Customer Segmentation Analysis:

Pritam Majumder

2024-04-01

Customer Segmentation Project in R.

Customer Segmentation is one the most important applications of unsupervised learning. Using clustering techniques, companies can identify the several segments of customers allowing them to target the potential user base. In this machine learning project, we will make use of K-means clustering which is the essential algorithm for clustering unlabeled dataset.

What is Customer Segmentation?

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.

Implementing Customer Segmentation in R?

In the first step of this data science project, we will perform data exploration.

We will import the essential packages required for this role and then read our data.

Finally, we will go through the input data to gain necessary insights about it.

Libraries

```
# Load necessary libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔dplyr     1.1.4     ✔readr     2.1.5
## ✔forcats   1.0.0     ✔stringr   1.5.1
## ✔ggplot2   3.4.4     ✔tibble    3.2.1
## ✔lubridate 1.9.3     ✔tidyr     1.3.1
## ✔purrr     1.0.2
```

```r
library(cluster)
library(factoextra)
```

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

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

```r
library(animation)
```

```
## Warning: package 'animation' was built under R version 4.3.3
```

Load the data

```r
# Load the data
customer_data <- read.csv("Supermarket _Customers.csv", header = TRUE)

head(customer_data)
```

```
##   CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100.
## 1          1   Male  23                 15                     39
## 2          2   Male  21                 15                     81
## 3          3 Female  25                 16                      6
## 4          4 Female  28                 16                     77
## 5          5 Female  31                 17                     40
## 6          6 Female  22                 17                     76
```

Dimension of our Data_frame

```r
dim(customer_data)
```

```
## [1] 200   5
```

```r
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  23 21 25 28 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 ...
```

```r
names(customer_data)
```

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

Summery

```r
head(customer_data)
```

```
##   CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100.
## 1          1   Male  23                 15                      39
## 2          2   Male  21                 15                      81
## 3          3 Female  25                 16                       6
## 4          4 Female  28                 16                      77
## 5          5 Female  31                 17                      40
## 6          6 Female  22                 17                      76
```

```
ncol(customer_data)
```

```
## [1] 5
```

We will now display the first six rows of our dataset using the head() function and use the summary() function to Age.

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.75   36.00   38.92   49.00   70.00
```

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

```
## [1] 13.8895
```

We will now display the first six rows of our dataset using the head() function and use the summary() function to Annual.Income..k..

```
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
```

We will now display the first six rows of our dataset using the head() function and use the summary() function to Spending.Score..1.100.

```
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
```

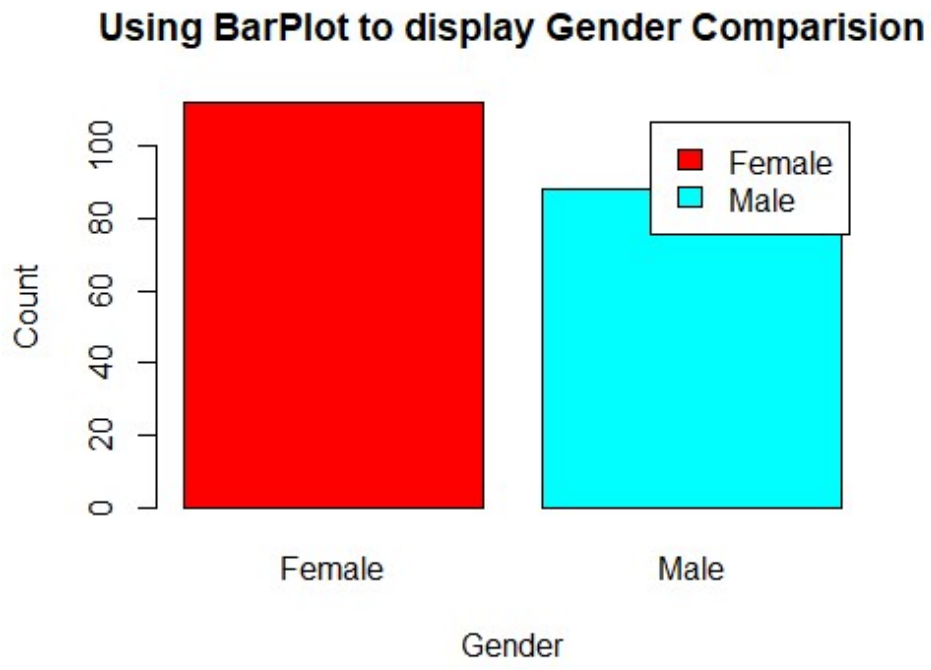
```
sd(customer_data$Spending.Score..1.100.)
```

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

Customer Gender Visualization

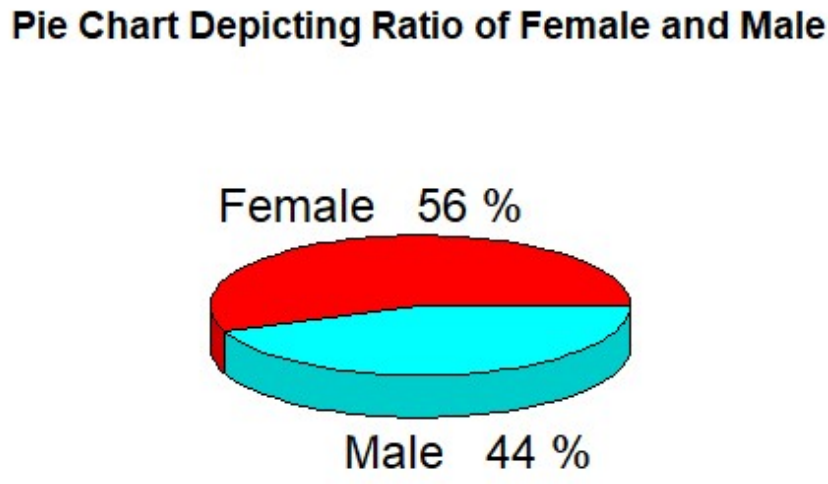
we will create a barplot and a piechart to show the gender distribution across our customer_data dataset.

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



Pie_chart Gender visualization.

```
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")
```



From the above graph, we conclude that the percentage of females is 56%, whereas the percentage of male in the customer dataset is 44%.

Visualization of Age Distribution

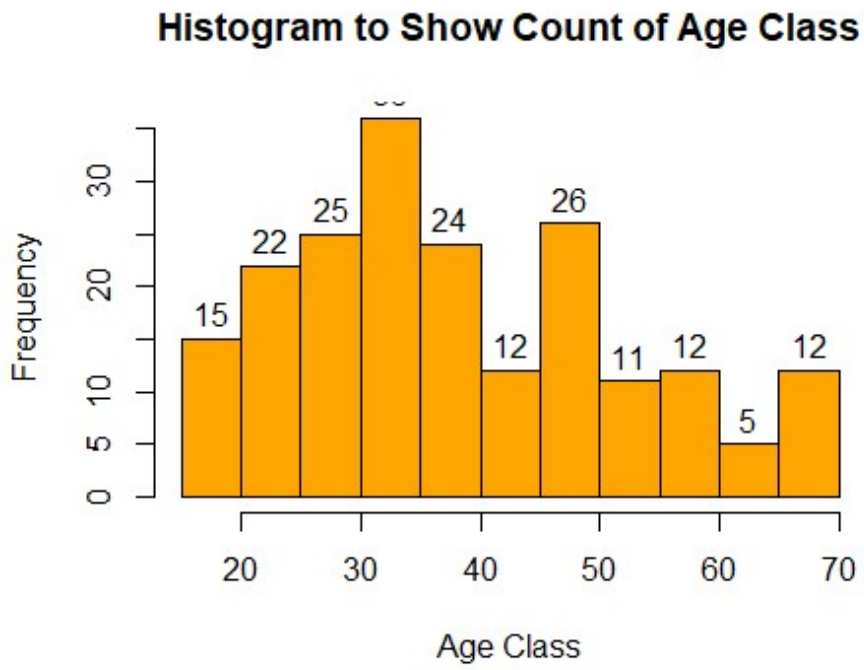
Let us plot a histogram to view the distribution to plot the frequency of customer ages. We will first proceed by taking summary of the Age variable.

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.75   36.00   38.92   49.00   70.00
```

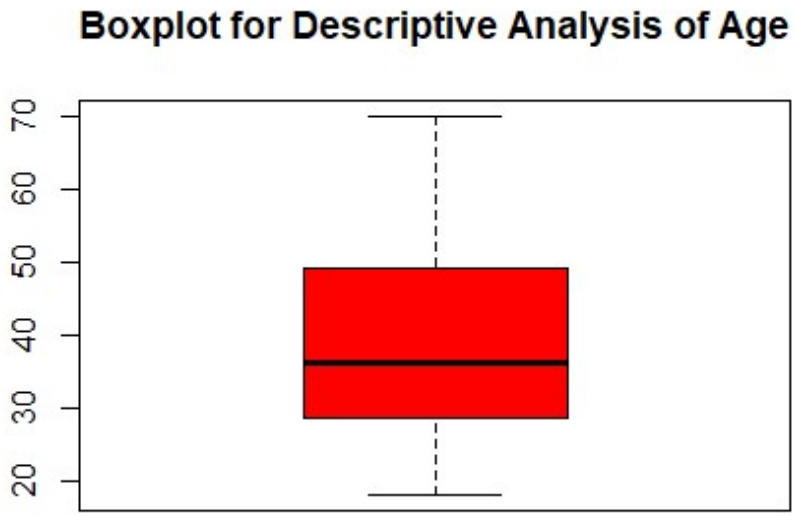
Histogram Visualization.

```
#visualization

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



Boxplot Visualization.

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



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.

Analysis of the Annual Income of the Customers

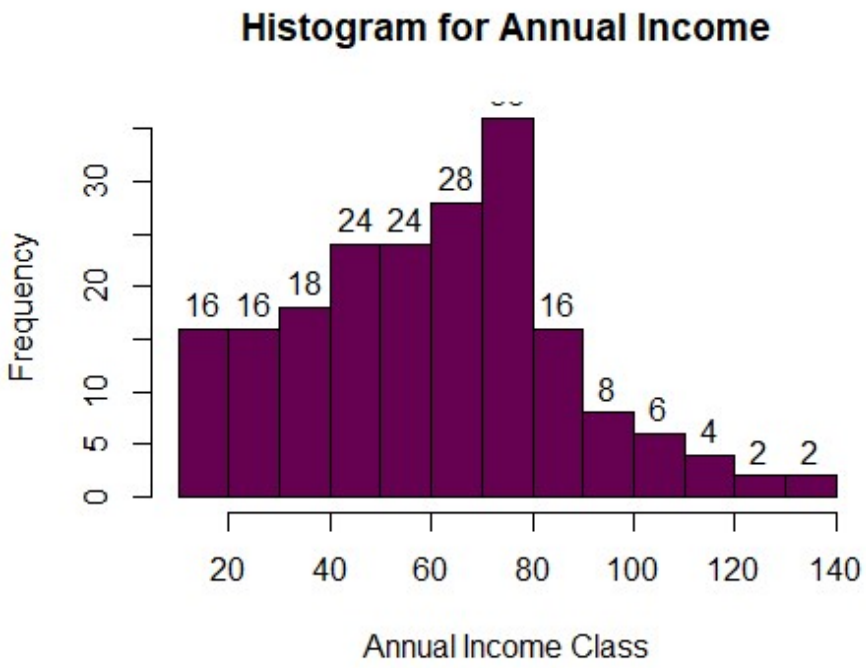
we will create visualizations to analyze the annual income of the customers. We will plot a histogram and then we will proceed to examine this data using a density plot.

Histogram for Annual Income.

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

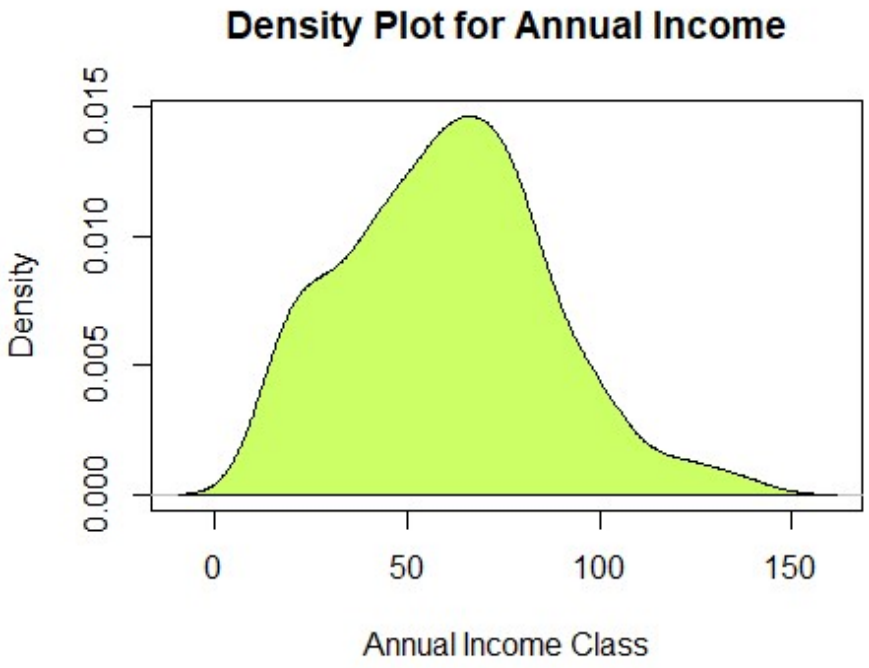
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   41.50   61.50   60.56   78.00  137.00
```

```
hist(customer_data$Annual.Income..k..,
  col="#660050",
  main="Histogram for Annual Income",
  xlab="Annual Income Class",
  ylab="Frequency",
  labels=TRUE)
```



Density Plot for Annual Income.

```
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.

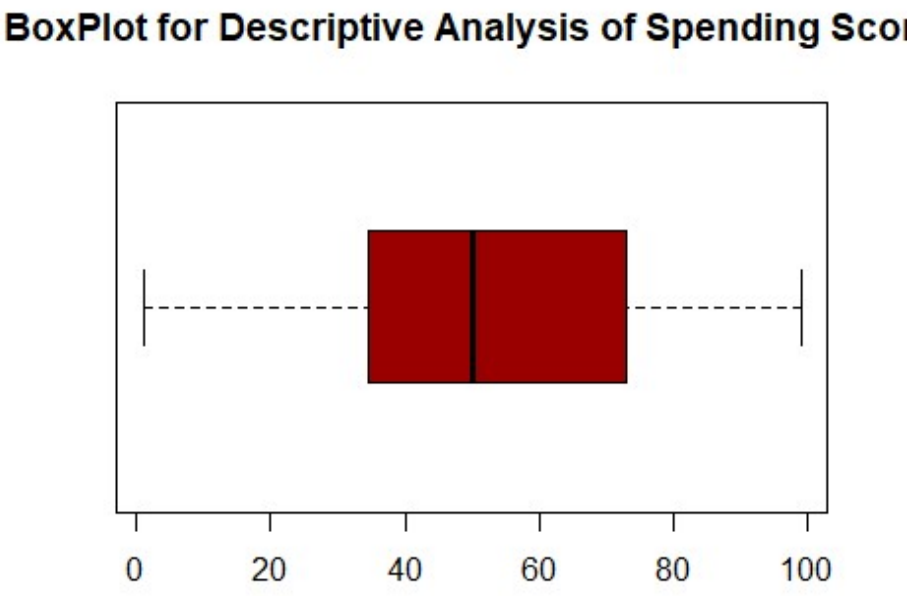
Analyzing Spending Score of the Customers

Boxplot for Spending Score

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

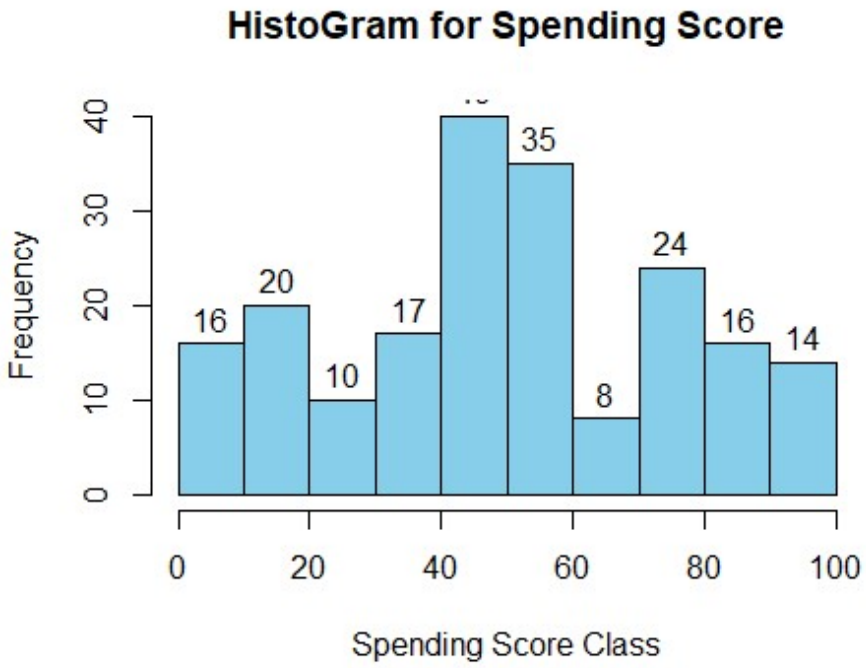
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   34.75   50.00   50.20   73.00   99.00
```

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



Histogram for Spending Score.

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



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.

While using the k-means clustering algorithm, the first step is to indicate the number of clusters (k) that we wish to produce in the final output. The algorithm starts by selecting k objects from dataset randomly that will serve as the initial centers for our clusters. These selected objects are the cluster means, also known as centroids. Then, the remaining objects have an assignment of the closest centroid. This centroid is defined by the Euclidean Distance present between the object and the cluster mean. We refer to this step as “cluster assignment”. When the assignment is complete, the algorithm proceeds to calculate new mean value of each cluster present in the data. After the recalculation of the centers, the observations are checked if they are closer to a different cluster. Using the updated cluster mean, the objects undergo reassignment. This goes on repeatedly through several iterations until the cluster assignments stop altering. The clusters that are present in the current iteration are the same as the ones obtained in the previous iteration.

Determining Optimal Clusters.

While working with clusters, you need to specify the number of clusters to use. You would like to utilize the optimal number of clusters. To help you in determining the optimal clusters, there are three popular methods –

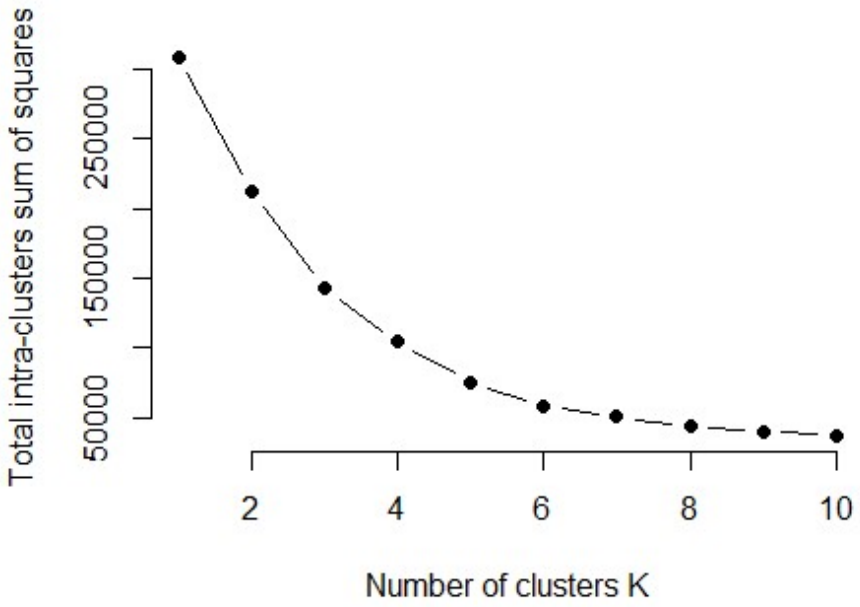
-> Elbow method

-> Silhouette method

-> Gap statistic

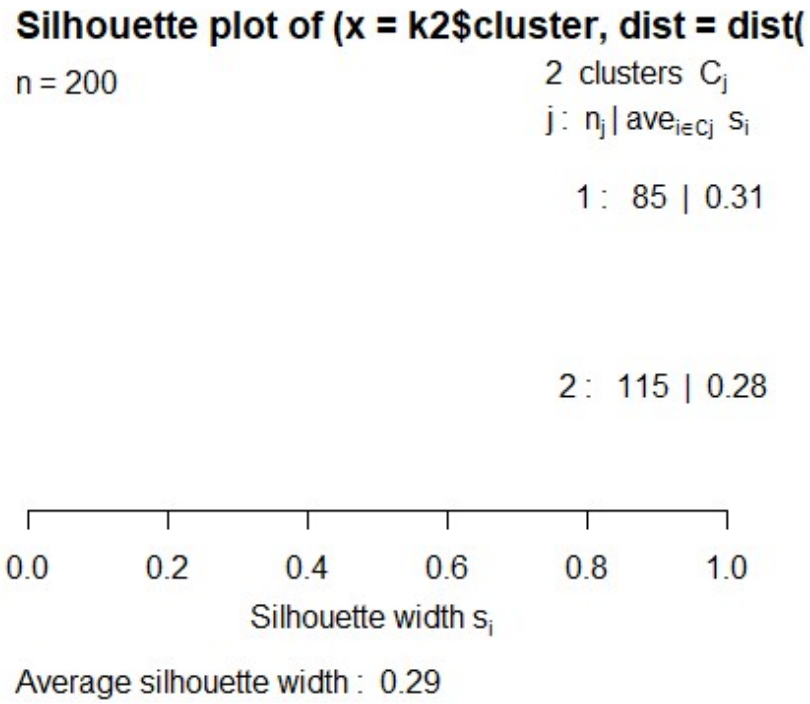
Elbow Method

```
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")
```

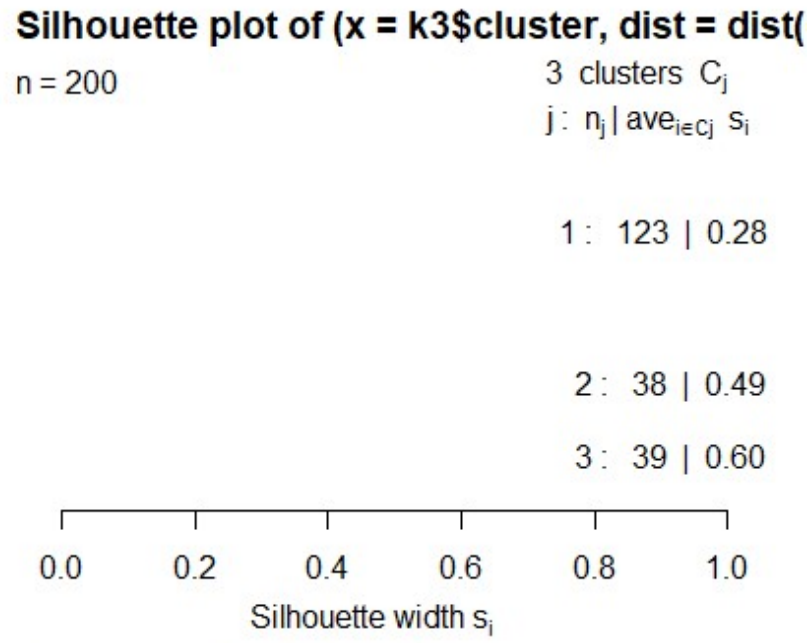


Silhouette plot

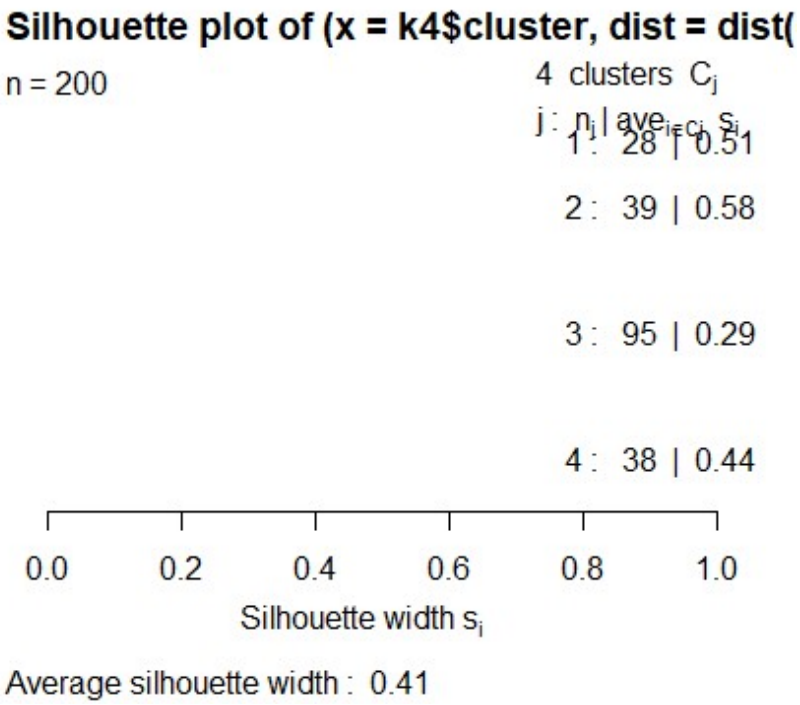
```
library(cluster)
library(gridExtra)

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

##
## Attaching package: 'gridExtra'

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

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")))
```



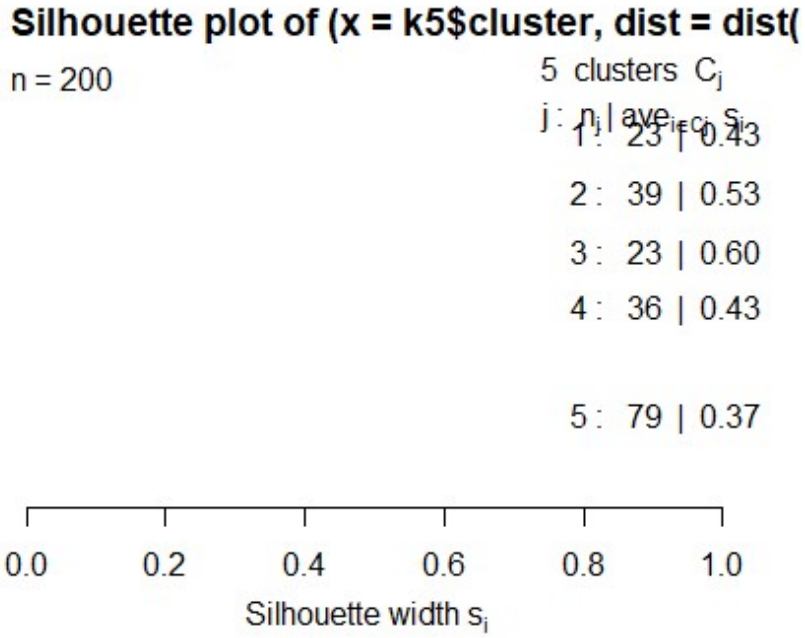

```
k3<-kmeans(customer_data[,3:5],3,iter.max=100,nstart=50,algorithm="Lloyd")
s3<-plot(silhouette(k3$cluster,dist(customer_data[,3:5],"euclidean")))
```



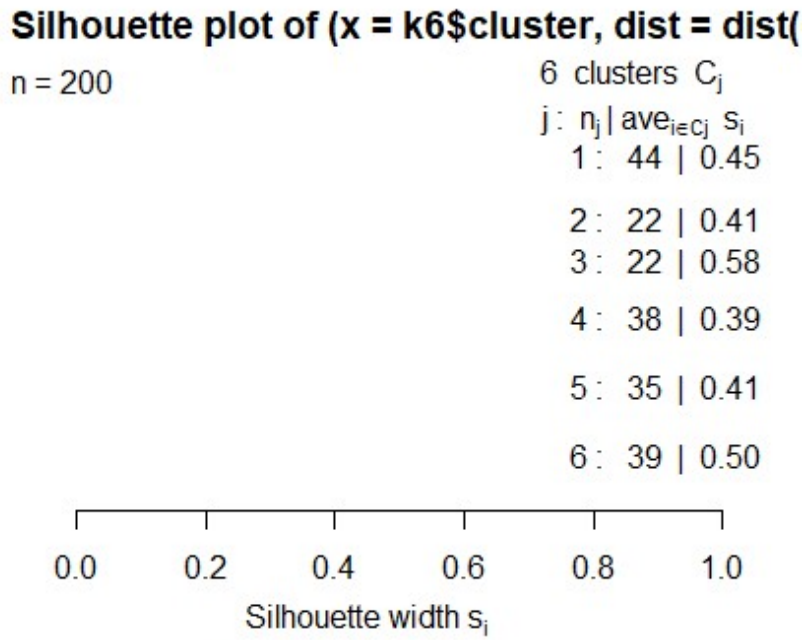
```
k4<-kmeans(customer_data[,3:5],4,iter.max=100,nstart=50,algorithm="Lloyd")
s4<-plot(silhouette(k4$cluster,dist(customer_data[,3:5],"euclidean")))
```



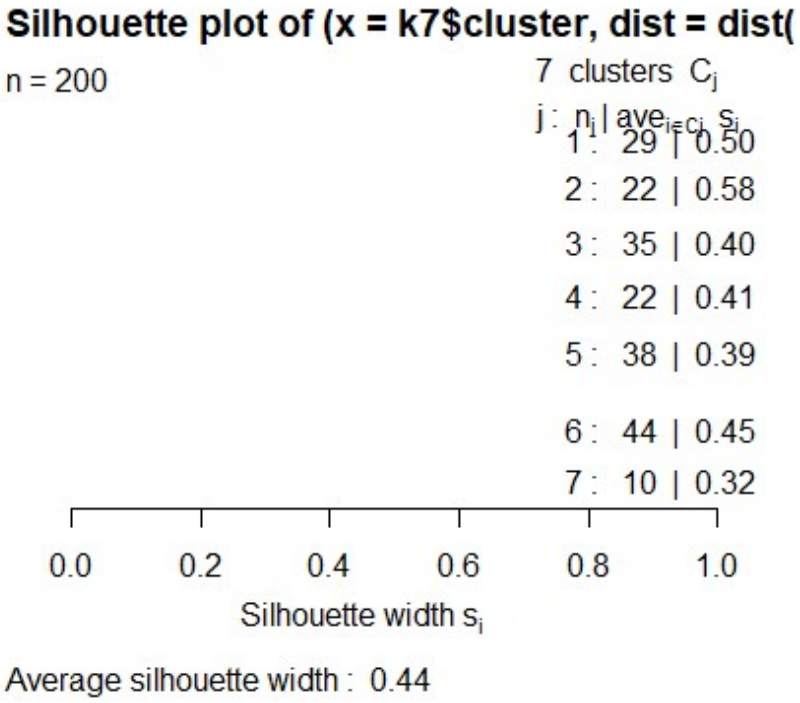

```
k5<-kmeans(customer_data[,3:5],5,iter.max=100,nstart=50,algorithm="Lloyd")
s5<-plot(silhouette(k5$cluster,dist(customer_data[,3:5],"euclidean")))
```



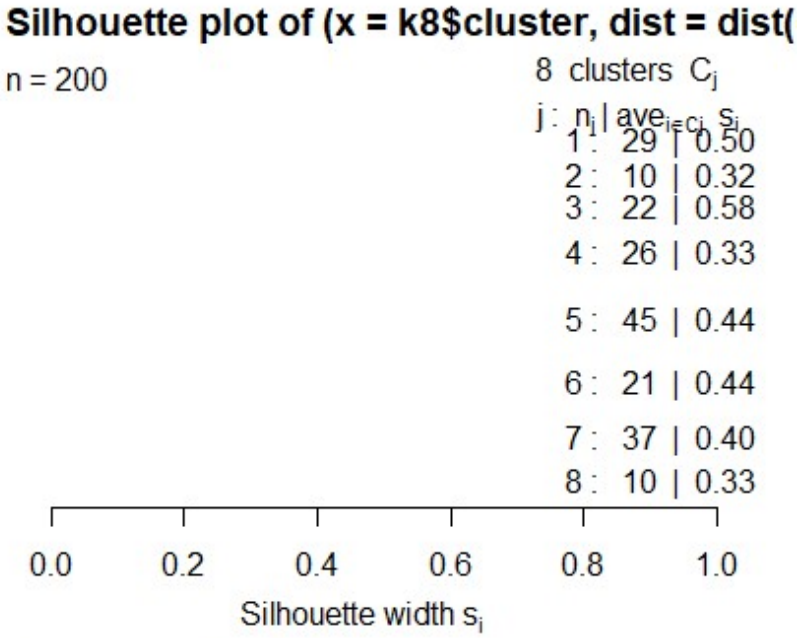

```r
k6<-kmeans(customer_data[,3:5],6,iter.max=100,nstart=50,algorithm="Lloyd")
s6<-plot(silhouette(k6$cluster,dist(customer_data[,3:5],"euclidean")))
```



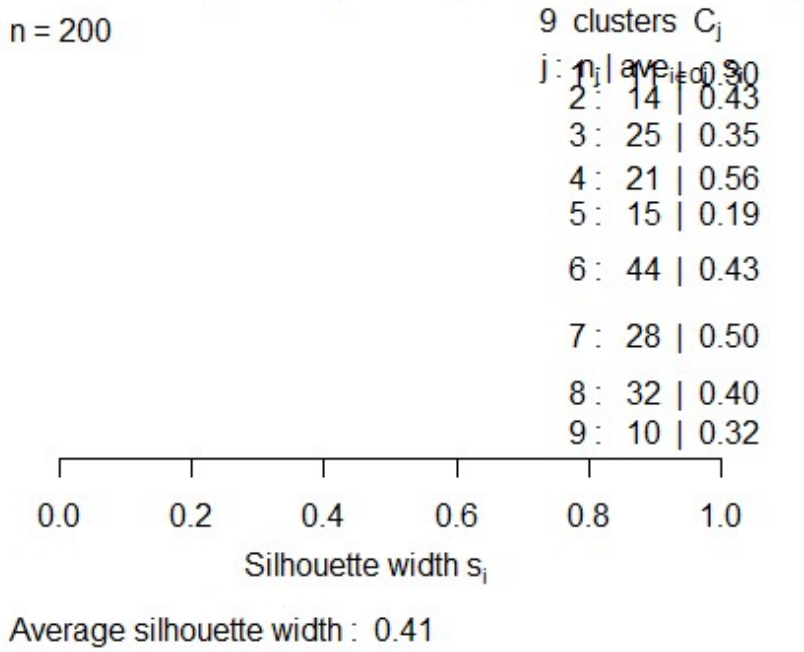
```
k7<-kmeans(customer_data[,3:5],7,iter.max=100,nstart=50,algorithm="Lloyd")
s7<-plot(silhouette(k7$cluster,dist(customer_data[,3:5],"euclidean")))
```



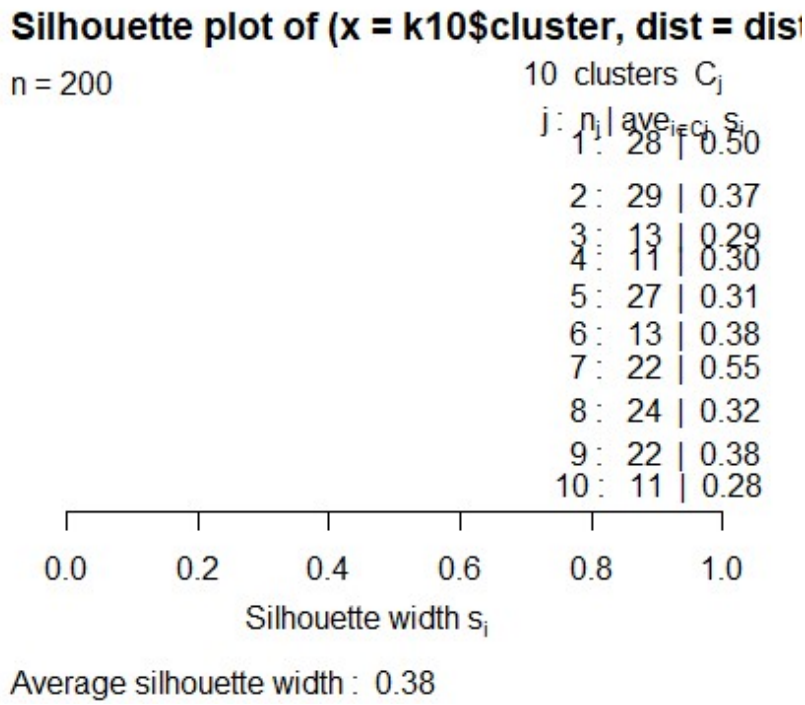

```
k8<-kmeans(customer_data[,3:5],8,iter.max=100,nstart=50,algorithm="Lloyd")
s8<-plot(silhouette(k8$cluster,dist(customer_data[,3:5],"euclidean")))
```




```
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")))
```

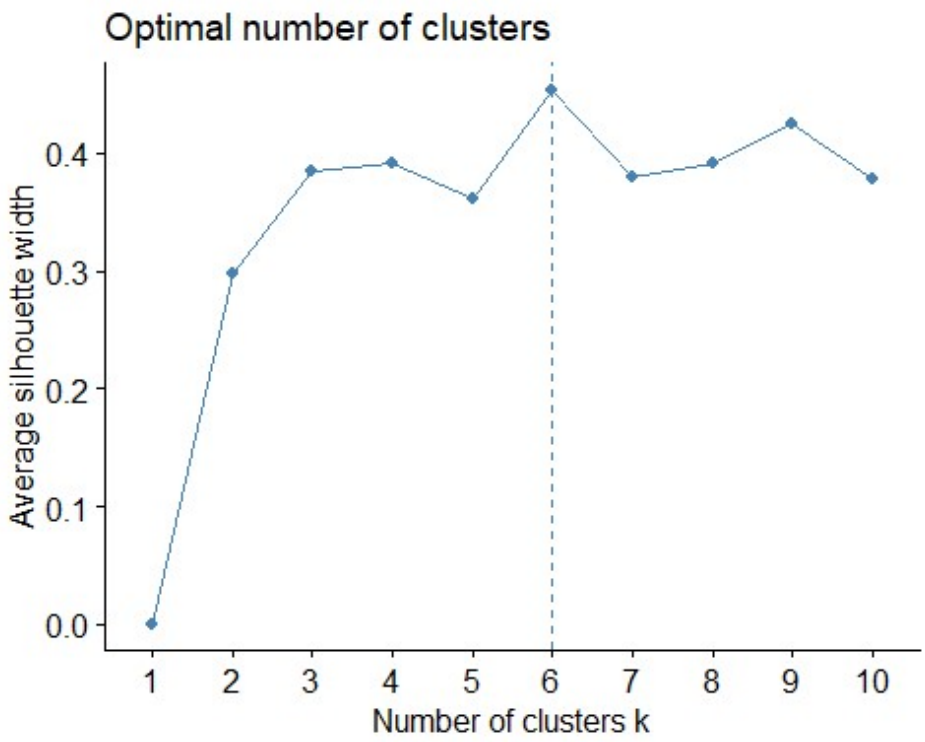


```
k10<-kmeans(customer_data[,3:5],10,iter.max=100,nstart=50,algorithm="Lloyd")
s10<-plot(silhouette(k10$cluster,dist(customer_data[,3:5],"euclidean")))
```



Optical Numbers of Clusters

```
library(NbClust)
library(factoextra)

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



Gap Statistic Method

-> We can use this method to any of the clustering method like K-means, hierarchical clustering etc. Using the gap statistic, one can compare the total intracluster variation for different values of k along with their expected values under the null reference distribution of data. With the help of Monte Carlo simulations, one can produce the sample dataset. For each variable in the dataset, we can calculate the range between min(xi) and max (xj) through which we can produce values uniformly from interval lower bound to upper bound.

For computing the gap statistics method we can utilize the clusGap function for providing gap statistic as well as standard error for a given output.

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

## K-means clustering with 6 clusters of sizes 22, 35, 39, 44, 22, 38
##
## Cluster means:
##        Age Annual.Income..k.. Spending.Score..1.100.
## 1 44.72727           25.77273               20.27273
## 2 41.68571           88.22857               17.28571
## 3 32.69231           86.53846               82.12821
## 4 56.34091           53.70455               49.38636
## 5 25.50000           25.72727               79.36364
```

```
## 6 27.00000           56.65789               49.13158
##
## Clustering vector:
##   [1] 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5 1 5
1 5 1
##  [38] 5 1 5 4 5 1 6 1 5 4 6 6 6 4 6 6 4 4 4 4 4 6 4 4 6 4 4 4 6 4 4 6 6 4
4 4 4
##  [75] 4 6 4 6 6 4 4 6 4 4 6 4 4 6 6 4 4 6 4 6 6 6 4 6 4 6 6 4 4 6 4 6 4 4
4 4 4
## [112] 6 6 6 6 6 4 4 4 4 6 6 6 3 6 3 2 3 2 3 2 3 6 3 2 3 2 3 2 3 2 3 6 3 2
3 2 3
## [149] 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 2
## [186] 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3
##
## Within cluster sum of squares by cluster:
## [1]  7780.591 16690.857 13972.359  7607.477  4100.955  7742.895
##  (between_SS / total_SS =  81.2 %)
##
## 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)
```

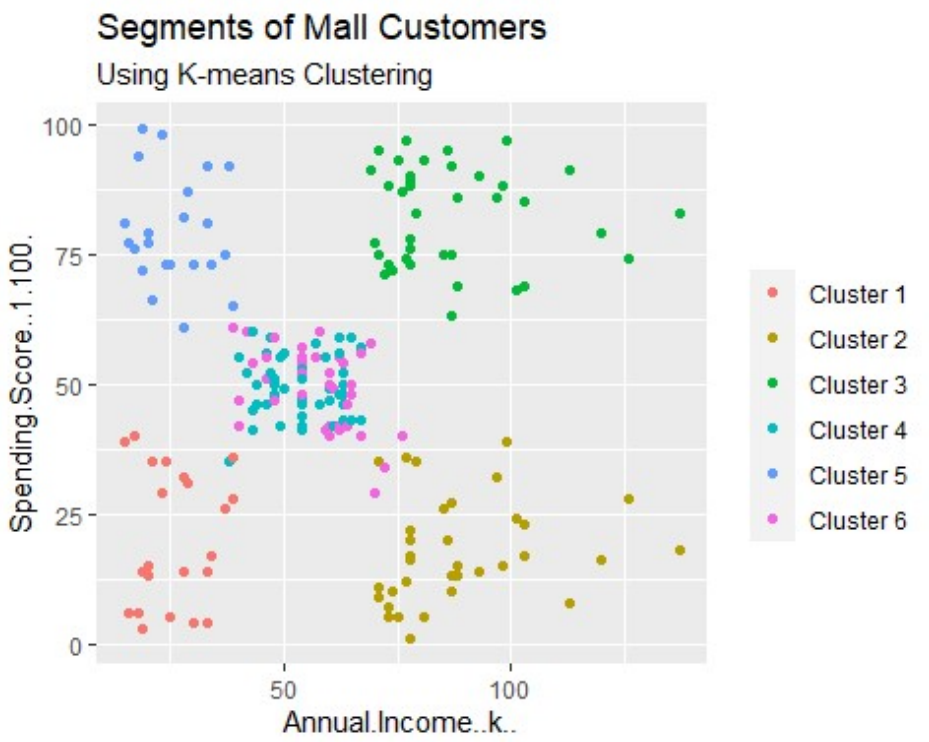
```
## Importance of components:
##                             PC1     PC2     PC3
## Standard deviation     26.4781 26.1483 12.8368
## Proportion of Variance  0.4524  0.4412  0.1063
## Cumulative Proportion   0.4524  0.8937  1.0000
```

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

```
##                                PC1        PC2
## Age                     0.1911852 -0.1282256
## Annual.Income..k..     -0.5961364 -0.8028031
## Spending.Score..1.100. -0.7797882  0.5822932
```

```
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")
```



From the above visualization, we observe that there is a distribution of 6 clusters as follows –

-> 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.

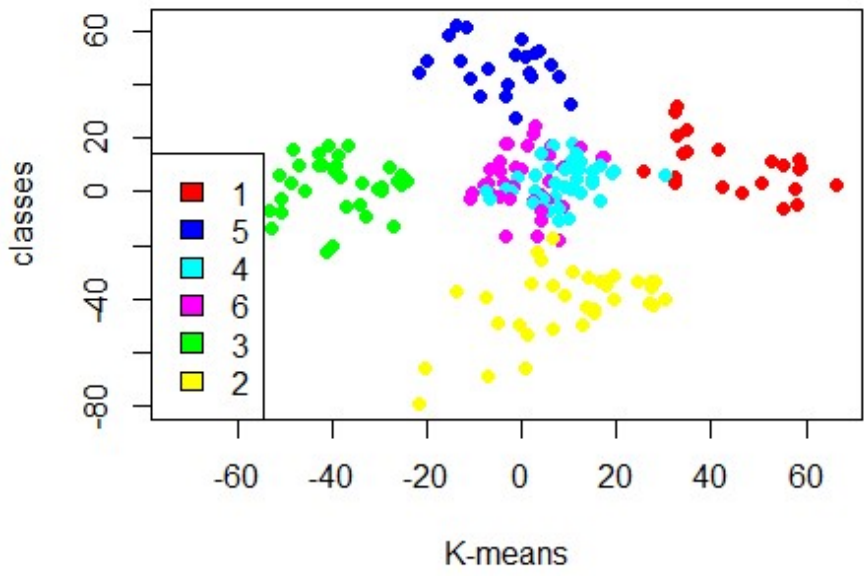
-> Cluster 3 – This cluster denotes the customer_data with low annual income as well as low yearly spend of income.

-> 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.

```
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.*

With the help of clustering, we can understand the variables much better, prompting us to take careful decisions. With the identification of customers, companies can release products and services that target customers based on several parameters like income, age, spending patterns, etc. Furthermore, more complex patterns like product reviews are taken into consideration for better segmentation.