Customer Data Analysis

Sahil Thorat

2024-07-25

Loading and Exploring the Data

```
getwd()
```

```
## [1] "D:/Projects/Data_Projects_AIO/Customer_Segmentation/Code"
```

```
library(readr)
# Load the data
customer_data=read.csv("D:/Projects/Data_Projects_AIO/Customer_Segmentation/data/Mall_Customers.csv")

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
```

```
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 ...
```

```
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
```

```
names(customer_data)
```

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

```
# Rename the columns
names(customer_data)=c("CustomerID", "Gender", "Age", "AnnualIncome", "SpendingScore")

# Standard deviation for Age
sd(customer_data$Age)
```

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

```
# Summary and SD statistics for Annual Income
summary(customer_data$AnnualIncome)
```

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

```
sd(customer_data$AnnualIncome)
```

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

```
# Summary and SD statistics for Spending Score
summary(customer_data$SpendingScore)
```

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

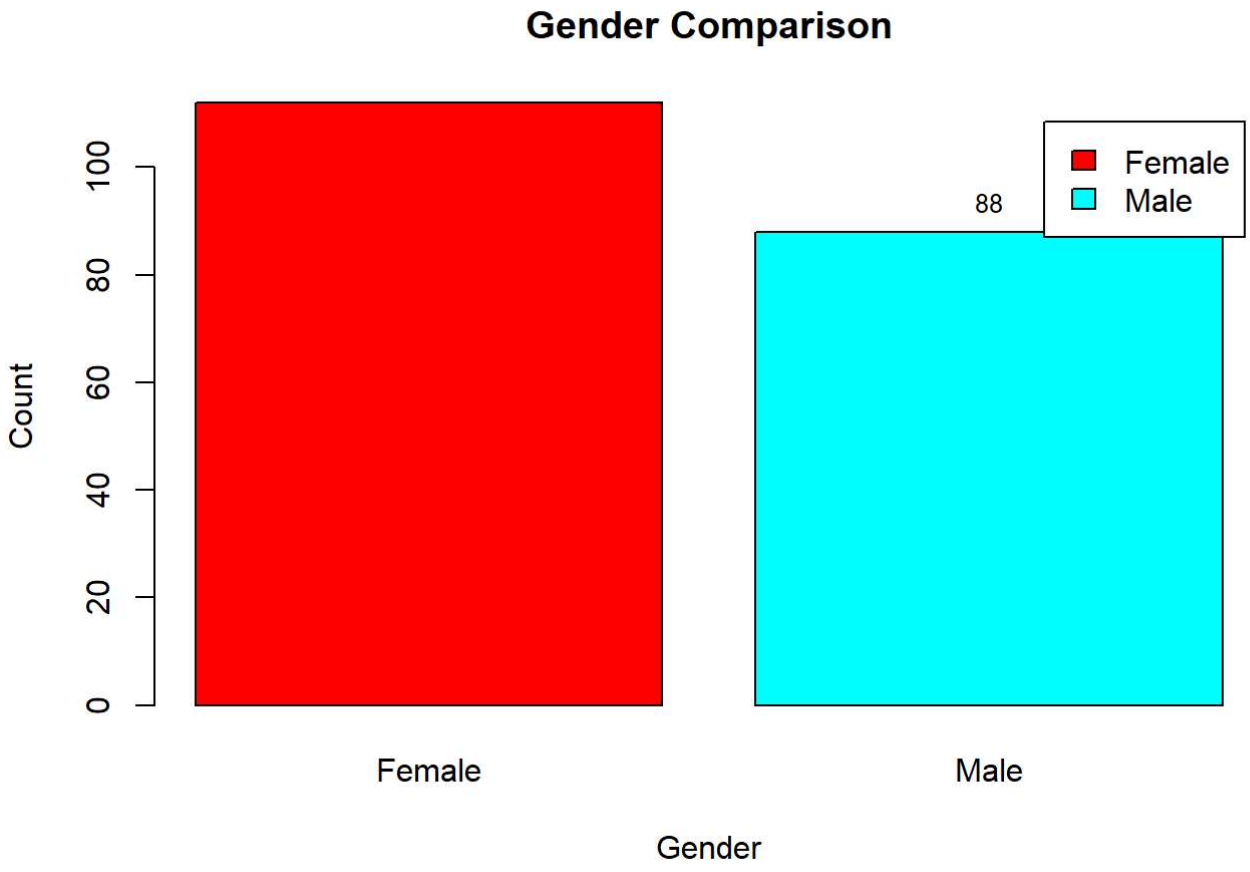
```
sd(customer_data$SpendingScore)
```

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

Visualizations

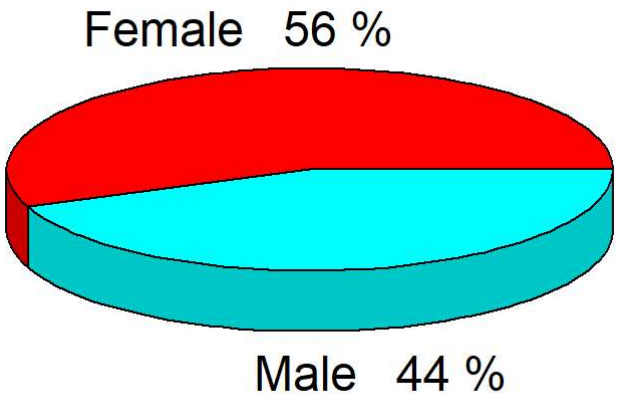
Bar Plot for Gender Distribution

```
# Frequency table for Gender
a=table(customer_data$Gender)

# Barplot for gender distribution
barplot_heights=barplot(a, main="Gender Comparison",
        ylab="Count", xlab="Gender",
        col=rainbow(2),
        legend=rownames(a))

text(barplot_heights, a, labels = a, pos = 3, cex = 0.8, col = "black")
```



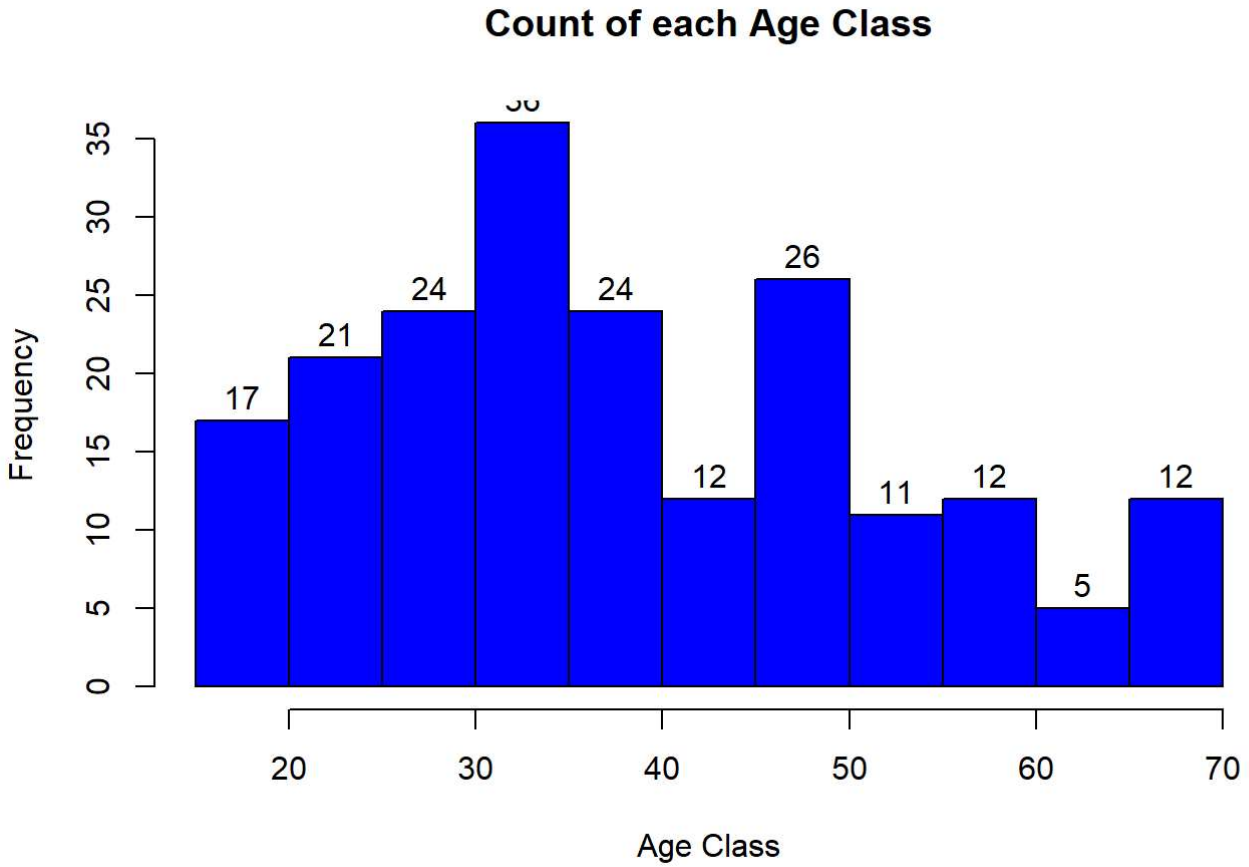
3D Pie Chart for Gender Ratio

```
library(plotrix)

pct=round(a / sum(a) * 100)

# Labels for the pie chart
lbs=paste(c("Female", "Male"), " ", pct, "%", sep=" ")
pie3D(a, labels=lbs, main="Pie Chart Ratio of Female Vs Male")
```

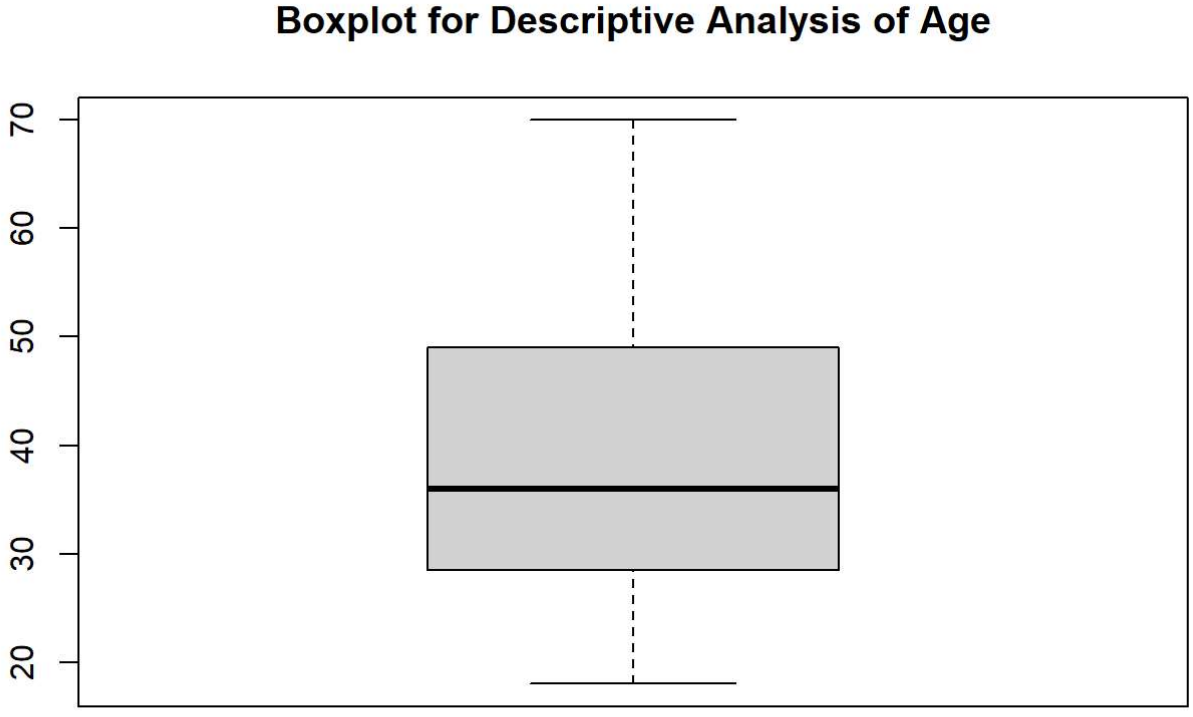


Histogram and Boxplot for Age

```
# Histogram for Age distribution
hist(customer_data$Age, col="blue",
     main="Count of each Age Class",
     xlab="Age Class", ylab="Frequency",
     labels=TRUE)
```

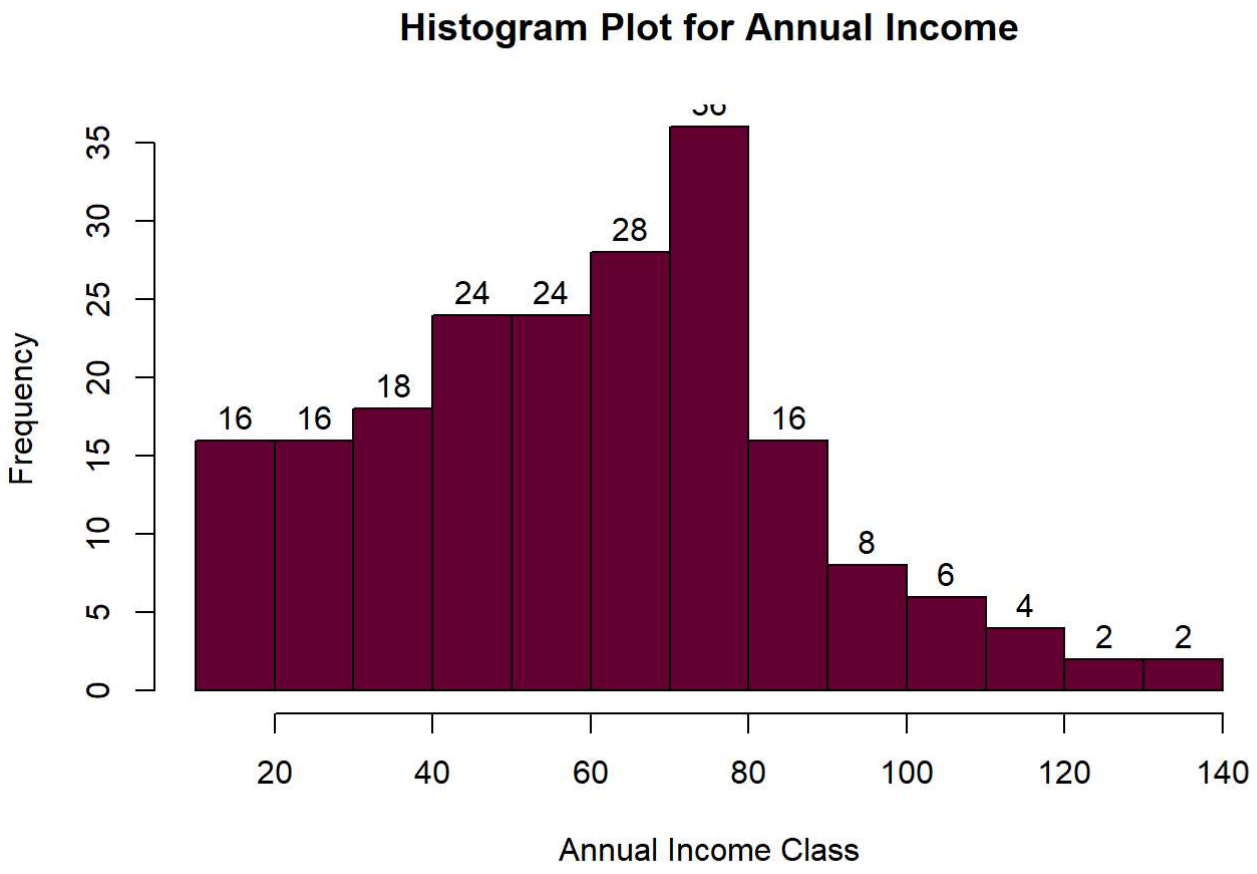



```
# Boxplot for Age
boxplot(customer_data$Age, main="Boxplot for Descriptive Analysis of Age")
```

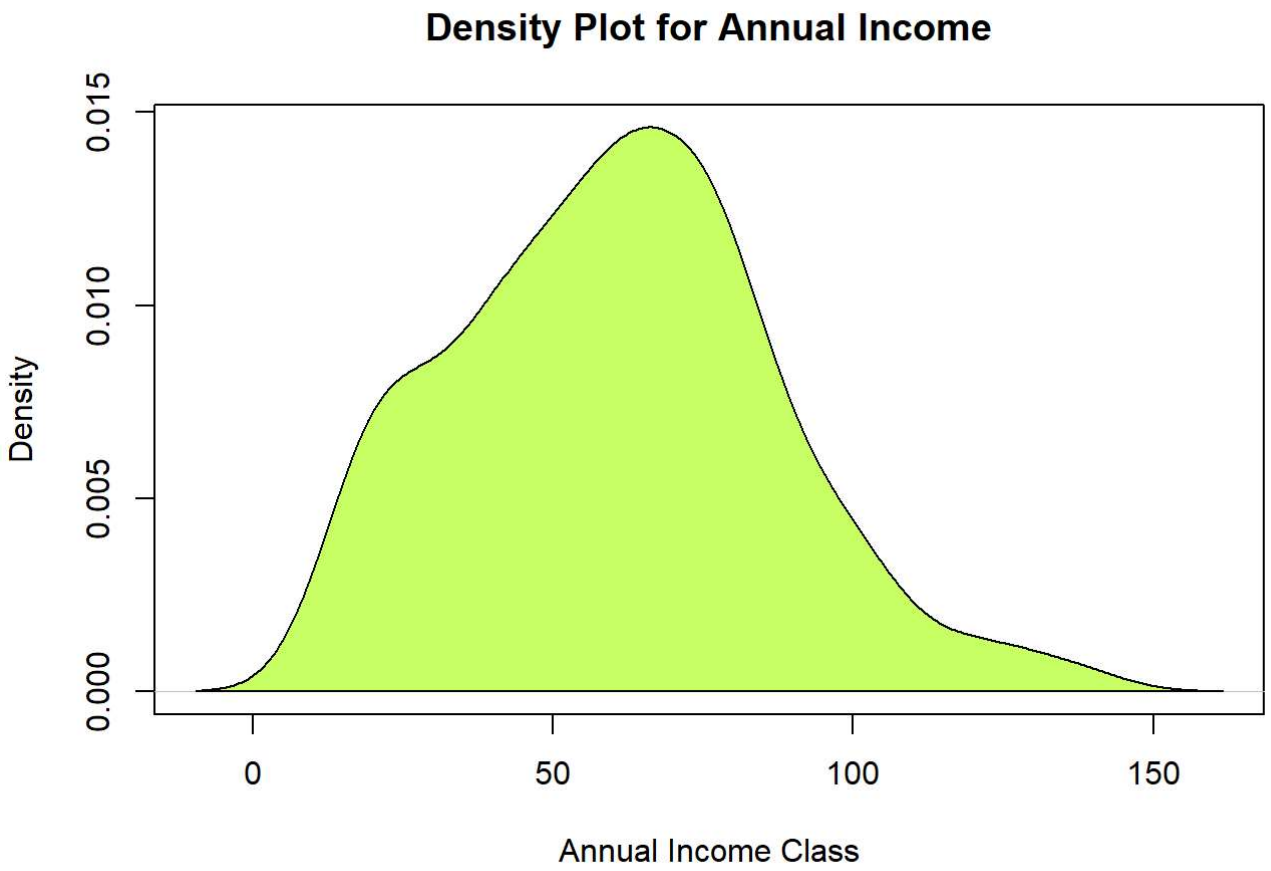


Annual Income Analysis

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

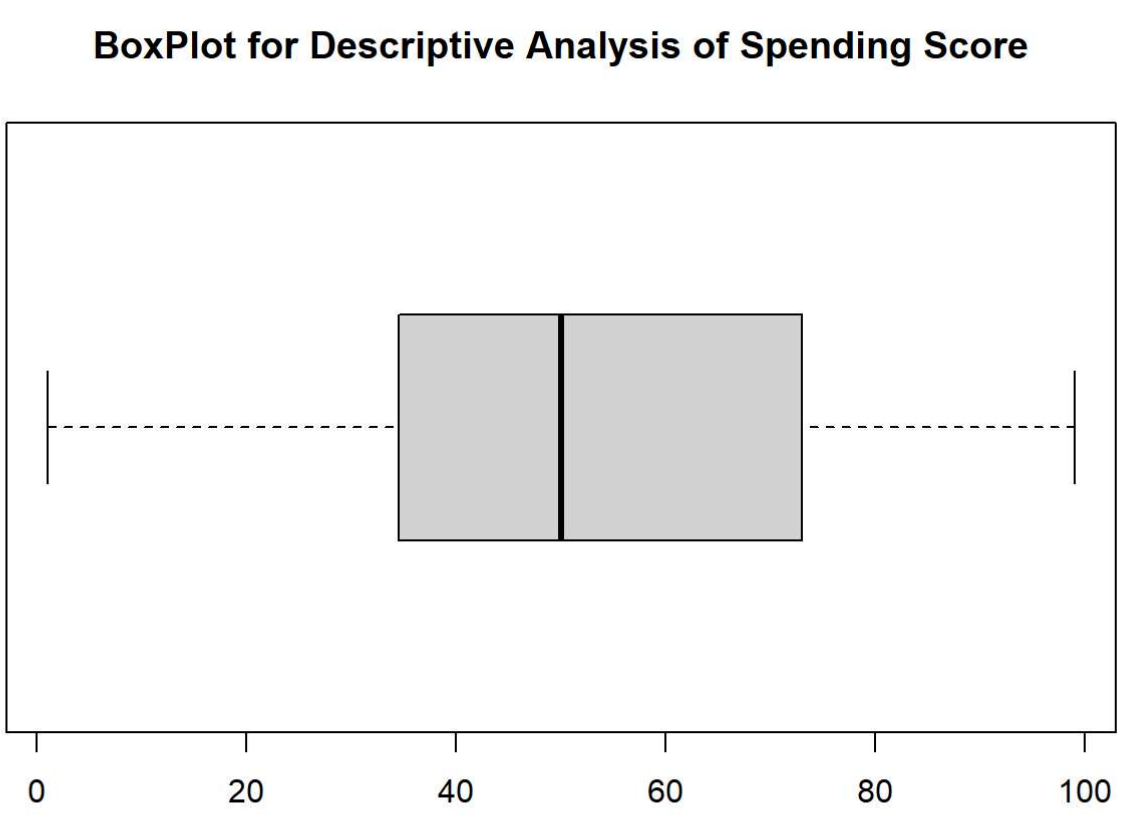



```
# Density plot for Annual Income
plot(density(customer_data$AnnualIncome), col="yellow",
     main="Density Plot for Annual Income",
     xlab="Annual Income Class", ylab="Density")

# Adding polygon to the density plot
polygon(density(customer_data$AnnualIncome), col="#ccff66")
```

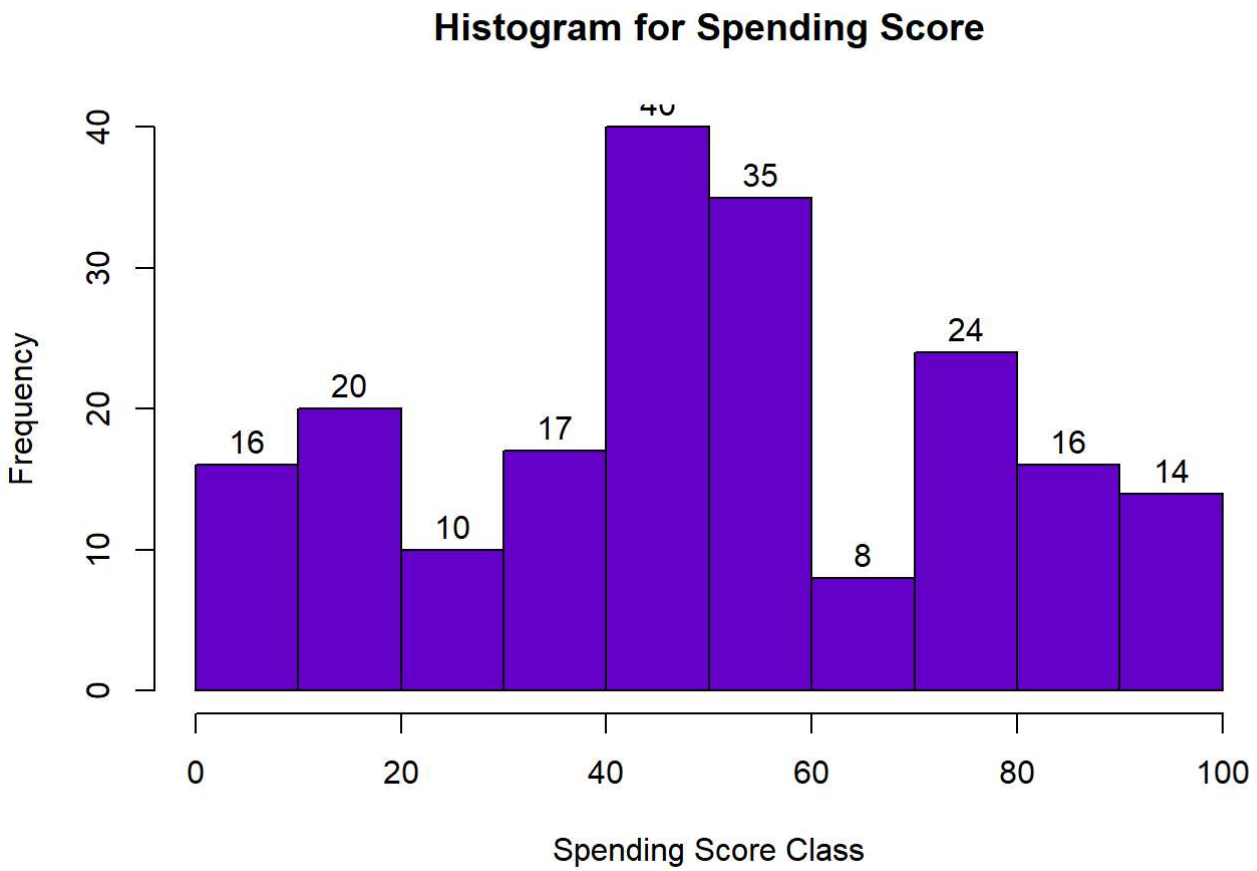


Spending Score Analysis

```
# Boxplot for Spending Score
boxplot(customer_data$SpendingScore, horizontal=TRUE,main="BoxPlot for Descriptive Analysis o
f Spending Score")
```

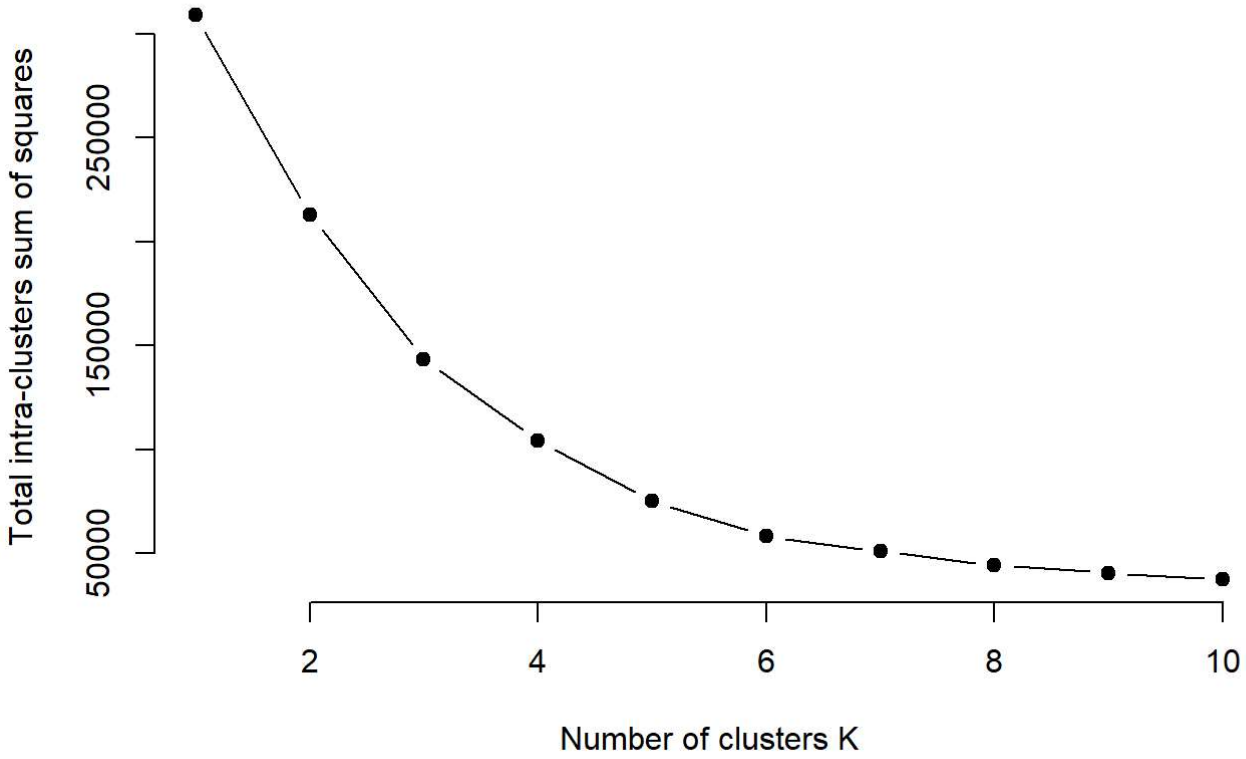



```
# Histogram for Spending Score
hist(customer_data$SpendingScore, main="Histogram for Spending Score",
     xlab="Spending Score Class", ylab="Frequency",
     col="#6600cc", labels=TRUE)
```



K-Means Clustering

```
# Load the purrr package
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
}

# Define the range of k values
k.values=1:10

# Calculate the total intra-cluster sum of square for each k
iss_values=map_dbl(k.values, iss)

# Plot the total intra-cluster sum of squares against k
plot(k.values, iss_values, type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K", ylab="Total intra-clusters sum of squares")
```



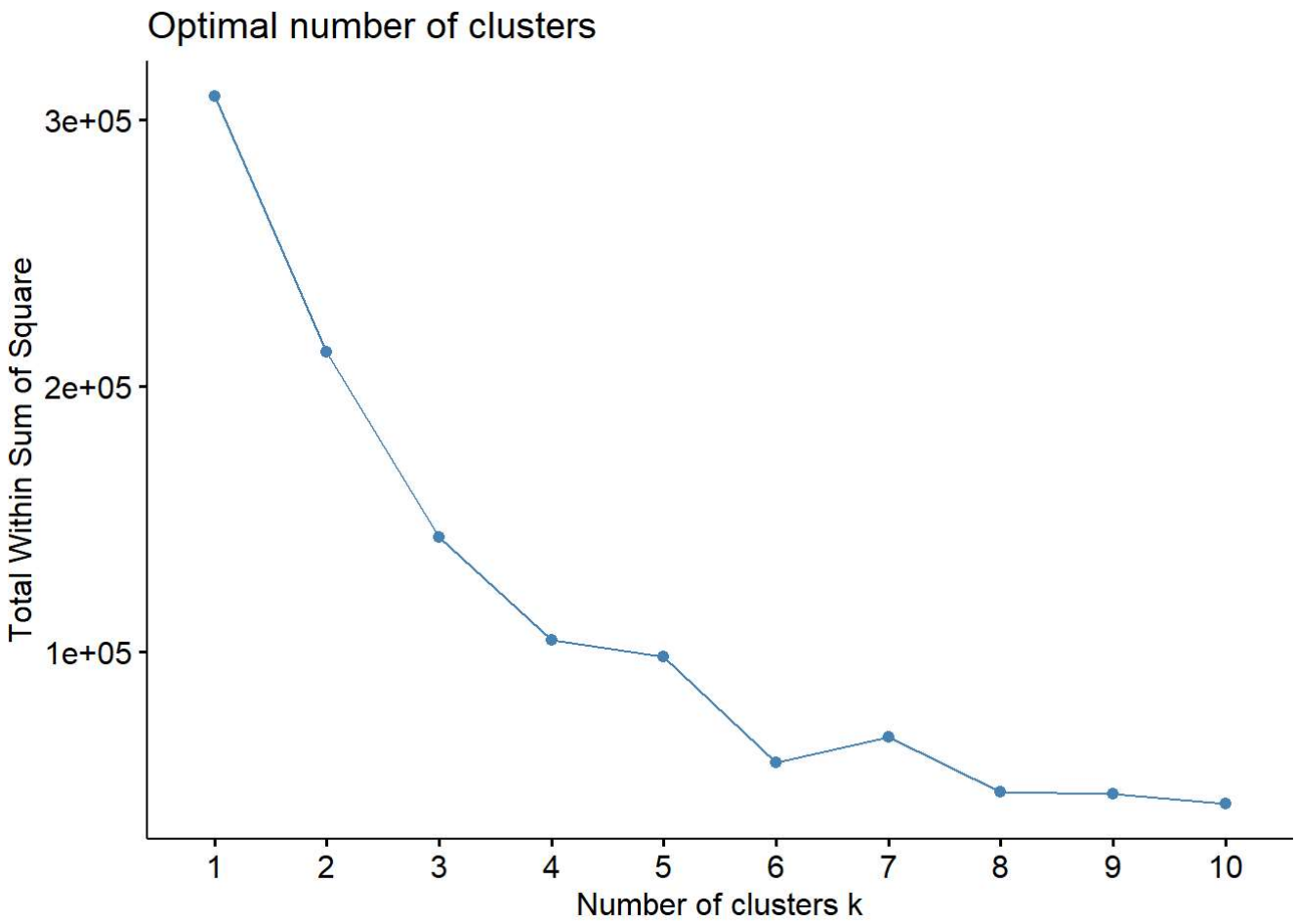

```
library(NbClust)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.1
```

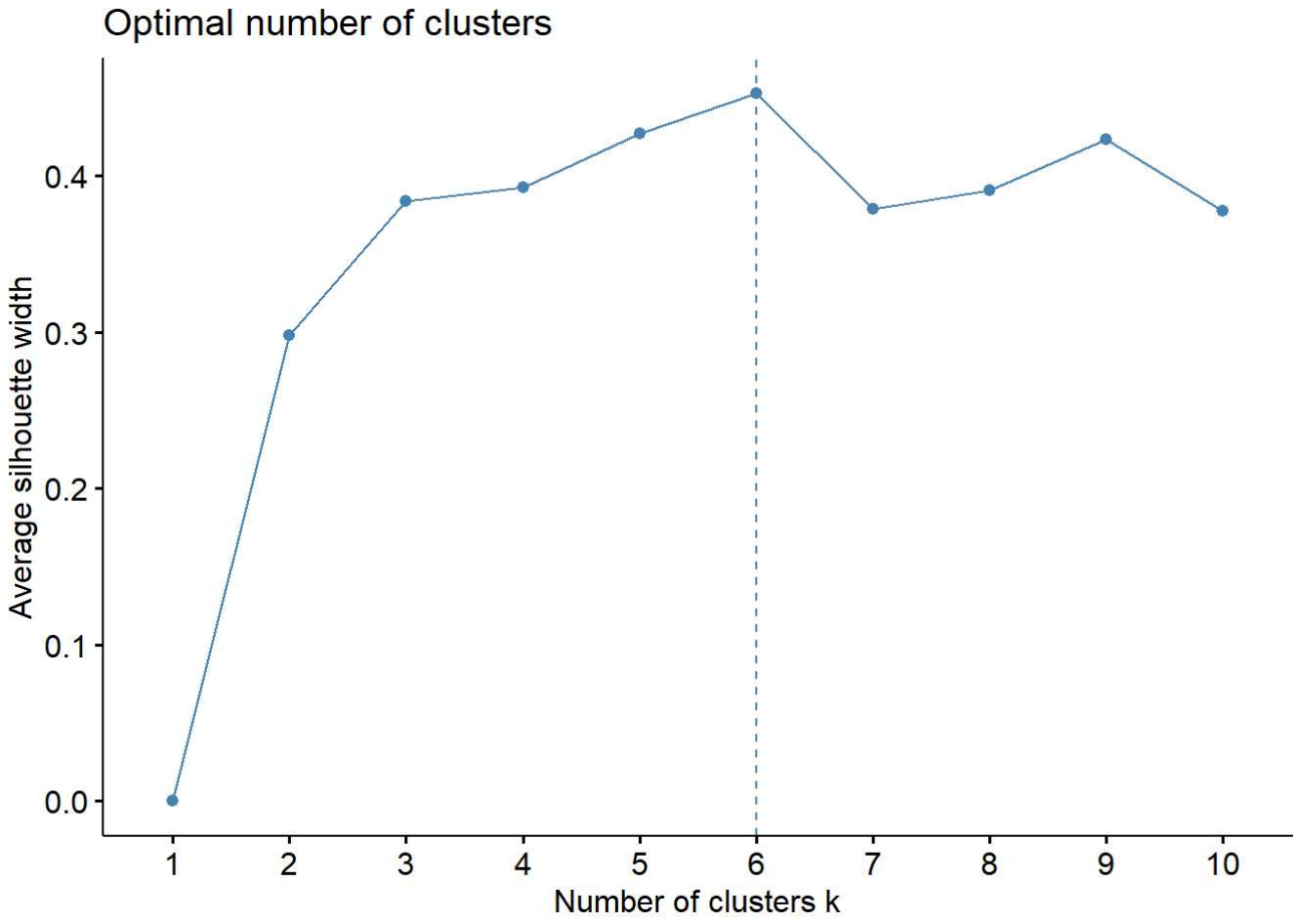
```
## Loading required package: ggplot2
```

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

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




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



Optimal Selection would be with 5 (wss) or 6 (silhouette).After looking with both we would be going with 5(wss).

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

```
## K-means clustering with 5 clusters of sizes 39, 37, 22, 79, 23
##
## Cluster means:
##        Age AnnualIncome SpendingScore
## 1 32.69231     86.53846      82.12821
## 2 40.32432     87.43243      18.18919
## 3 25.27273     25.72727      79.36364
## 4 43.12658     54.82278      49.83544
## 5 45.21739     26.30435      20.91304
##
## Clustering vector:
##   [1] 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5 3 5
##  [38] 3 5 3 5 3 5 4 5 3 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
##  [75] 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
## [112] 4 4 4 4 4 4 4 4 4 4 4 4 1 2 1 4 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
## [149] 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
## [186] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 13972.359 18448.865  4099.818 29909.114  8948.609
##  (between_SS / total_SS =  75.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

K-Means Clustering Visualization

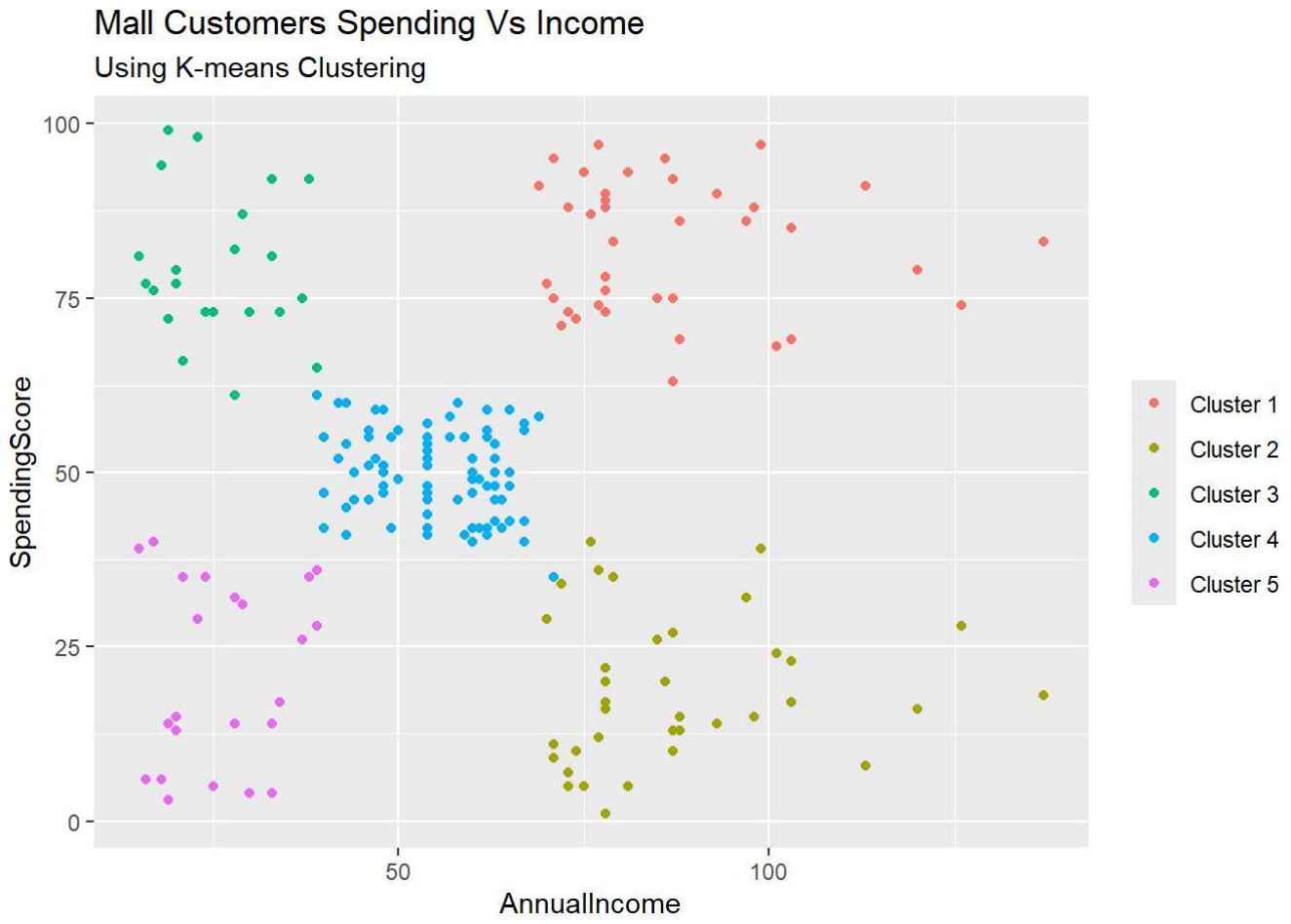
```
pcclust=prcomp(customer_data[,3:5],scale=FALSE) #principal component analysis
summary(pcclust)
```

```
## 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]
```

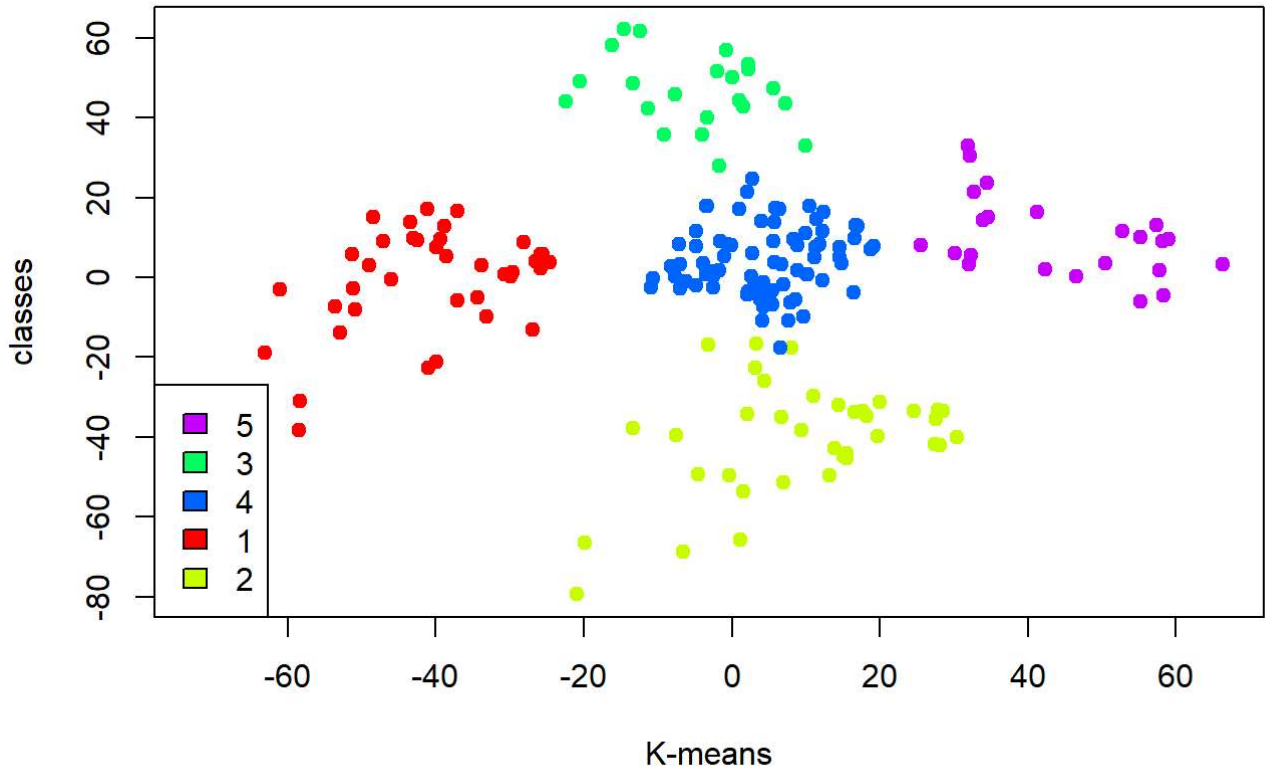
```
##                      PC1        PC2
## Age            0.1889742 -0.1309652
## AnnualIncome  -0.5886410 -0.8083757
## SpendingScore -0.7859965  0.5739136
```

```
set.seed(1)
ggplot(customer_data, aes(x =AnnualIncome, y = SpendingScore)) +
  geom_point(stat = "identity", aes(color = as.factor(k6$cluster))) +
  scale_color_discrete(name=" ",
                       breaks=c("1", "2", "3", "4", "5"),
                       labels=c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5")) +
  ggtitle("Mall Customers Spending Vs Income", subtitle = "Using K-means Clustering")
```



Final 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)))
```


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