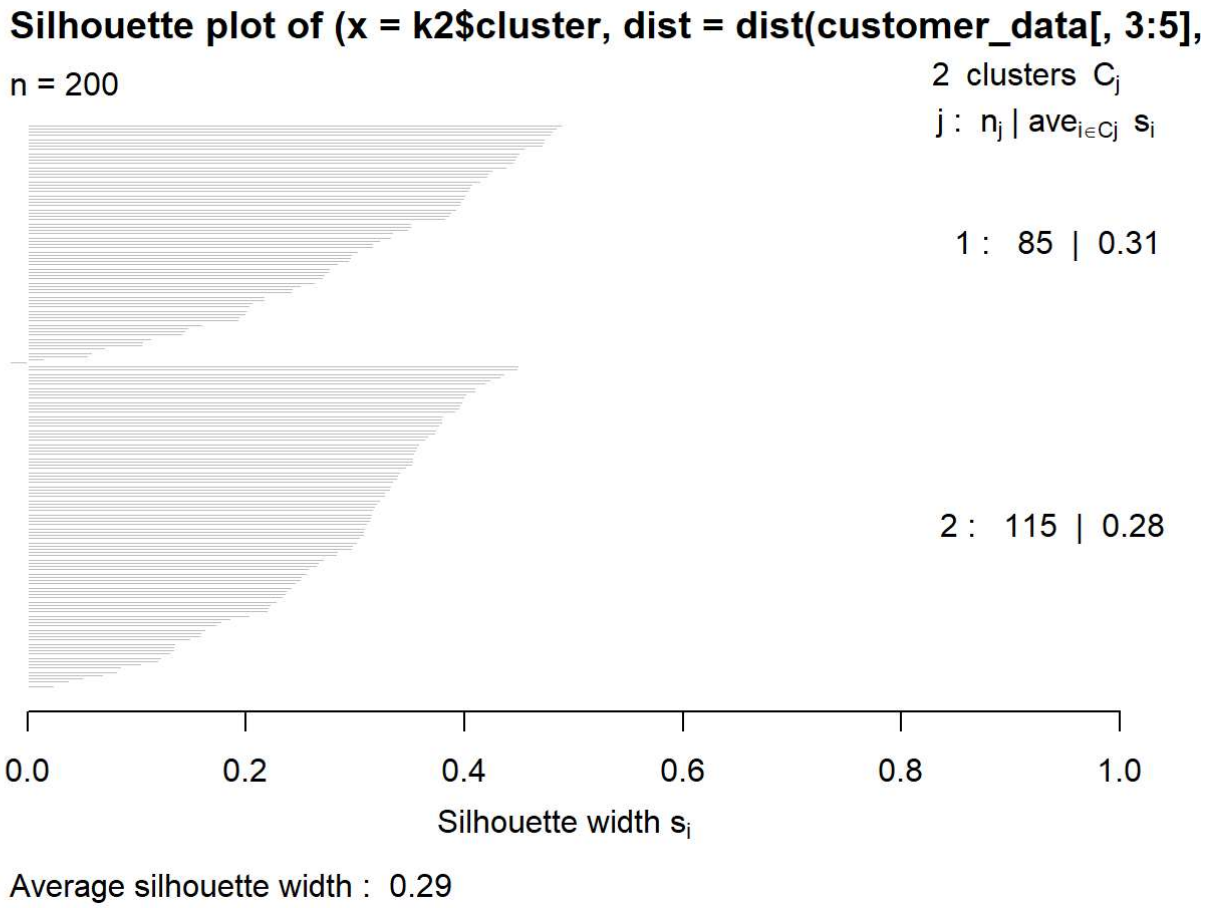
Average Silhouette Method

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

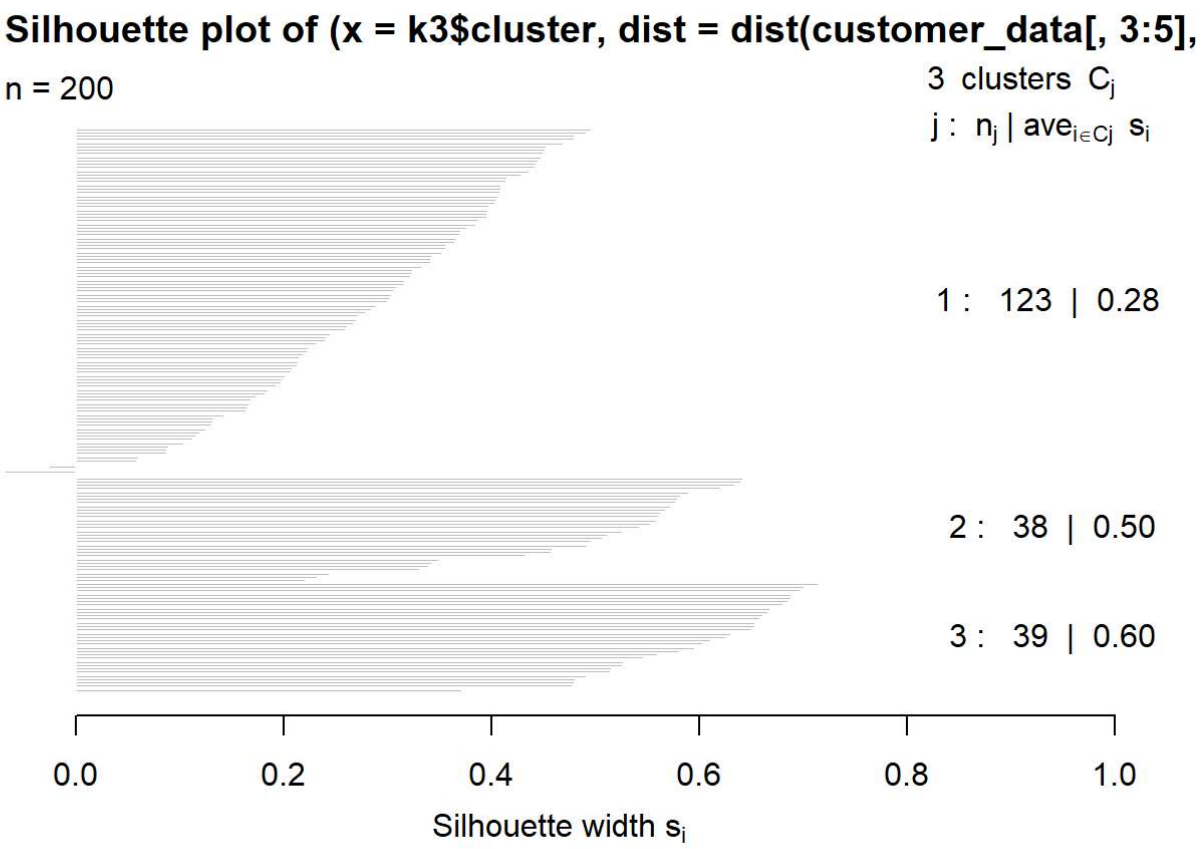
```
## Warning: package 'gridExtra' was built under R version 4.2.1
```

```
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")))
```

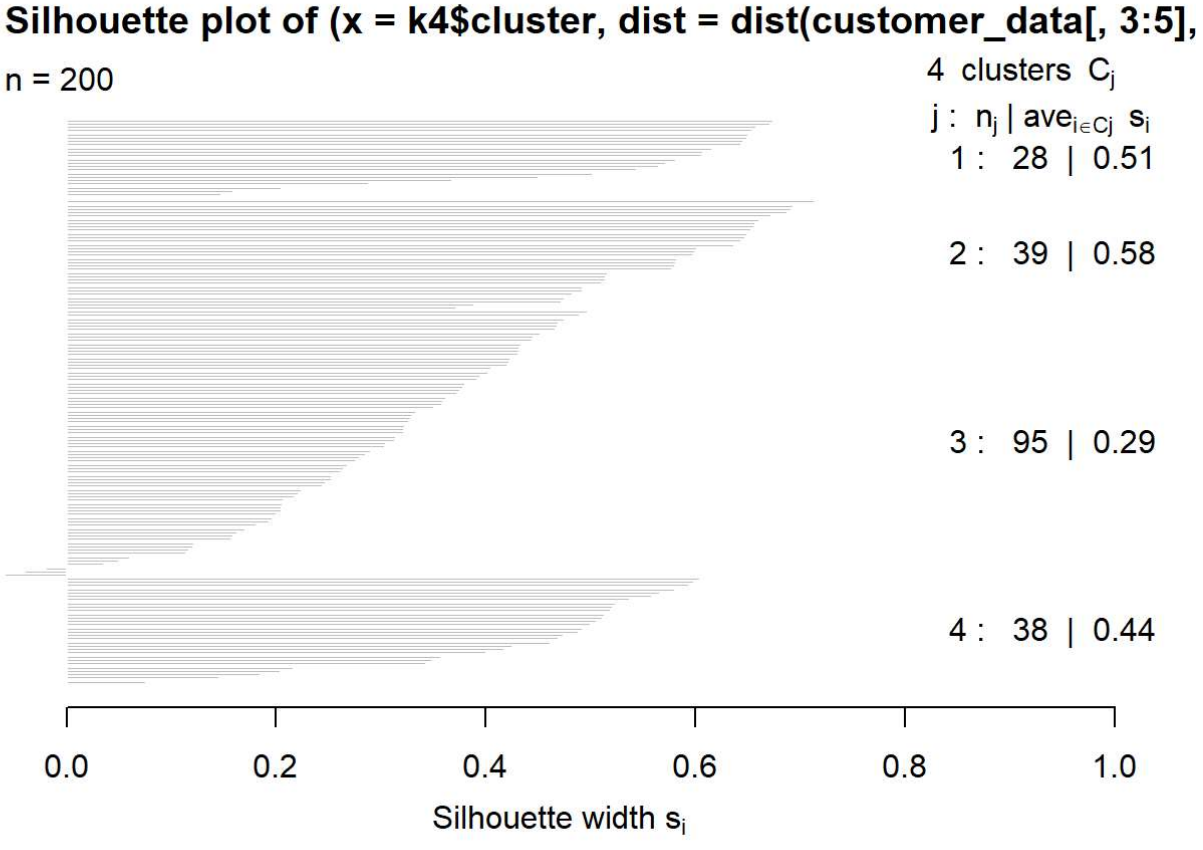



```
k3 <- kmeans(customer_data[,3:5], 3, iter.max = 100, nstart = 50, algorithm = "Lloyd")
s3 <- plot(silhouette(k3$cluster, dist(customer_data[,3:5], "euclidean")))
```



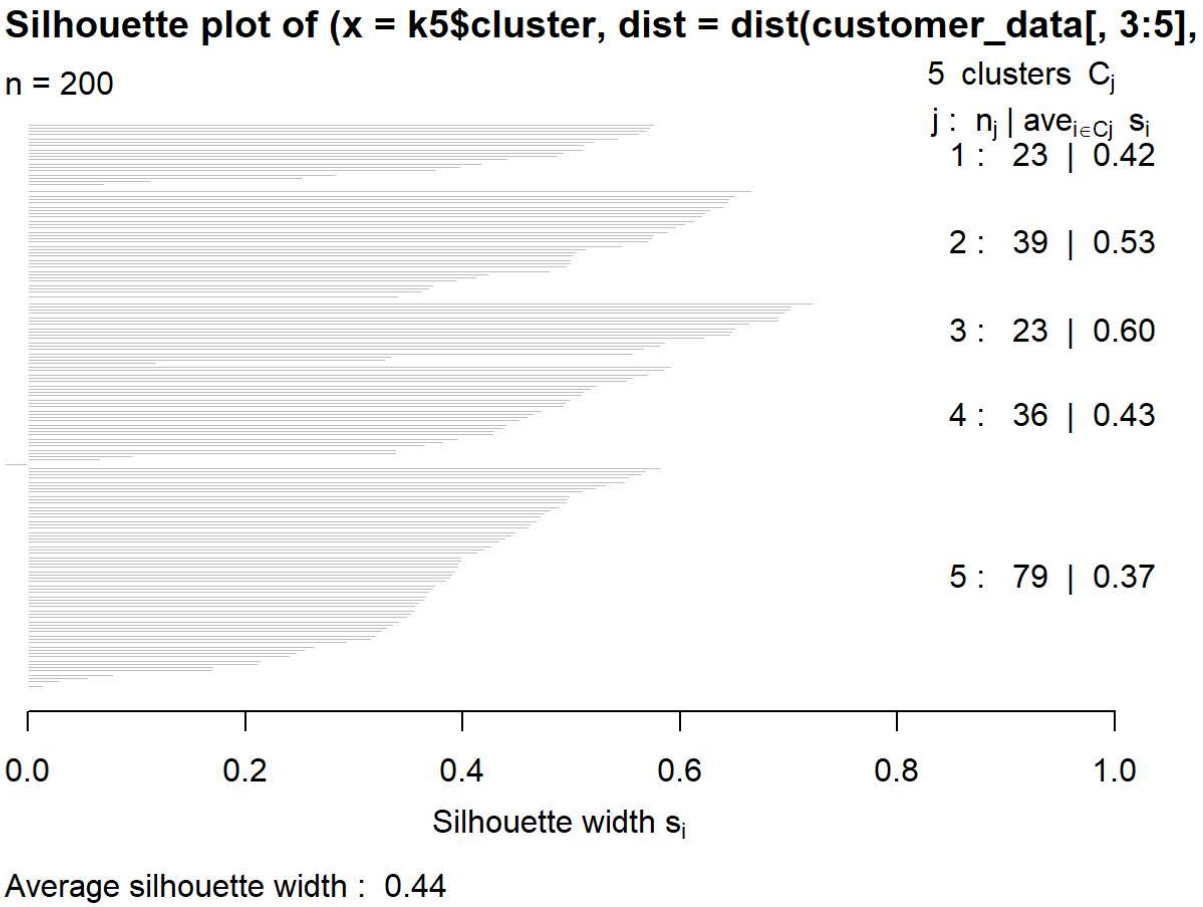
Average silhouette width : 0.38

```
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")))
```

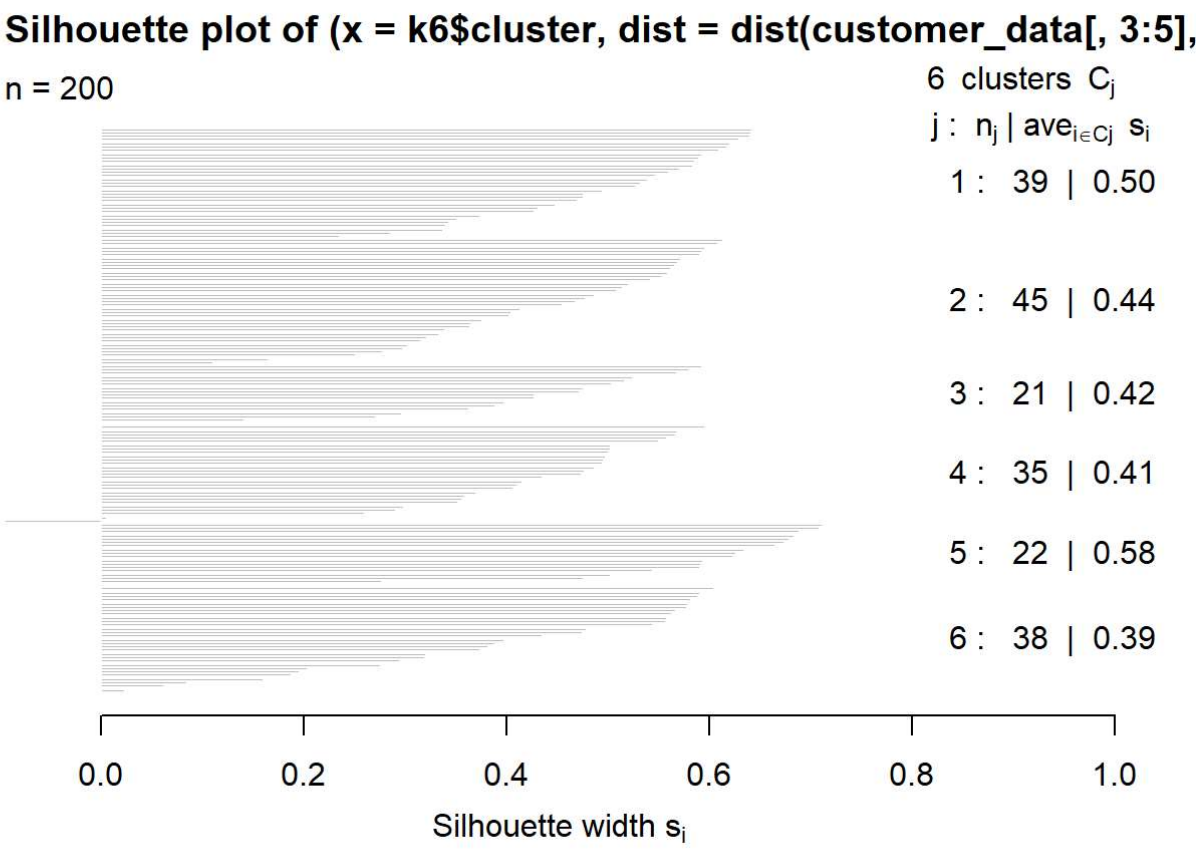


Average silhouette width : 0.41

```
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")))
```

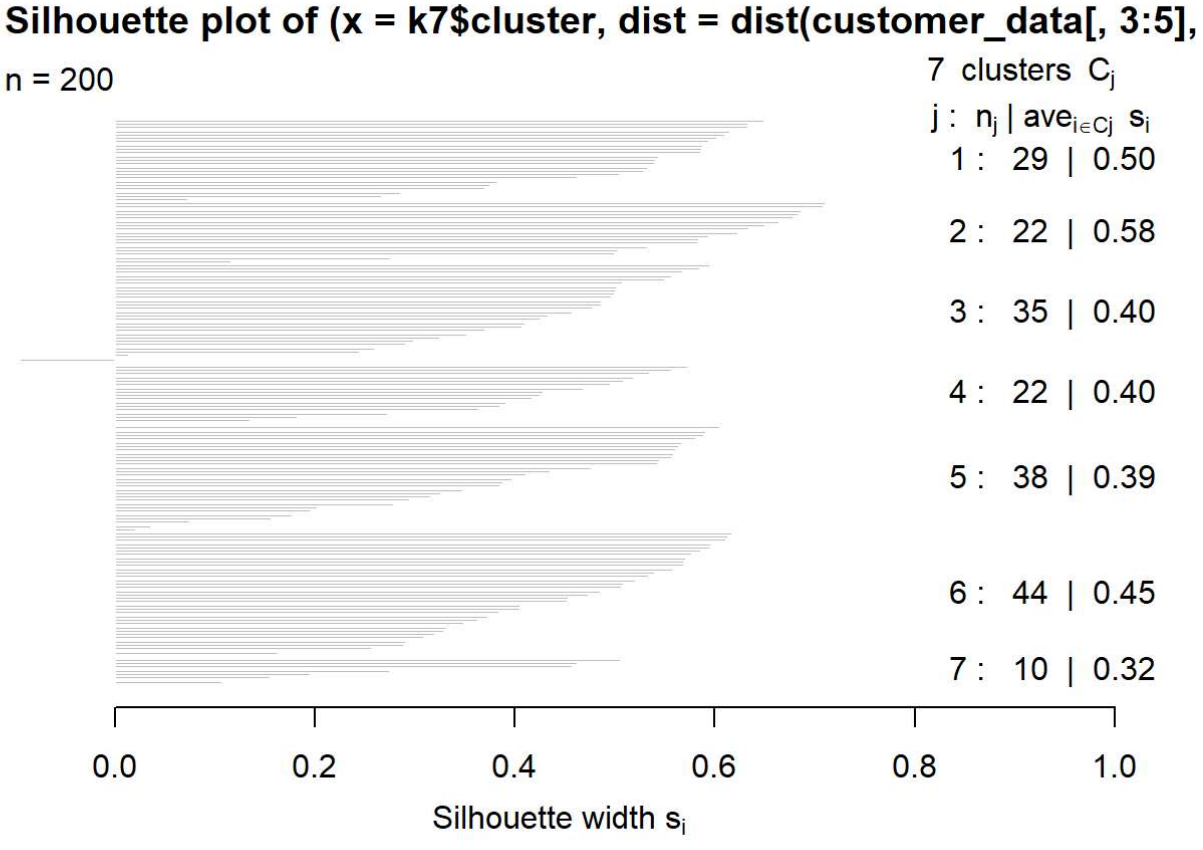



```
k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
s6<-plot(silhouette(k6$cluster,dist(customer_data[,3:5],"euclidean")))
```



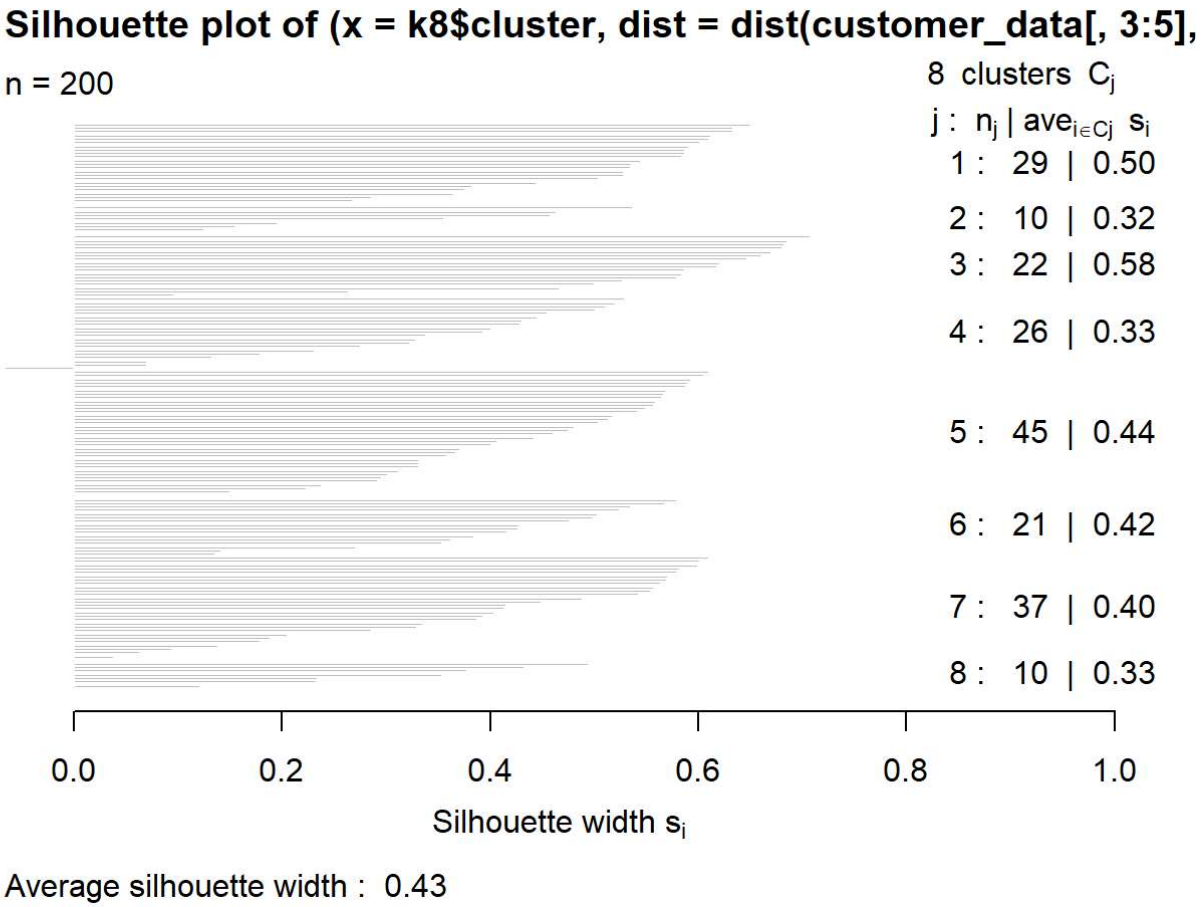
Average silhouette width : 0.45

```
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")))
```

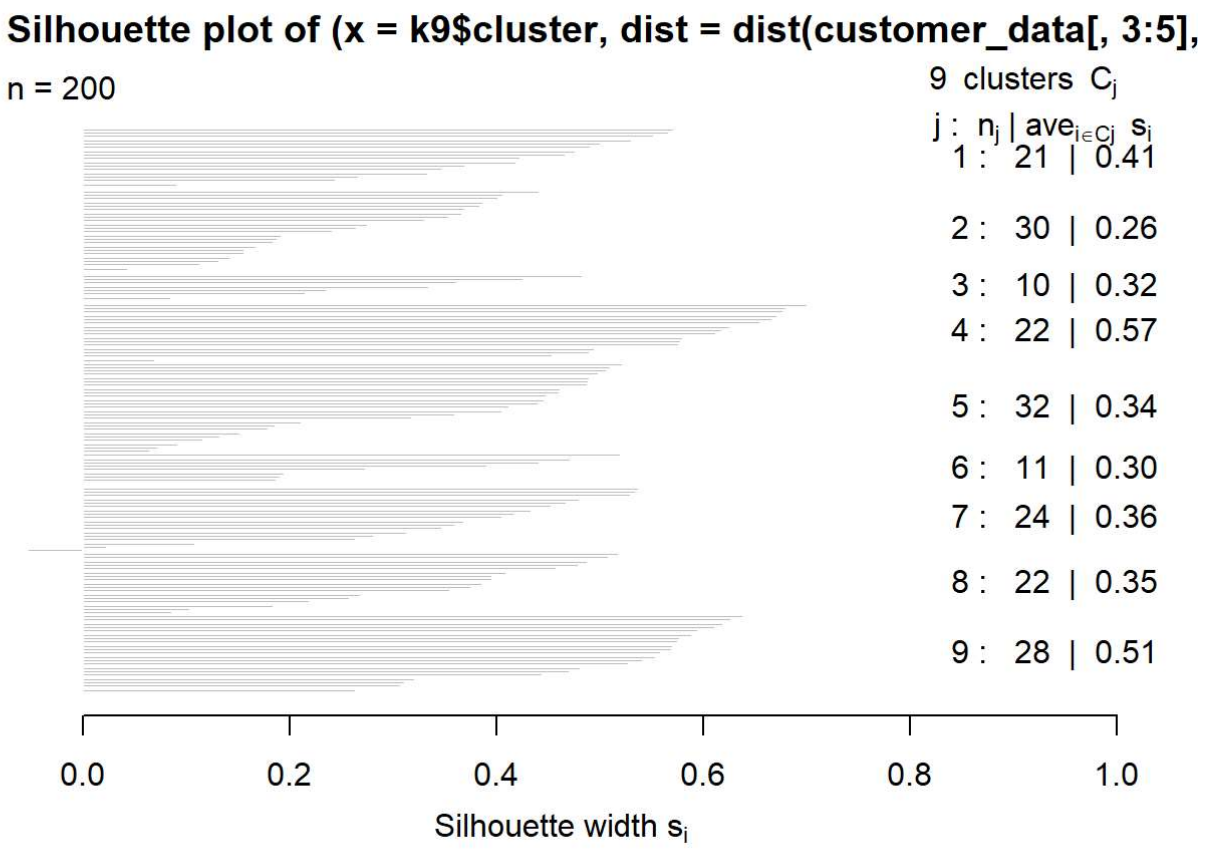


Average silhouette width : 0.44

```
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")))
```

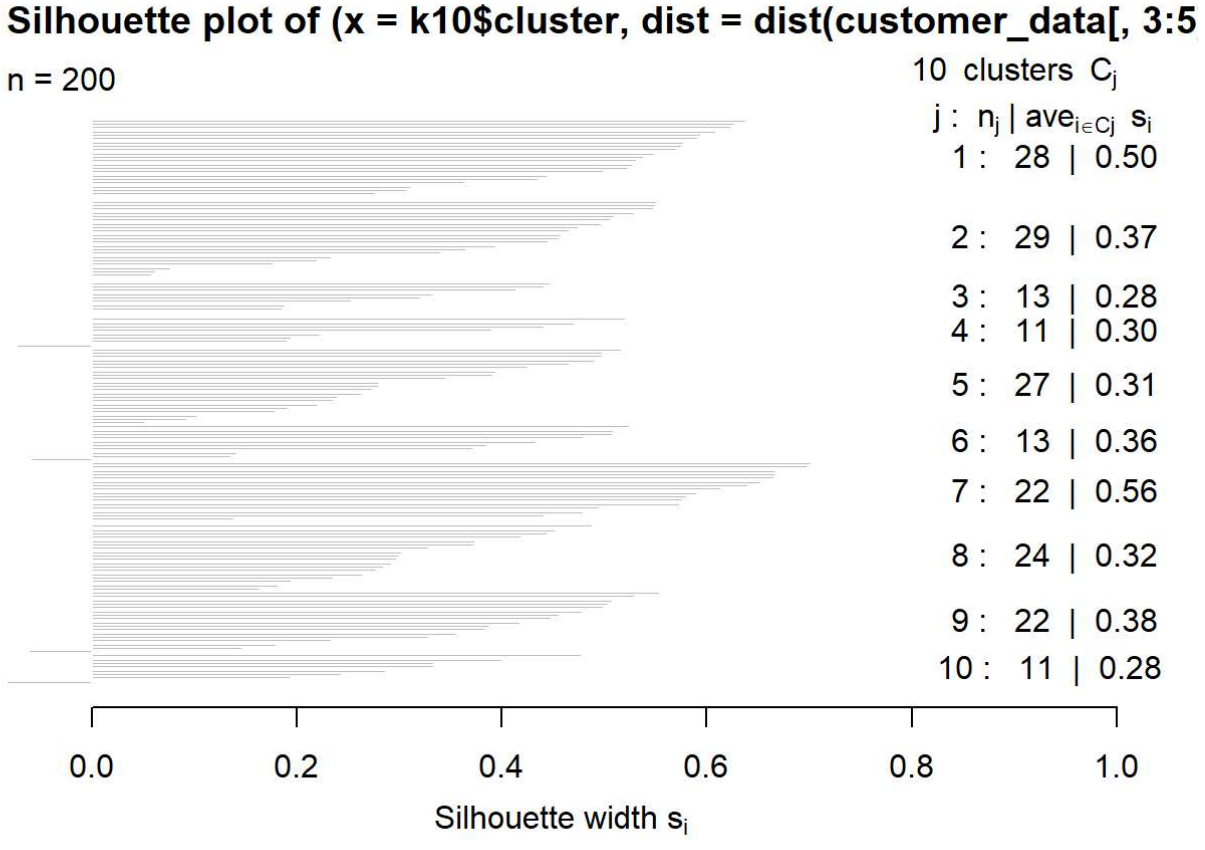



```
k9<-kmeans(customer_data[,3:5],9,iter.max=100,nstart=50,algorithm="Lloyd")
s9<-plot(silhouette(k9$cluster,dist(customer_data[,3:5],"euclidean")))
```



Average silhouette width : 0.39

```
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")))
```



Average silhouette width : 0.38

Now, we make use of the fviz_nbclust() function to determine and visualize the optimal number of clusters as follows

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

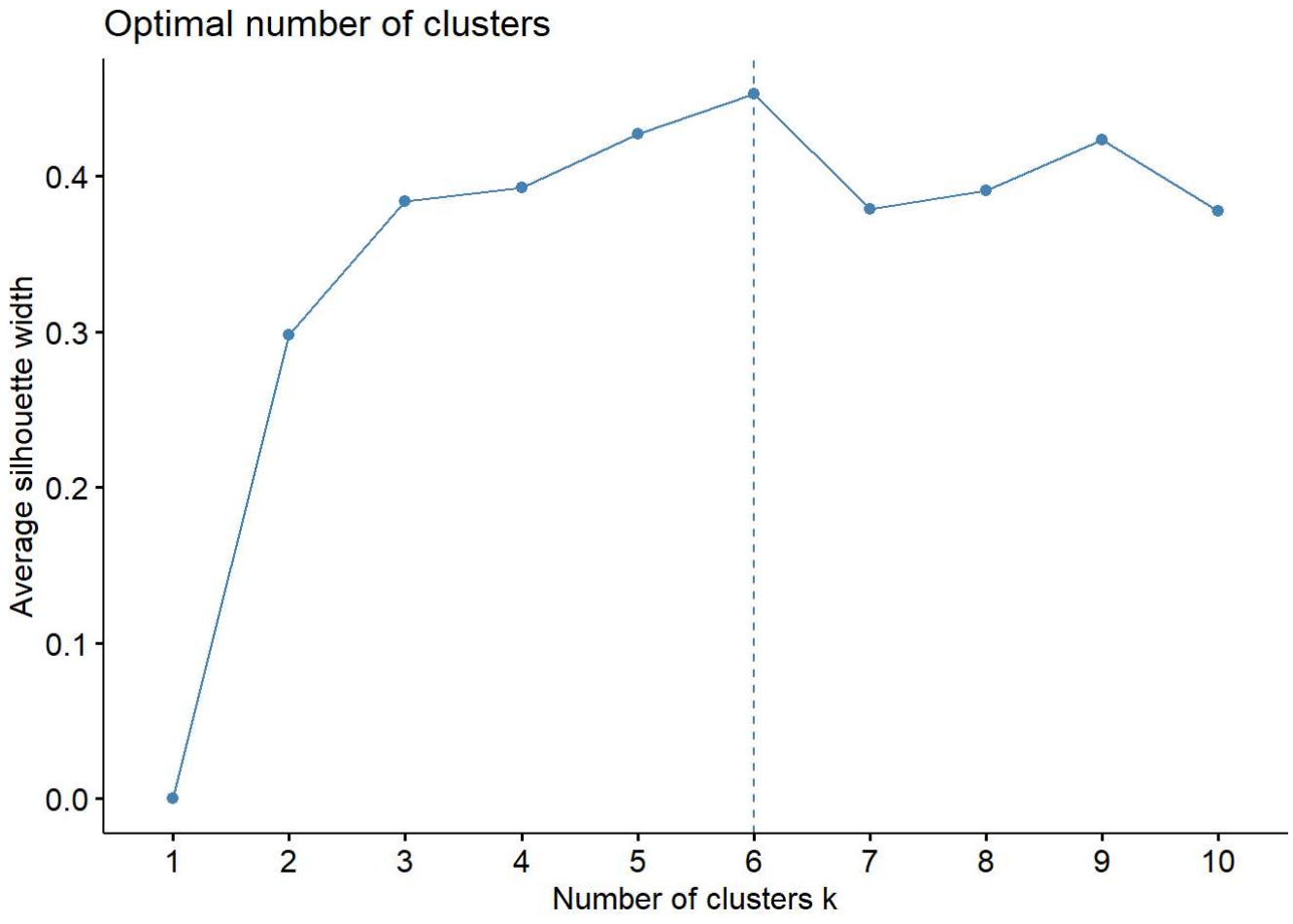
```
## Warning: package 'factoextra' was built under R version 4.2.1
```

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

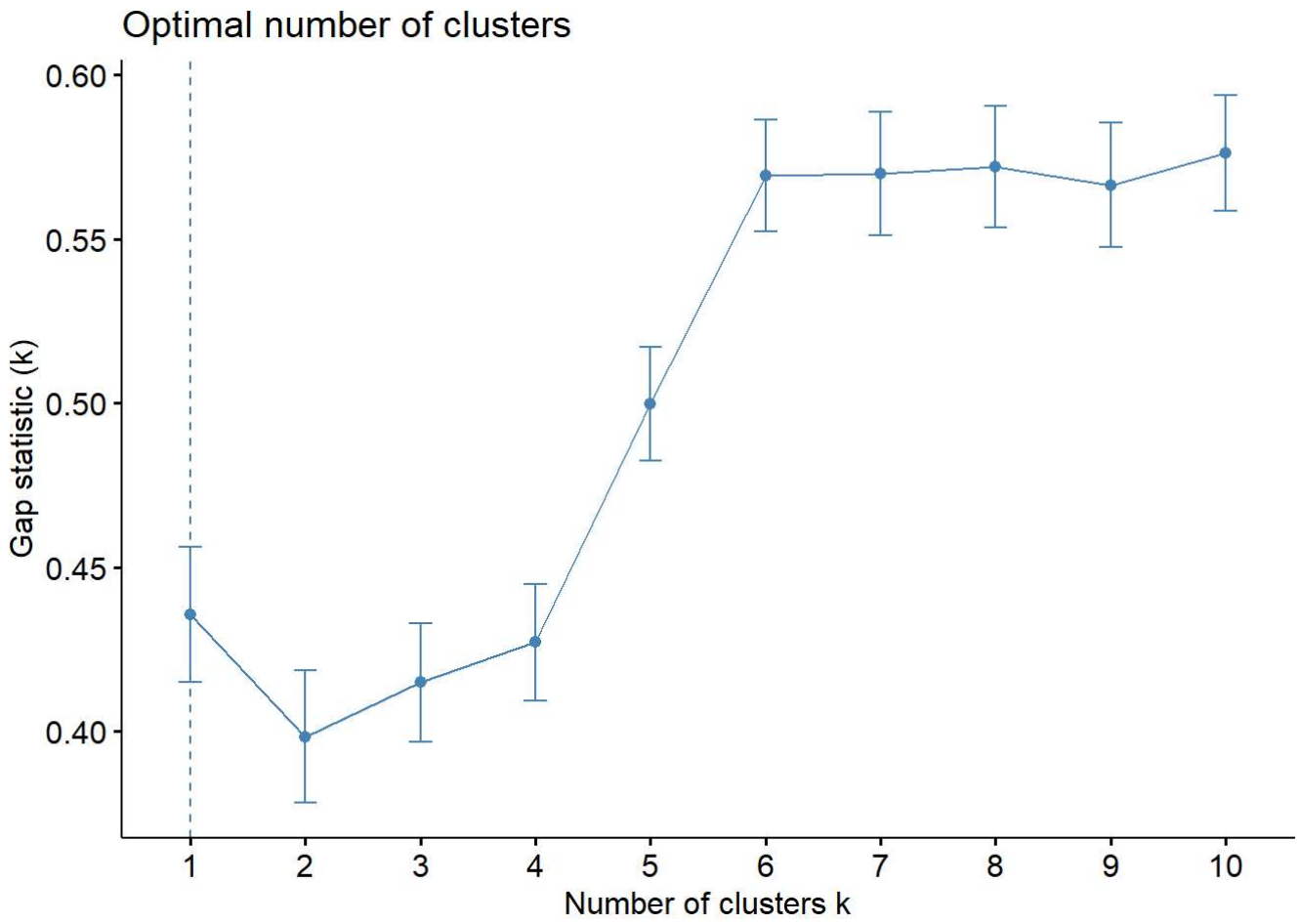
```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

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

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




```
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 45, 21, 35, 39, 38, 22
##
## Cluster means:
##        Age Annual.Income..k.. Spending.Score..1.100.
## 1 56.15556           53.37778               49.08889
## 2 44.14286           25.14286               19.52381
## 3 41.68571           88.22857               17.28571
## 4 32.69231           86.53846               82.12821
## 5 27.00000           56.65789               49.13158
## 6 25.27273           25.72727               79.36364
##
## Clustering vector:
##   [1] 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2
##  [38] 6 2 6 1 6 1 5 2 6 1 5 5 5 1 5 5 1 1 1 1 1 5 1 1 5 1 1 1 5 1 1 5 5 1 1 1 1
##  [75] 1 5 1 5 5 1 1 5 1 1 5 1 1 5 5 1 1 5 1 5 5 5 1 5 1 5 5 1 1 5 1 5 1 1 1 1 1
## [112] 5 5 5 5 5 1 1 1 1 5 5 5 4 5 4 3 4 3 4 3 4 5 4 3 4 3 4 3 4 3 4 5 4 3 4 3 4
## [149] 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3
## [186] 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
##
## Within cluster sum of squares by cluster:
## [1]  8062.133  7732.381 16690.857 13972.359  7742.895  4099.818
##  (between_SS / total_SS =  81.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

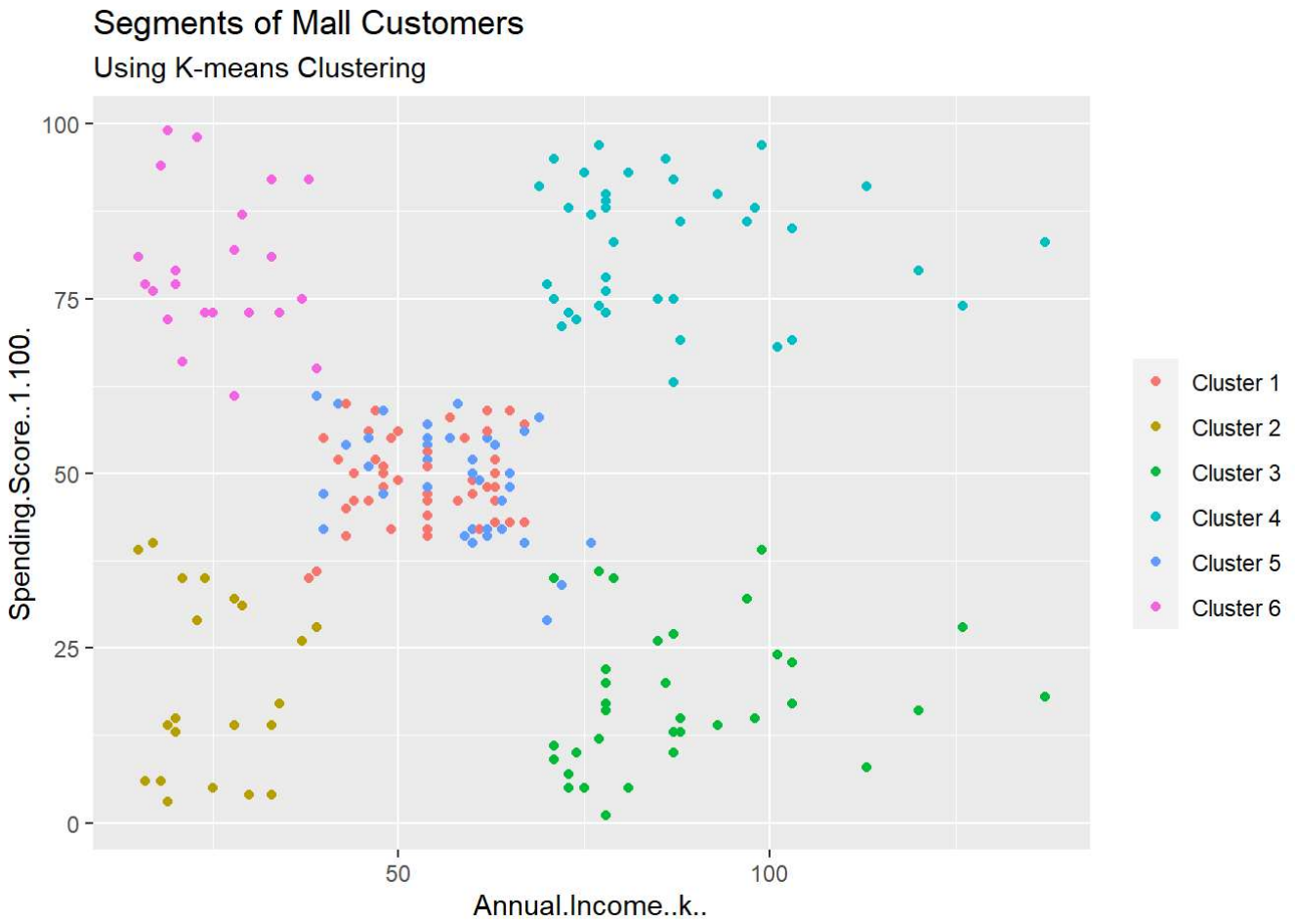
```
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)
```

```
## 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
```

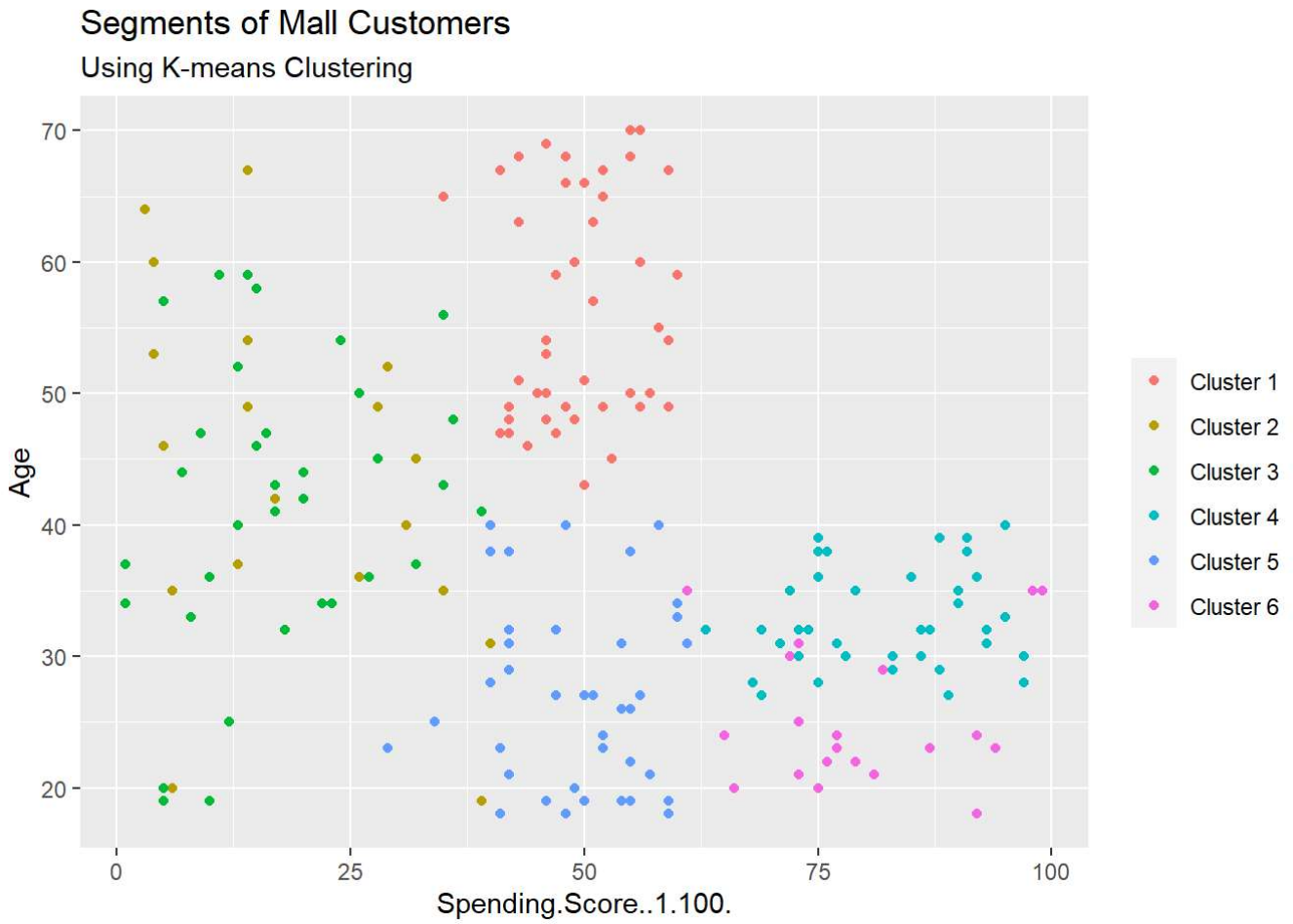
```
pcclust$rotation[,1:2]
```

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



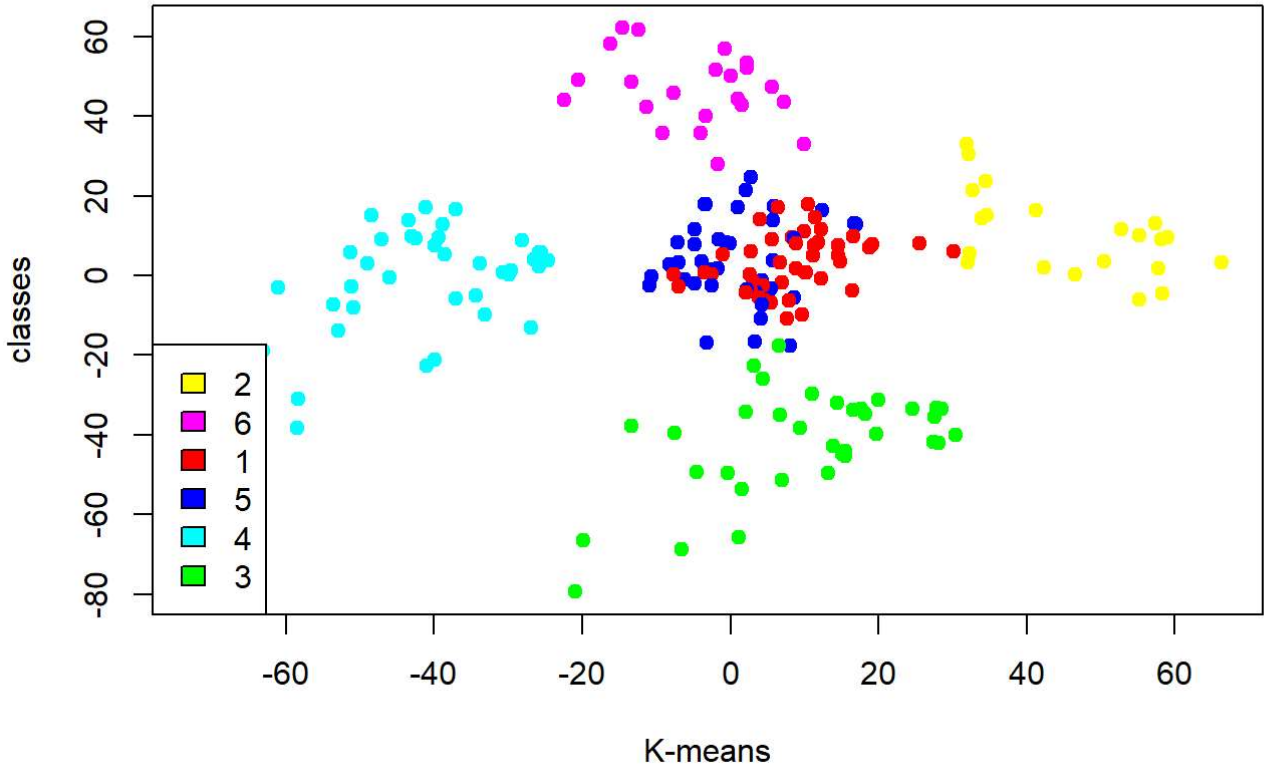

```
ggplot(customer_data, aes(x =Spending.Score..1.100., y =Age)) +
  geom_point(stat = "identity", aes(color = as.factor(k6$cluster))) +
  scale_color_discrete(name=" ",
                       breaks=c("1", "2", "3", "4", "5","6"),
                       labels=c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster
5","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="classes")
legend("bottomleft",unique(dignm),fill=unique(kCols(digCluster)))
```


classes
K-means
2
6
1
5
4
3
-80
-60
-40
-20
0
20
40
60