Module 4 Assignment

Manisha Meka

5/9/2021

Customer Segmentation using Machine Learning in R

In this data science project, we create a customer segmentation model using Mall Customers datset. We develop this using a class of machine learning. Specifically, we use of a clustering algorithm called K-means clustering. We analyze and visualize the data and then proceeded to implement the algorithm.

Import and read the data set

```
library("readr")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
customer_data=read.csv("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 ...
```

Variables in the data set

```
names(customer_data)
```

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

Display the first six rows of our dataset using the head() function and use the summary() function to output summary of it.

```r
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
```

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

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

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

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

```r
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
```

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

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

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

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

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

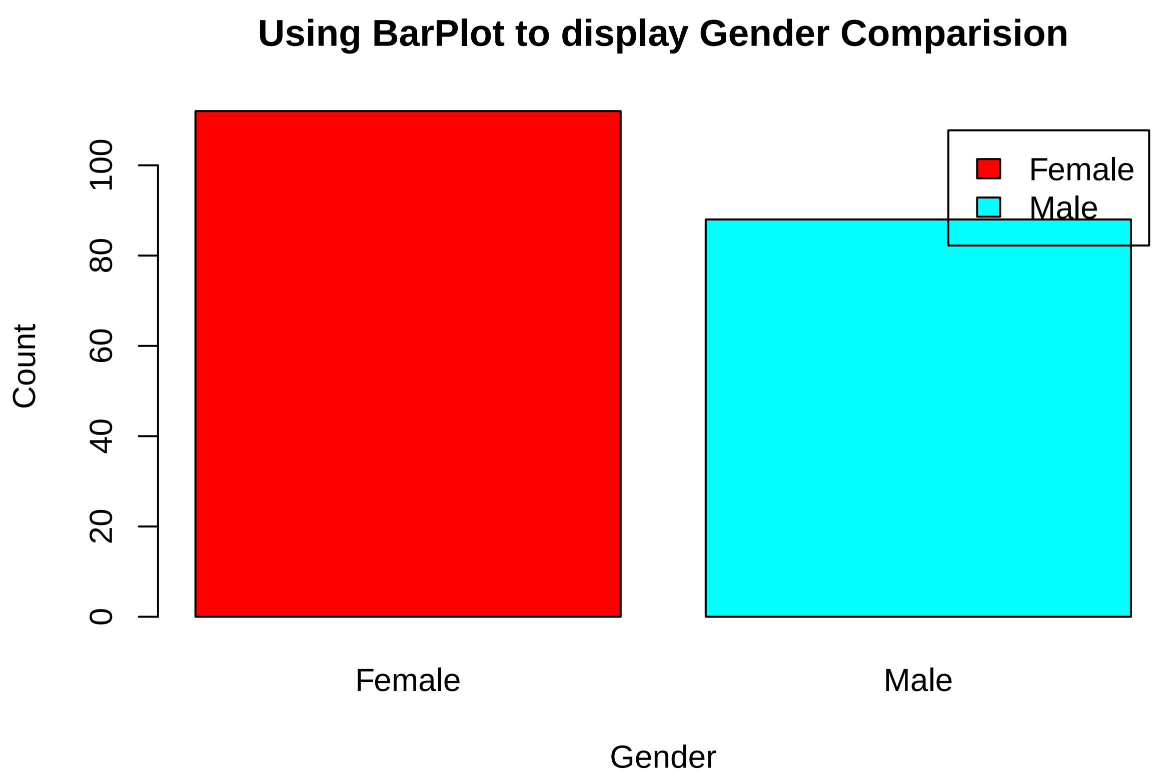
```
## [1] 25.82352
```

```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
```

#Customer Gender Visualization Create a barplot and a piechart to show the gender distribution across our customer_data dataset.

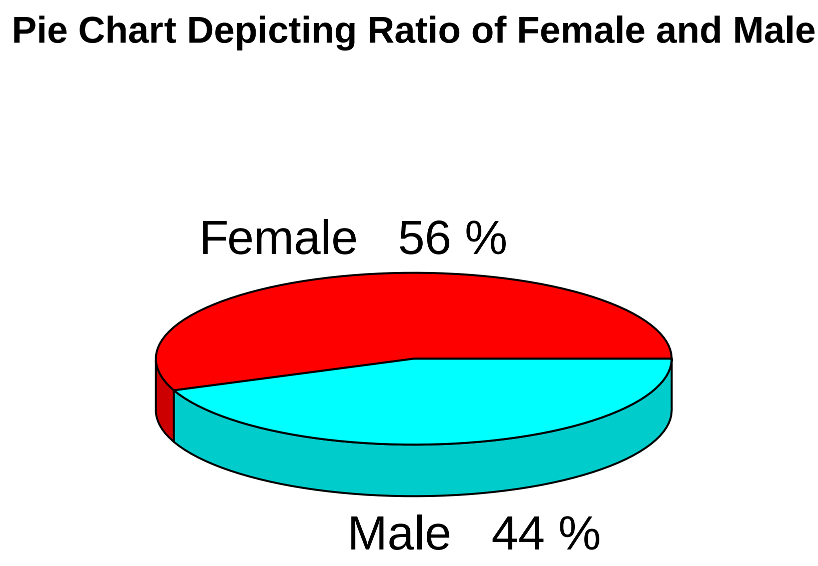
```r
a=table(customer_data$Gender)
barplot(a,main="Using BarPlot to display Gender Comparision",
       ylab="Count",
       xlab="Gender",
       col=rainbow(2),
       legend=rownames(a))
```



From the Bar plot, we observe that the number of females is higher than the males.

Create a pie chart to observe the ratio of male and female distribution.

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



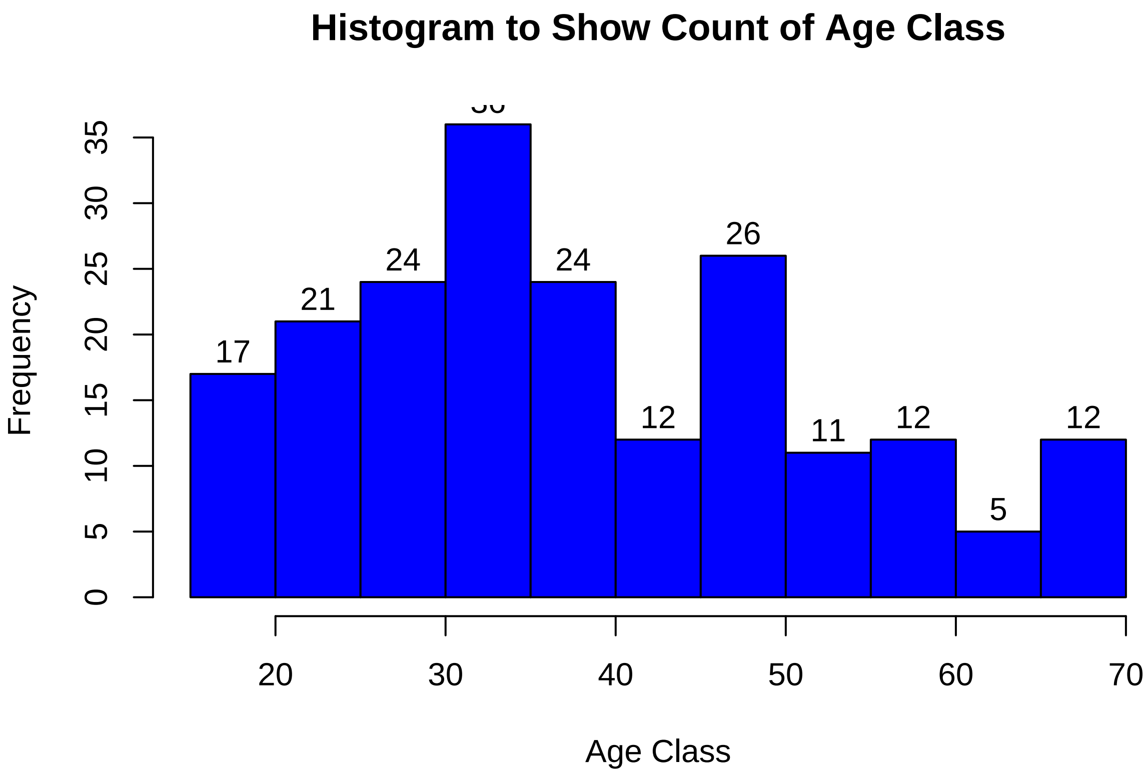
From the pie chart, the percentage of females is 56%, whereas the percentage of male in the customer dataset is 44%.

#Visualization of Age Distribution PLot a histogram to view the distribution to plot the frequency of customer ages. We will first proceed by taking summary of the Age variable.

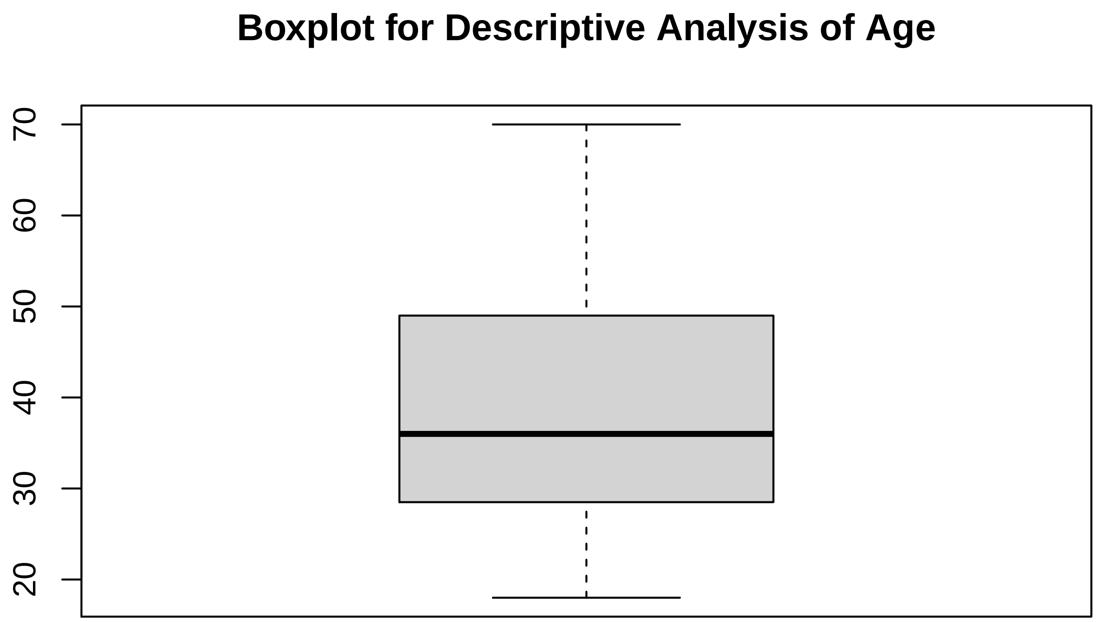
```
summary(customer_data$Age)
```

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

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




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



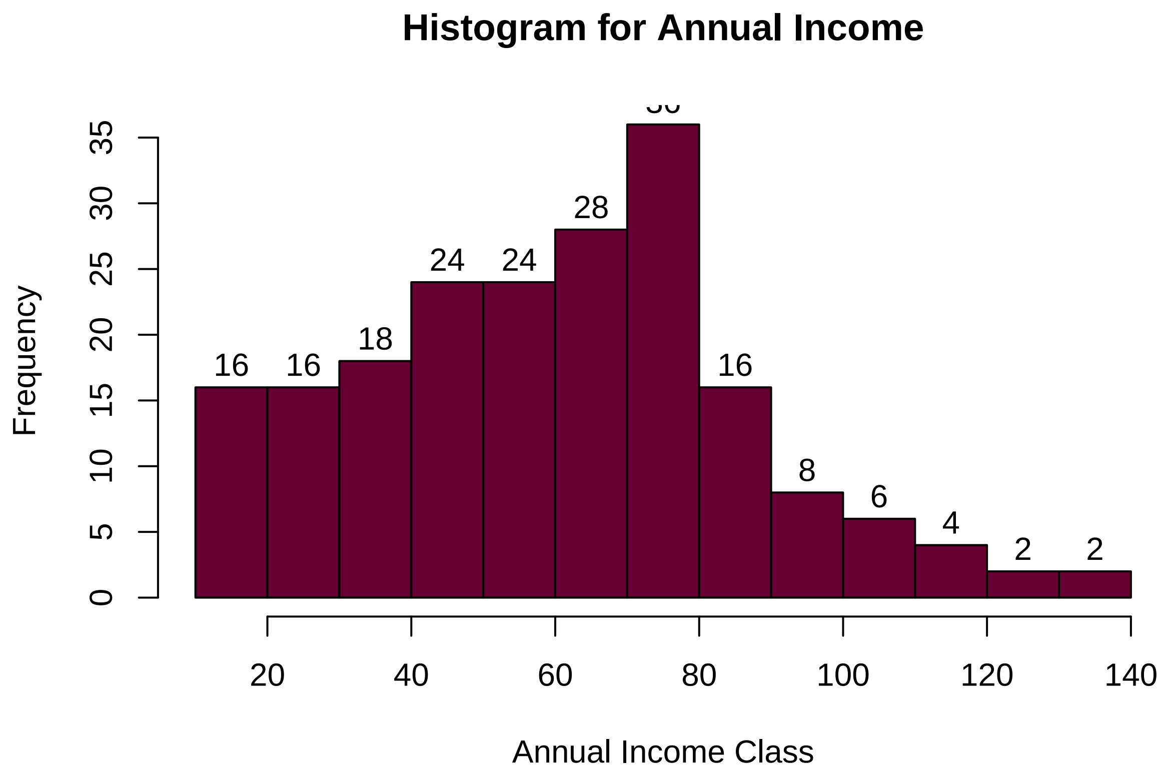
From the above two visualizations, we conclude that the maximum customer ages are between 30 and 35. The minimum age of customers is 18, whereas, the maximum age is 70.

#Visualization of Annual Income of the Customers Plot a histogram to view the distribution to plot the frequency of Annual Income of the Customers. We will first proceed by taking summary of the Annual Income variable.

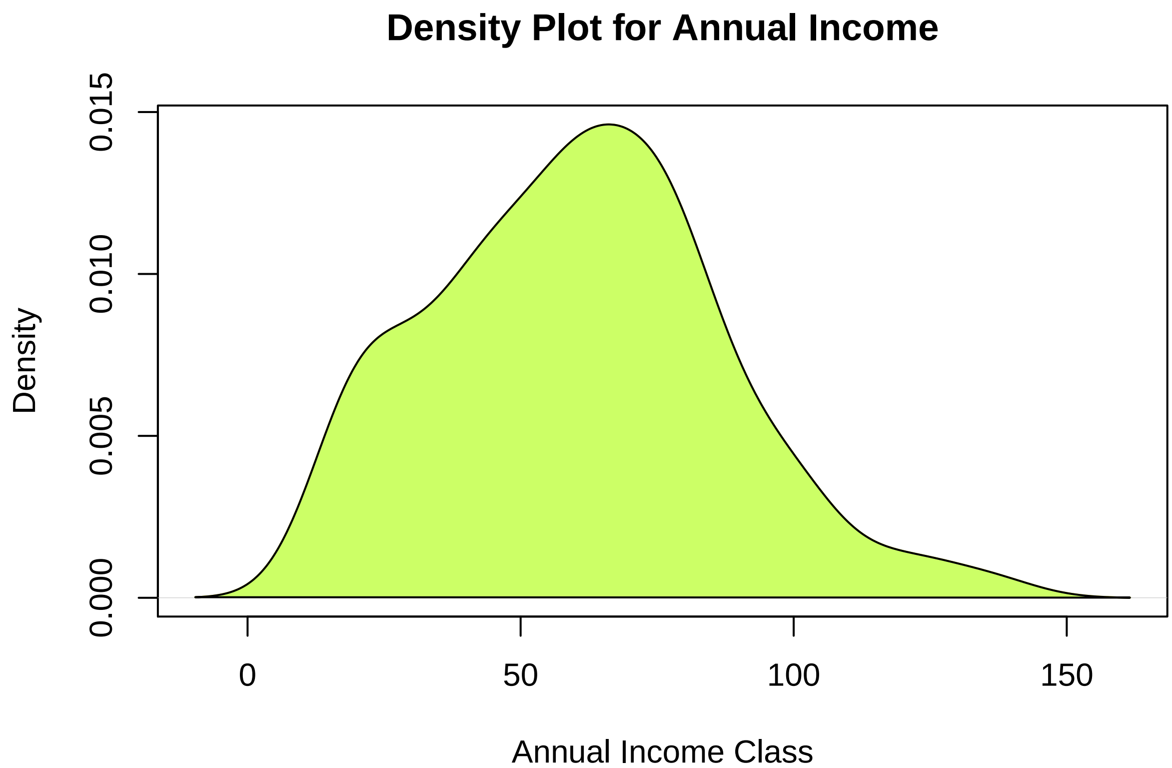
```
summary(customer_data$Annual.Income..k..)
```

```
##    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")
```



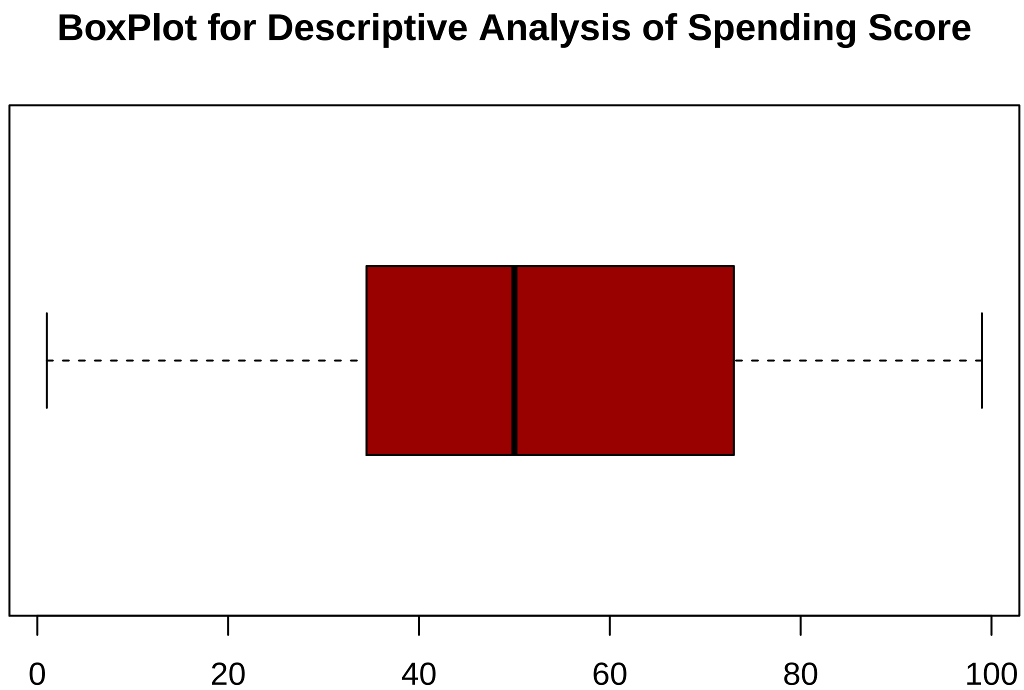
From the above descriptive analysis, we conclude that the minimum annual income of the customers is 15 and the maximum income is 137. People earning an average income of 70 have the highest frequency count in our histogram distribution. The average salary of all the customers is 60.56. In the Kernel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

#Analyzing Spending Score of the Customers

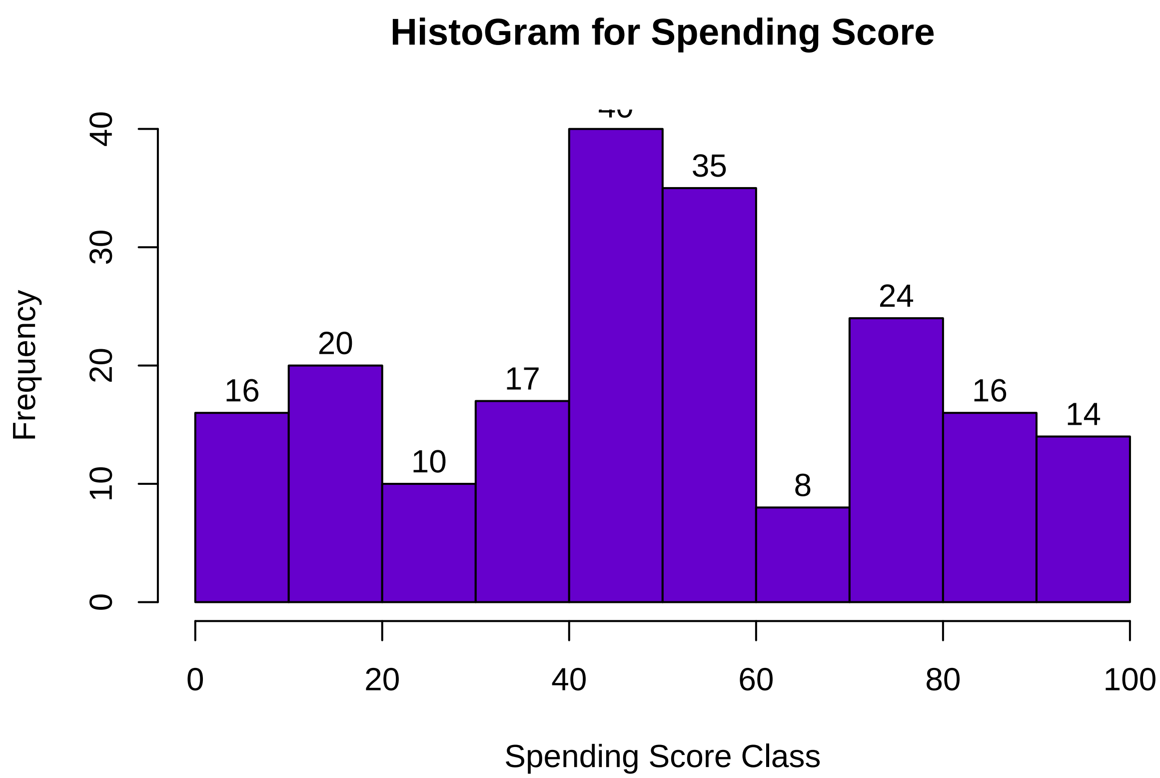
```
summary(customer_data$Spending.Score..1.100.)
```

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

```
boxplot(customer_data$Spending.Score..1.100.,
   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)
```



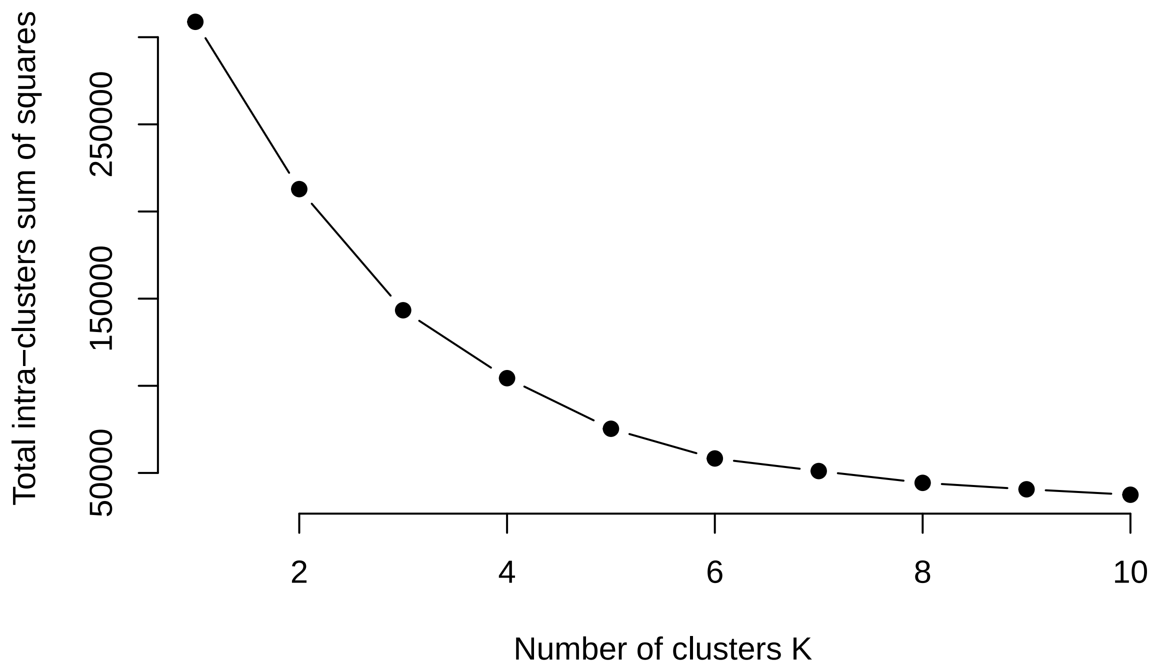
From the above, we see that the minimum spending score is 1, maximum is 99 and the average is 50.20. We can see Descriptive Analysis of Spending Score is that Min is 1, Max is 99 and avg. is 50.20. From the histogram, we conclude that customers between class 40 and 50 have the highest spending score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

#K-means Algorithm The Kmeans algorithm is an iterative algorithm that attempts to partition the dataset into K distinct non-overlapping subgroups (clusters), with each data point belonging to only one group. It attempts to keep intra-cluster data points as close as possible while making clusters as separate (far) as possible. It assigns data points to clusters in such a way that the sum of the squared distances between the data points and the cluster's centroid (the arithmetic mean of all the data points in that cluster) is as small as possible. The lower the heterogeneity between clusters, the more homogeneous (similar) the data points within the same cluster are. The way kmeans algorithm works is as follows: 1. Specify number of clusters K. 2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement. 3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing. 4. Compute the sum of the squared distance between data points and all centroids. 5. Assign each data point to the closest cluster (centroid). 6. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

To determine optimal Clusters we use 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
```

```r
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")
```



From the above graph, we conclude that 4 is the appropriate number of clusters since it seems to be appearing at the bend in the elbow plot.

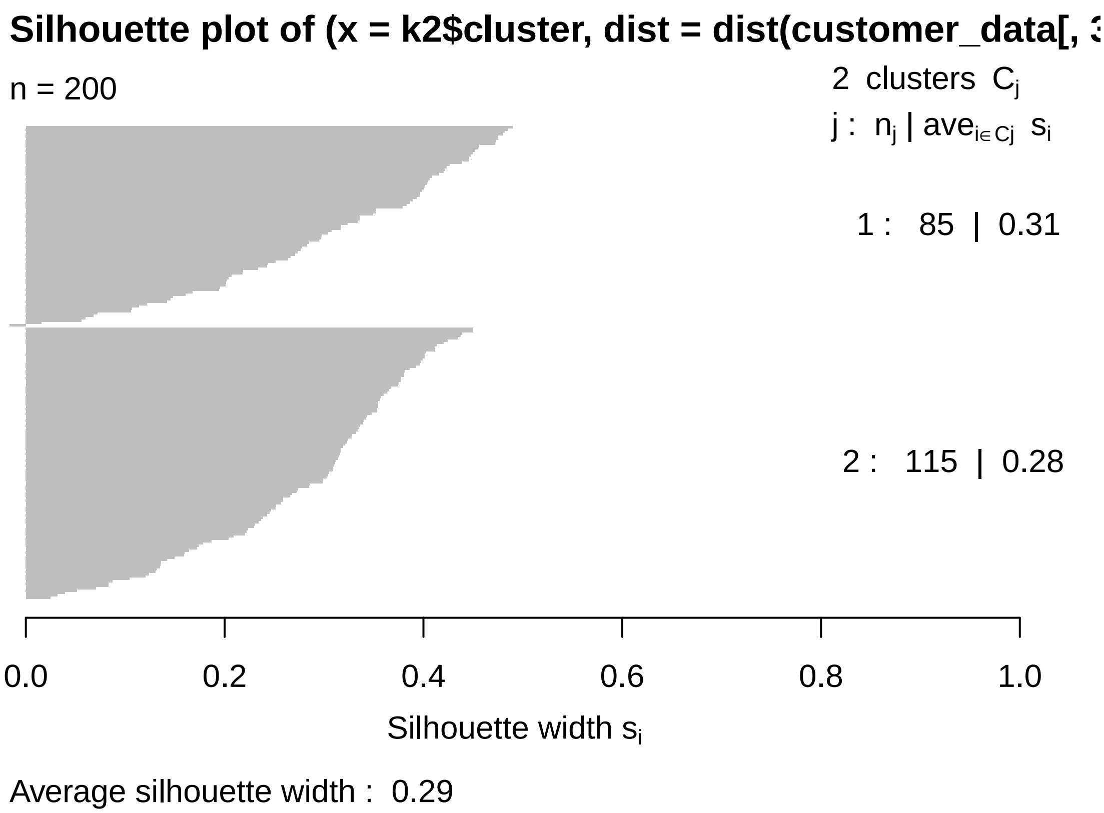
#Average Silhouette Method We may assess the quality of our clustering process using the average silhouette approach. We may use this to assess how well the data item fits into the cluster. A high average silhouette width indicates that we have strong clustering. The average silhouette equation computes the average of silhouette observations for various k values. With the optimal number of k clusters, one can maximize the average silhouette over significant values for k clusters.

```r
library(cluster)
#install.packages("gridExtra")
library(gridExtra)
```

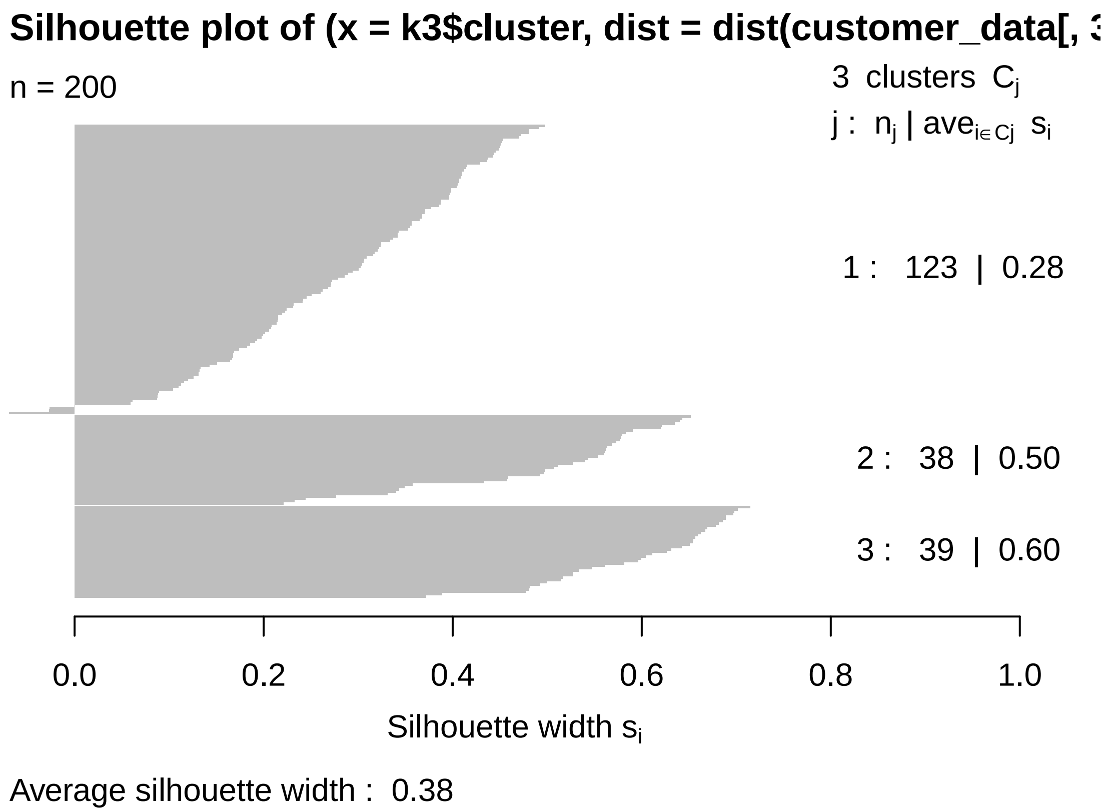
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
```

```
##
##     combine
```

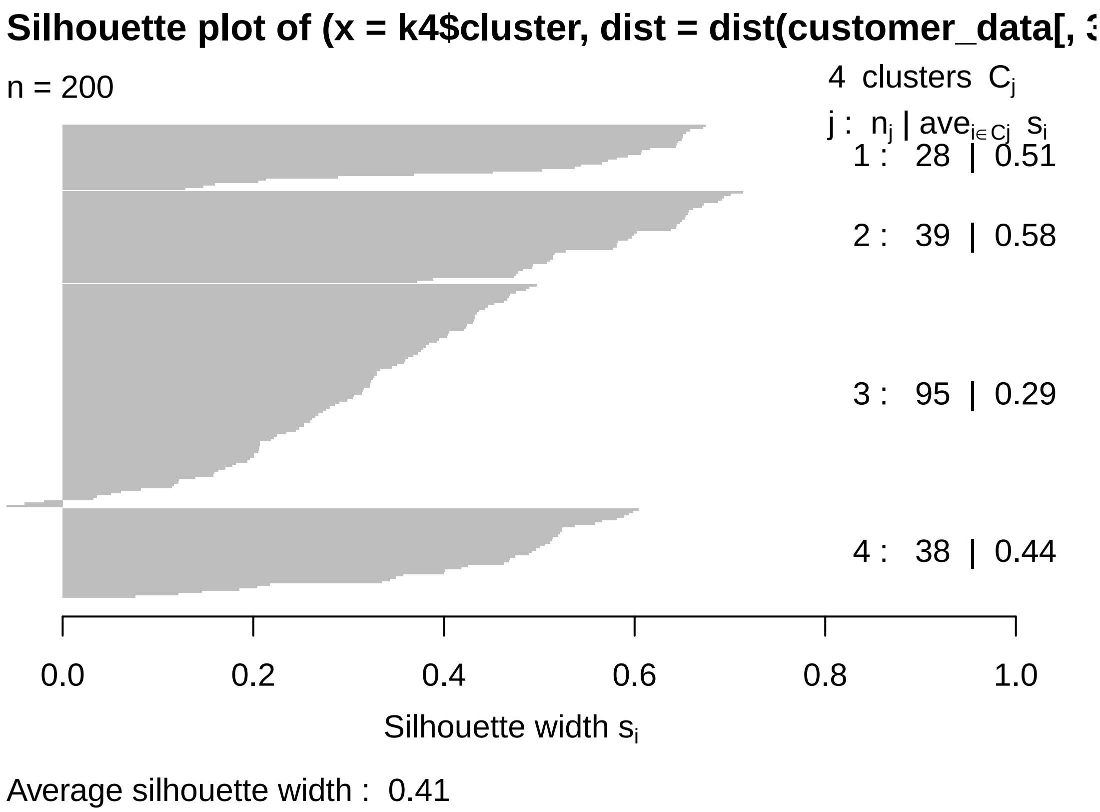
```r
library(grid)


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")))
```



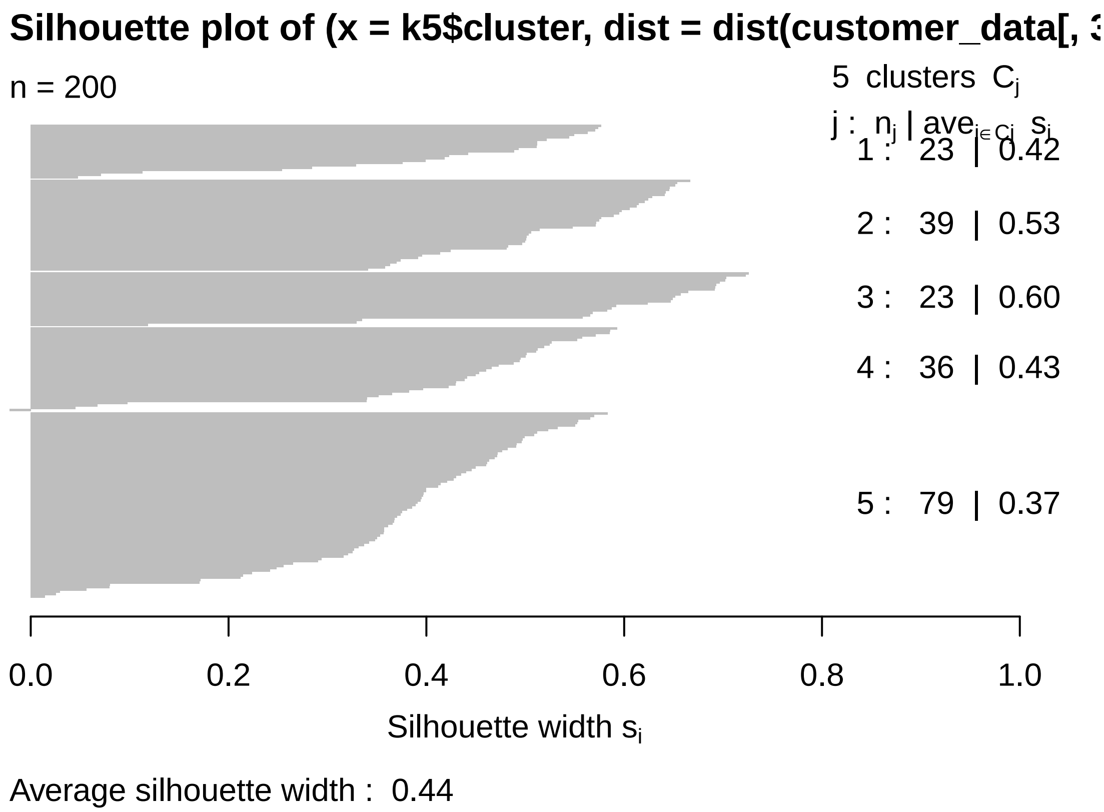

```r
k3<-kmeans(customer_data[,3:5],3,iter.max=100,nstart=50,algorithm="Lloyd")
s3<-plot(silhouette(k3$cluster,dist(customer_data[,3:5],"euclidean")))
```



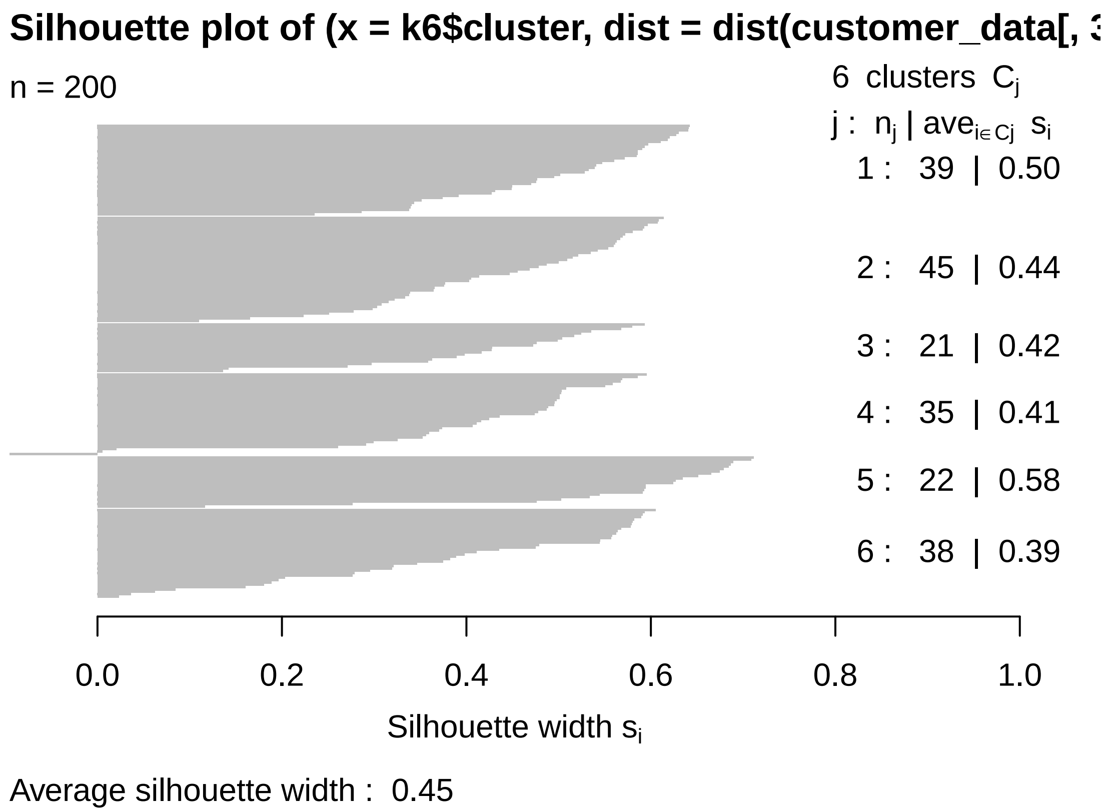

```
k4<-kmeans(customer_data[,3:5],4,iter.max=100,nstart=50,algorithm="Lloyd")
s4<-plot(silhouette(k4$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
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")))
```

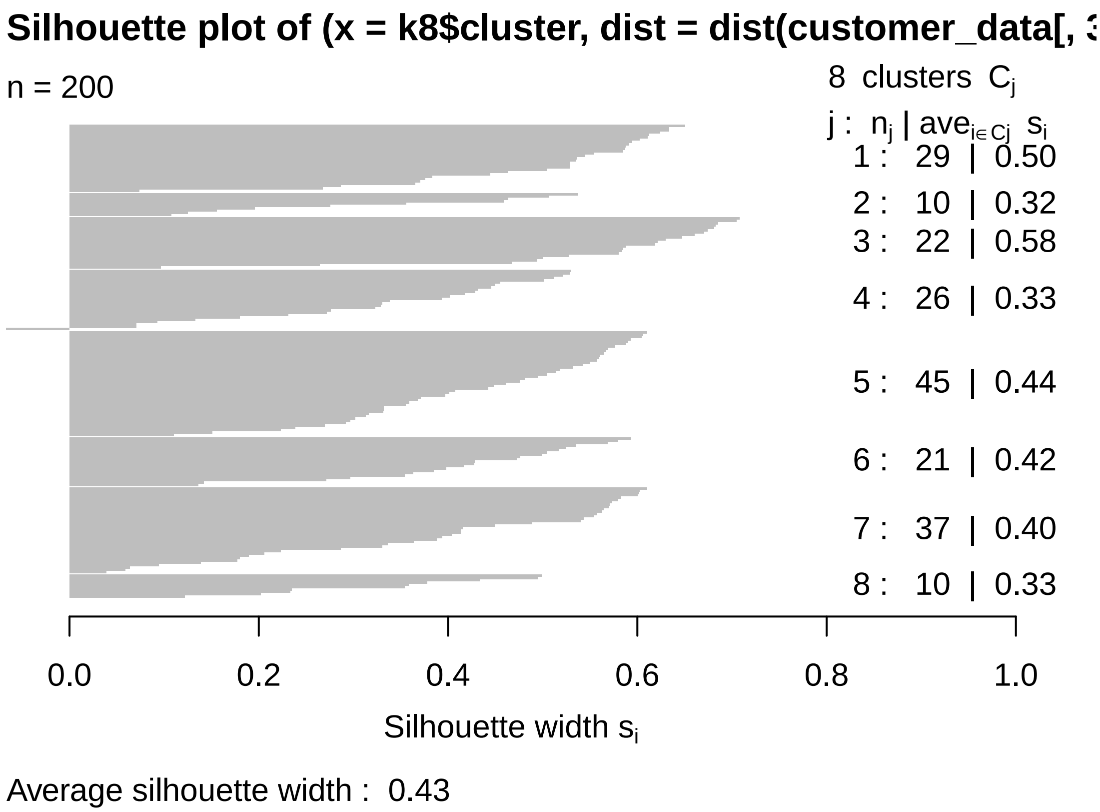



```
k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
s6<-plot(silhouette(k6$cluster,dist(customer_data[,3:5],"euclidean")))
```

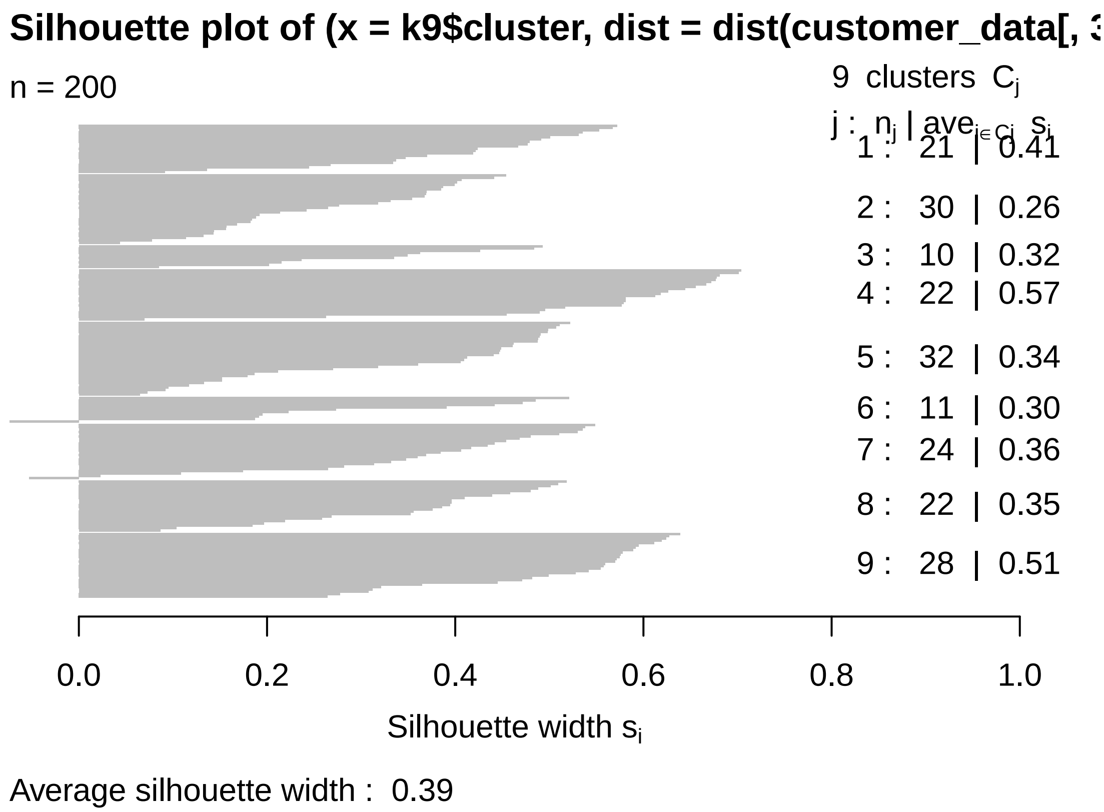



```
k7<-kmeans(customer_data[,3:5],7,iter.max=100,nstart=50,algorithm="Lloyd")
s7<-plot(silhouette(k7$cluster,dist(customer_data[,3:5],"euclidean")))
```

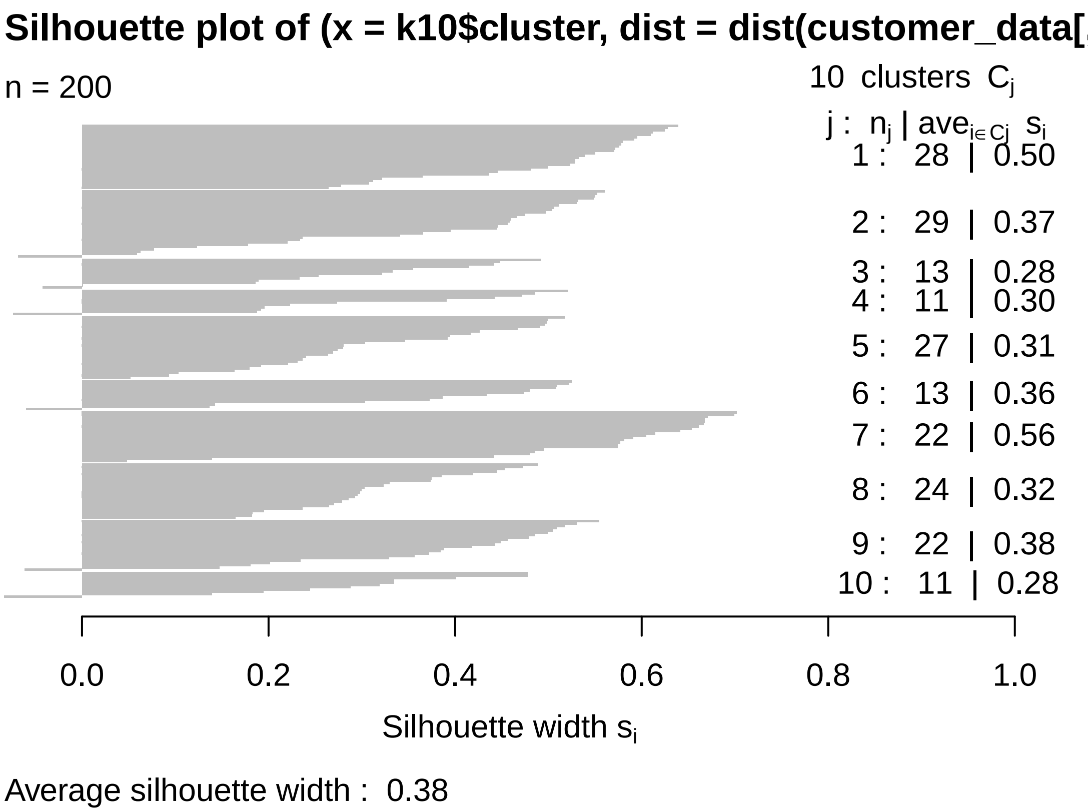

```
k8<-kmeans(customer_data[,3:5],8,iter.max=100,nstart=50,algorithm="Lloyd")
s8<-plot(silhouette(k8$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
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")))
```



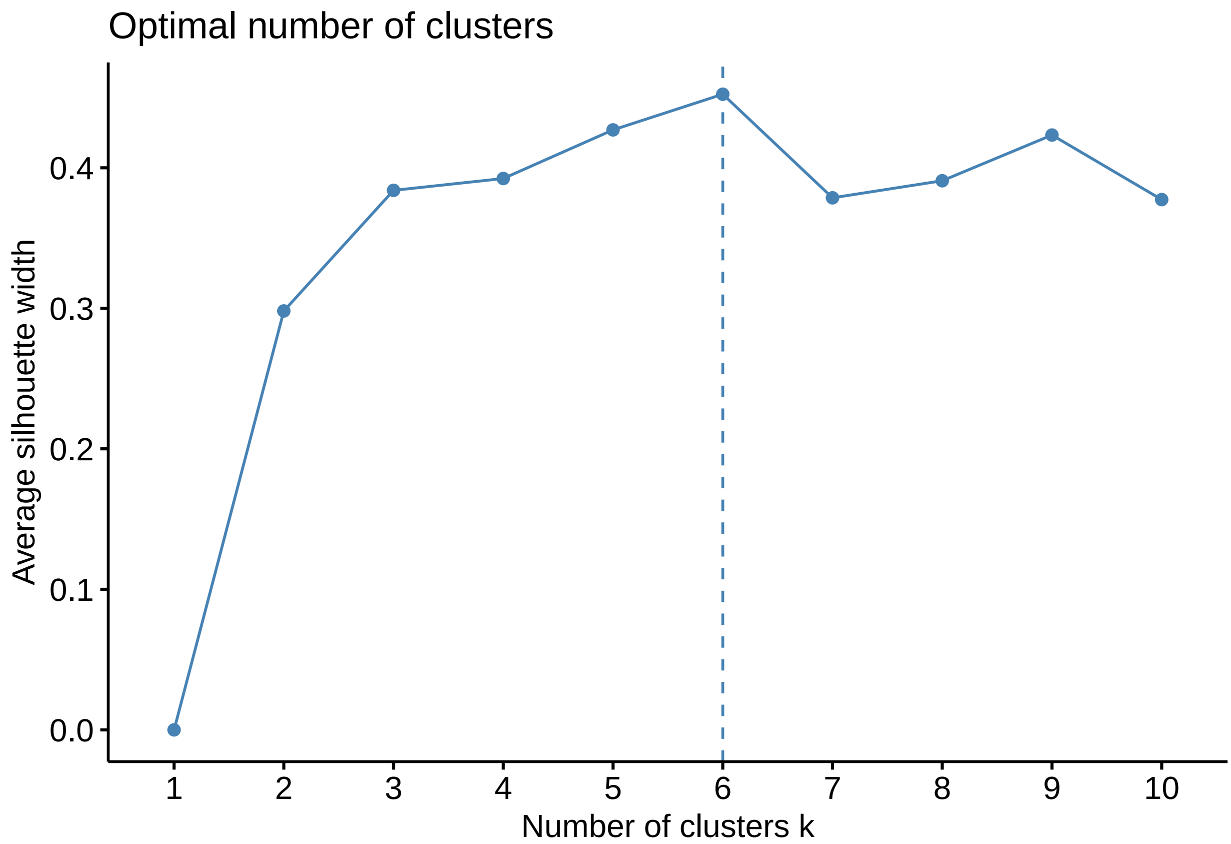
Determine and visualize the optimal number of clusters

```
#install.packages("NbClust")
library(NbClust)
#install.packages("factoextra")
library(factoextra)
```

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

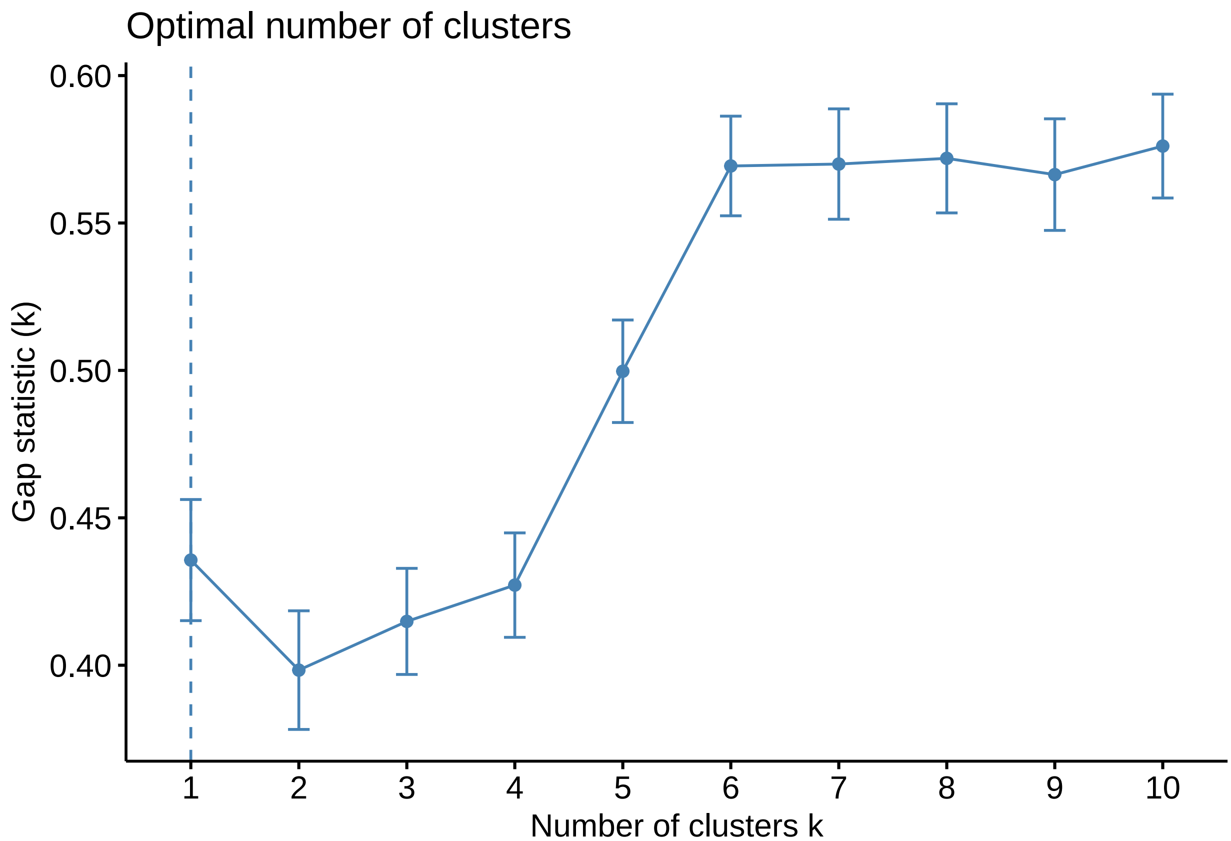
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

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



#Gap Statistic Method Another method to find the optimal cluster size

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

```
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

#Visualizing the Clustering Results using the First Two Principle Components

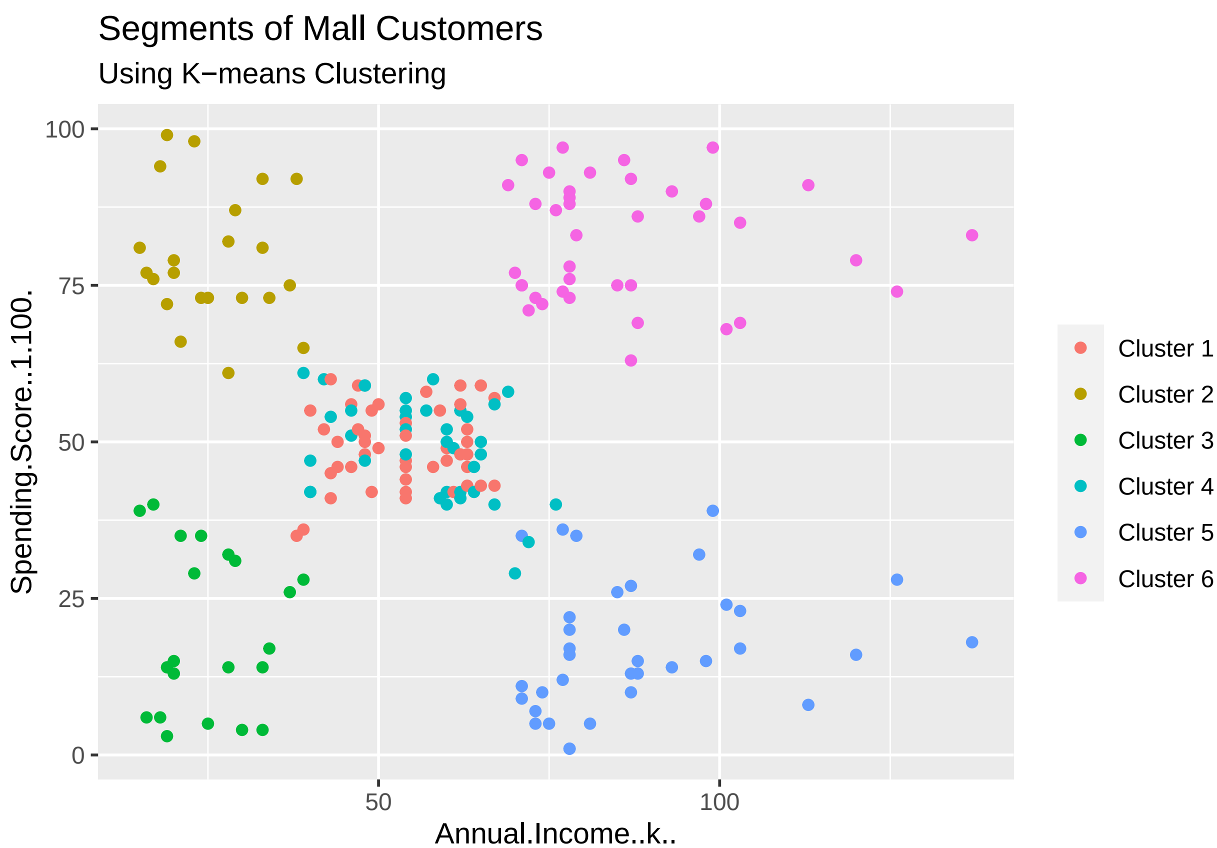
```
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)
```

```
## Importance of components:
##                            PC1     PC2     PC3
## Standard deviation     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]
```

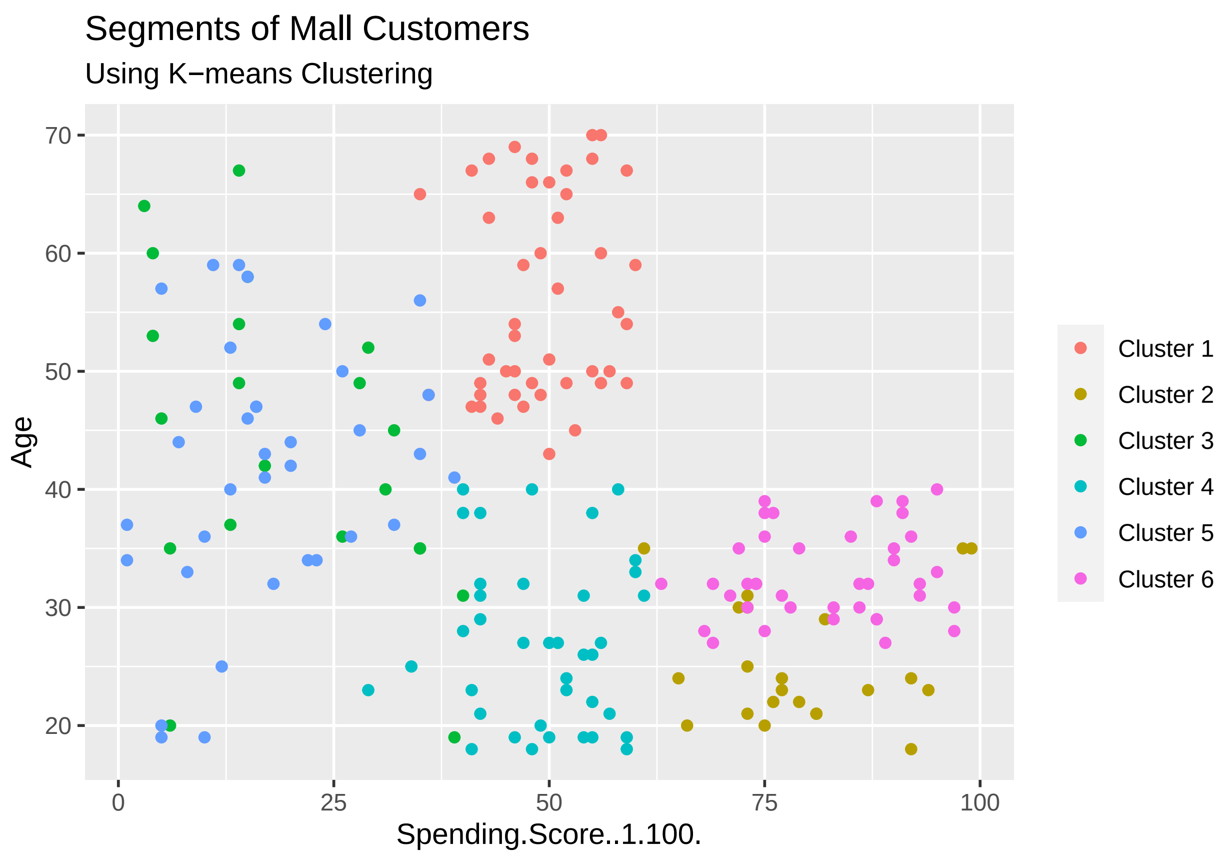
```
##                               PC1        PC2
## Age                     0.1889742 -0.1309652
## Annual.Income..k..     -0.5886410 -0.8083757
## Spending.Score..1.100. -0.7859965  0.5739136
```

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

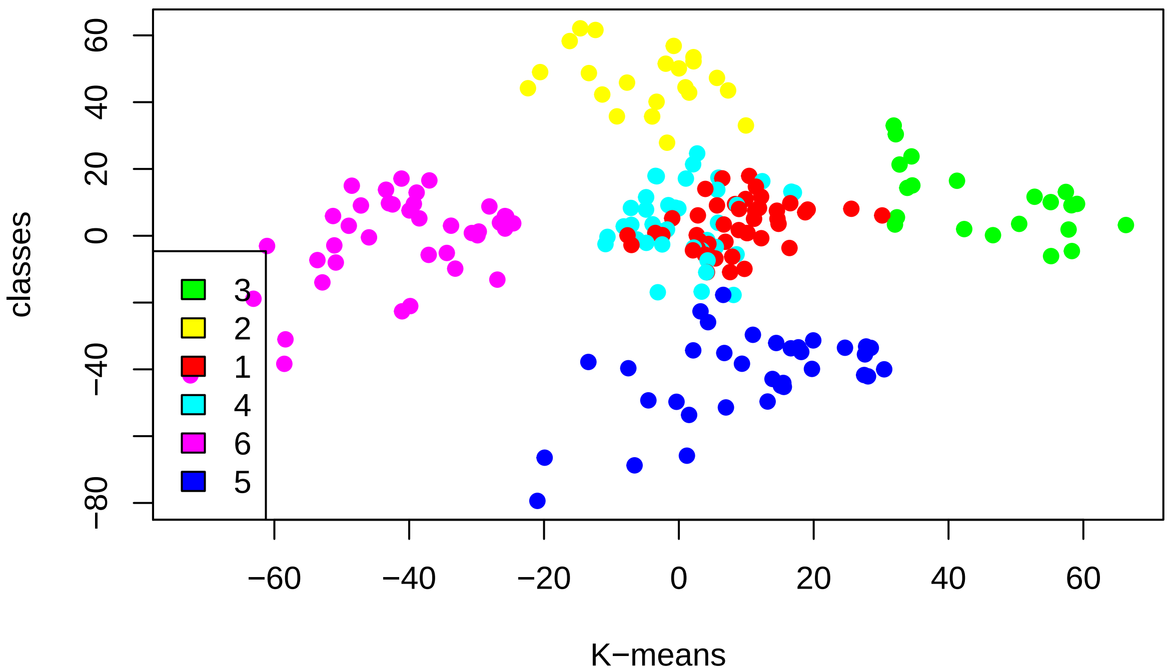


Cluster 6 and 4 – These clusters represent the customer_data with the medium income salary as well as the medium annual spend of salary. Cluster 1 – This cluster represents the customer_data having a high annual income as well as a high annual spend. Cluster 3 – This cluster denotes the customer_data with low annual income as well as low yearly spend of income. Cluster 2 – This cluster denotes a high annual income and low yearly spend. Cluster 5 – This cluster represents a low annual income but its 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", "Cluster 4", "Cluster 5","Cluster
  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)))
```



Cluster 4 and 1 – These two clusters consist of customers with medium PCA1 and medium PCA2 score. Cluster 6 – This cluster represents customers having a high PCA2 and a low PCA1. Cluster 5 – In this cluster, there are customers with a medium PCA1 and a low PCA2 score. Cluster 3 – This cluster comprises of customers with a high PCA1 income and a high PCA2. Cluster 2 – This comprises of customers with a high PCA2 and a medium annual spend of income.

We can better understand the variables with the assistance of clustering, prompting us to make more informed decisions. Companies will release products and services that attract consumers based on various criteria such as salary, age, purchasing habits, and so on with the identification of customers. Furthermore, more nuanced trends, such as product reviews, may be considered for improved segmentation.