Customer Segmentation Project

Morgan Rhee

```
library(plotrix, warn.conflicts = FALSE)
library(dplyr, warn.conflicts = FALSE)
library(cluster, warn.conflicts = FALSE)
```

```
## Warning: package 'cluster' was built under R version 4.1.2
```

```
library(ggplot2, warn.conflicts = FALSE)
library(gridExtra, warn.conflicts = FALSE)
library(grid, warn.conflicts = FALSE)
library(factoextra, warn.conflicts = FALSE)
```

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

```
library(grid, warn.conflicts = FALSE)
library(NbClust, warn.conflicts = FALSE)
```

Purpose: For any company, for any employer, **customer segmentation** is crucial in identifying the best customers. This can allow potential targeting of best user databases, as well as utilize clustering techniques to delineate the best marketing strategies. In this project, I conduct a deep dive analysis into a messy data set in order to **segment the profitable categorizations** needed to strategize differentiators that can direct the company best. With such, companies are capable of gaining better insight into their customers' preferences, suitable marketing techniques, and minimized investment risks.

1. Data Implementation and Basic Analysis

```
dataset <- read.csv("/Users/yonjerhee/Desktop/project/Mall_Customers.csv")
head(dataset)
```

```
##   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(dataset)
```

```
## '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 ...
```

```
names(dataset)
```

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

```
summary(dataset)
```

```
##    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
```

```
sd(dataset$Age)
```

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

```
sd(dataset$Annual.Income..k..)
```

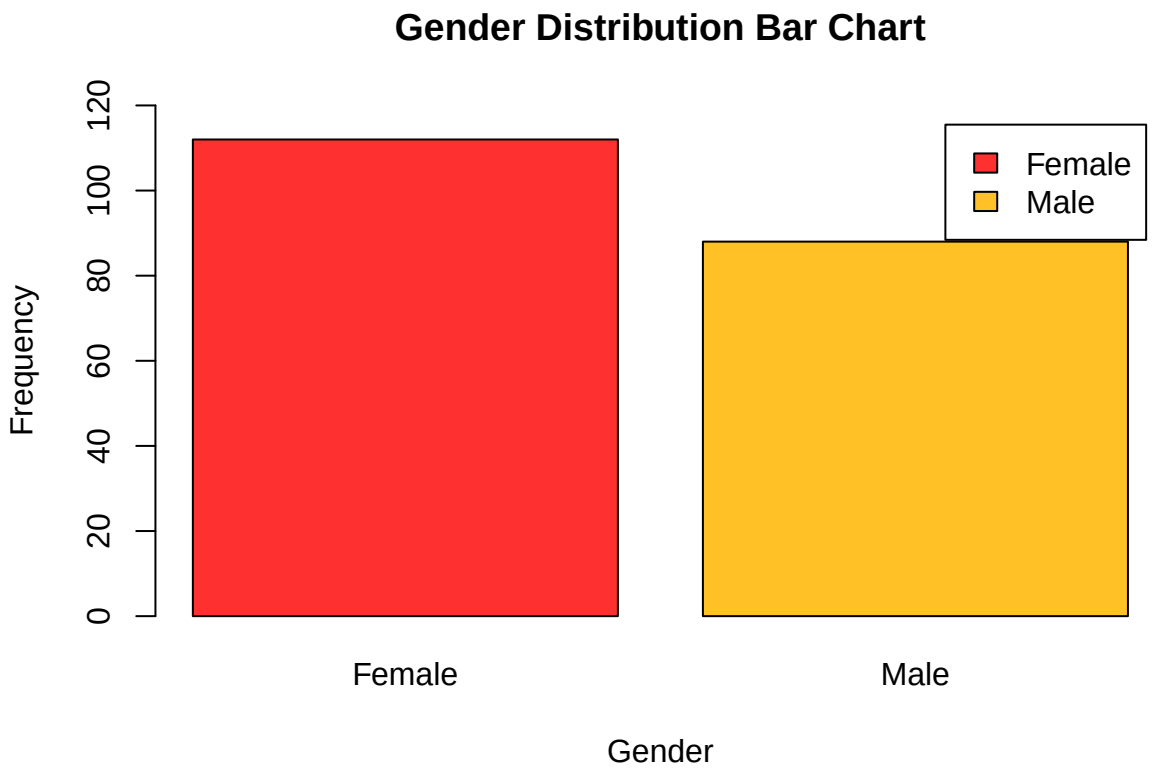
```
## [1] 26.26472
```

2. General Distribution Analysis

2A. Gender Distribution Visualization

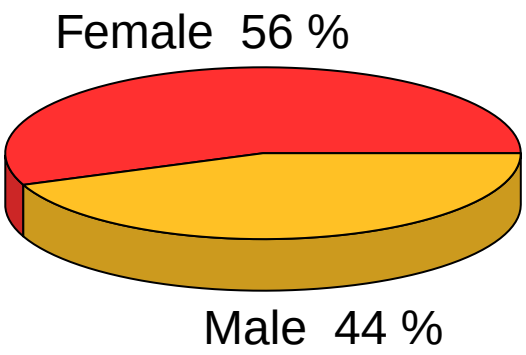
When given a data set of customers, the first set of visualization that could potentially be done is that of the gender, more specifically determining whether gender distribution is similar and if there are any discrepancies.

```
#bar chart for distribution of gender
barplot(table(dataset$Gender), main = "Gender Distribution Bar Chart",
        xlab = "Gender",
        ylab = "Frequency",
        ylim = c(0, 120),
        legend = rownames(table(dataset$Gender)),
        col = c("firebrick1", "goldenrod1"))
```



We can notice right away that in this data set, there are more females (approximately 20 more females) than men.

```
#pie chart for distribution of gender
gender_dat <- table(dataset$Gender)
percent <- round(gender_dat/sum(gender_dat)*100)
pie3D(gender_dat,
      main = "Gender Distribution Ratio Pie Chart",
      labels = paste(c("Female", "Male"), "", percent, "%", sep= " "),
      col = c("firebrick1", "goldenrod1"))
```




```
#number of females in population
nrow(filter(dataset, Gender == "Female"))
```

```
## [1] 112
```

```
#number of males in population
nrow(filter(dataset, Gender == "Male"))
```

```
## [1] 88
```

In terms of ratio distribution, we can see that the female population is more than half of the total population with 56%. Knowing that there are 200 individuals in the data, we can determine that there are thus 112 females and 88 males.

2B. Age Distribution Visualization

Following gender, we can further distribute the data set into age categories to not only get a sense of the age range but also generalize tailored distributions to respective age categories.

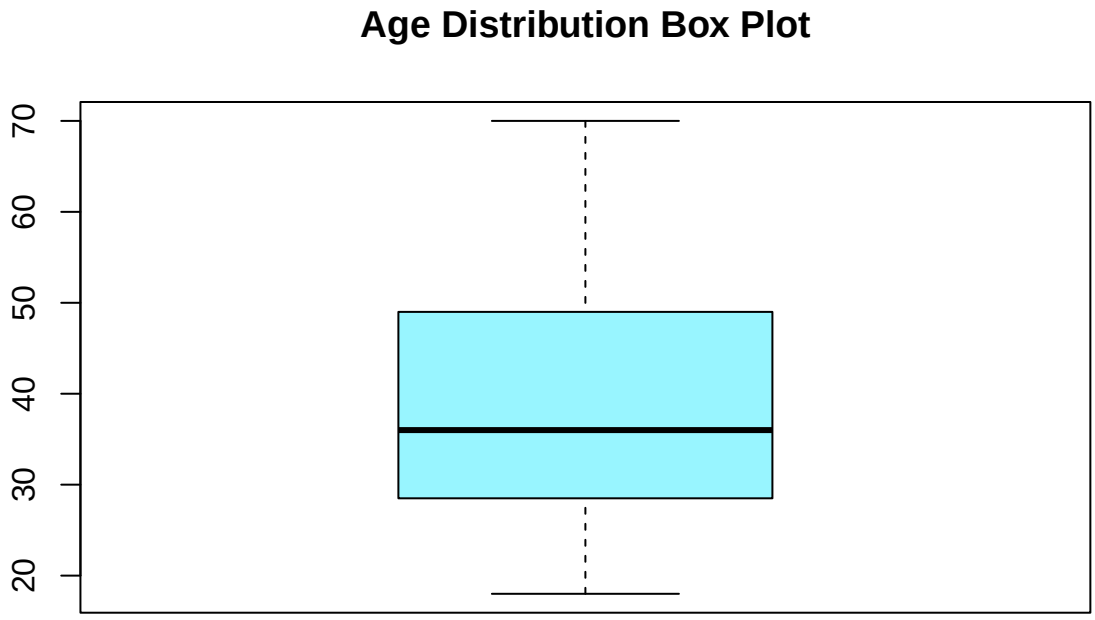
```
#summary of age distribution to obtain general sense
summary(dataset$Age)
```

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

There is quite an age range across the data, with the youngest being 18 years old and the oldest being 70 years old.

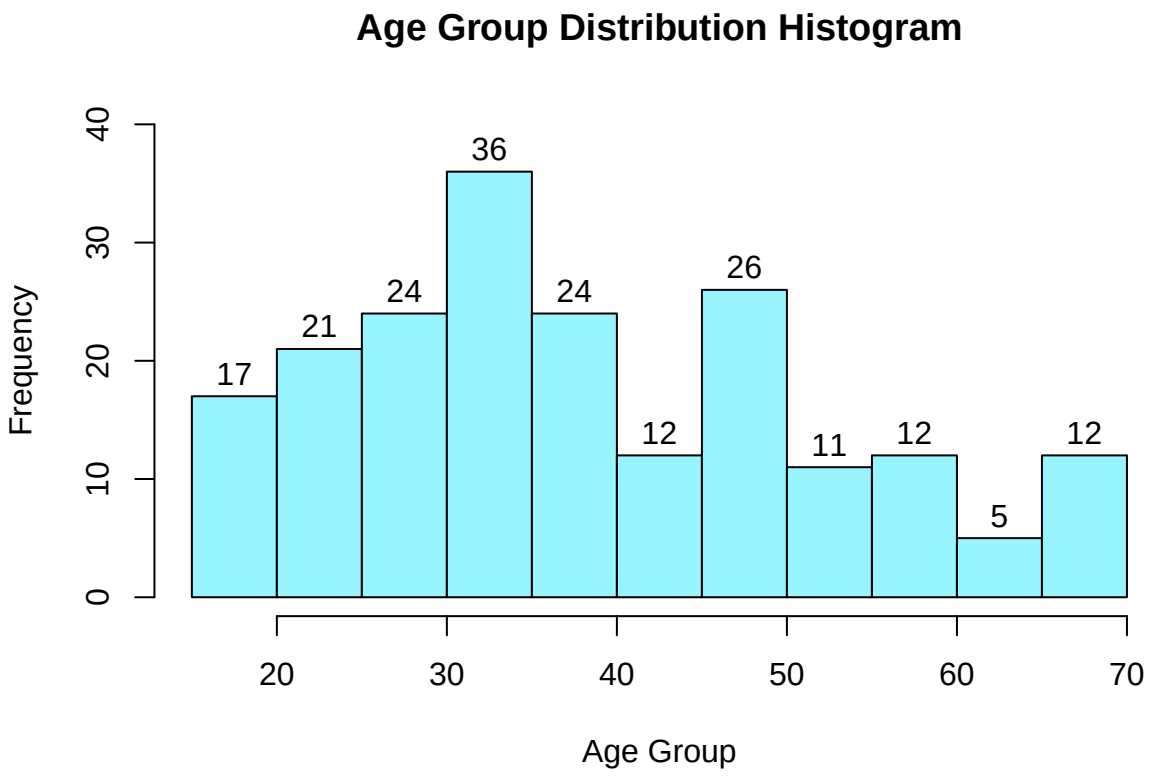
The summary of the data can best be visualized with a box plot which easily shows the the five features of the minimum, the maximum, the mean, as well as the 25th and 75th percentile

```
boxplot(dataset$Age,
        main = "Age Distribution Box Plot",
        col = "cadetblue1")
```



For a general scheme of visualization, the histogram would be the best way to classify and identify the varying age groups.

```
hist(dataset$Age,
     main = "Age Group Distribution Histogram",
     xlab = "Age Group",
     ylab = "Frequency",
     ylim = c(0, 40),
     labels = TRUE,
     col = "cadetblue1"
     )
```

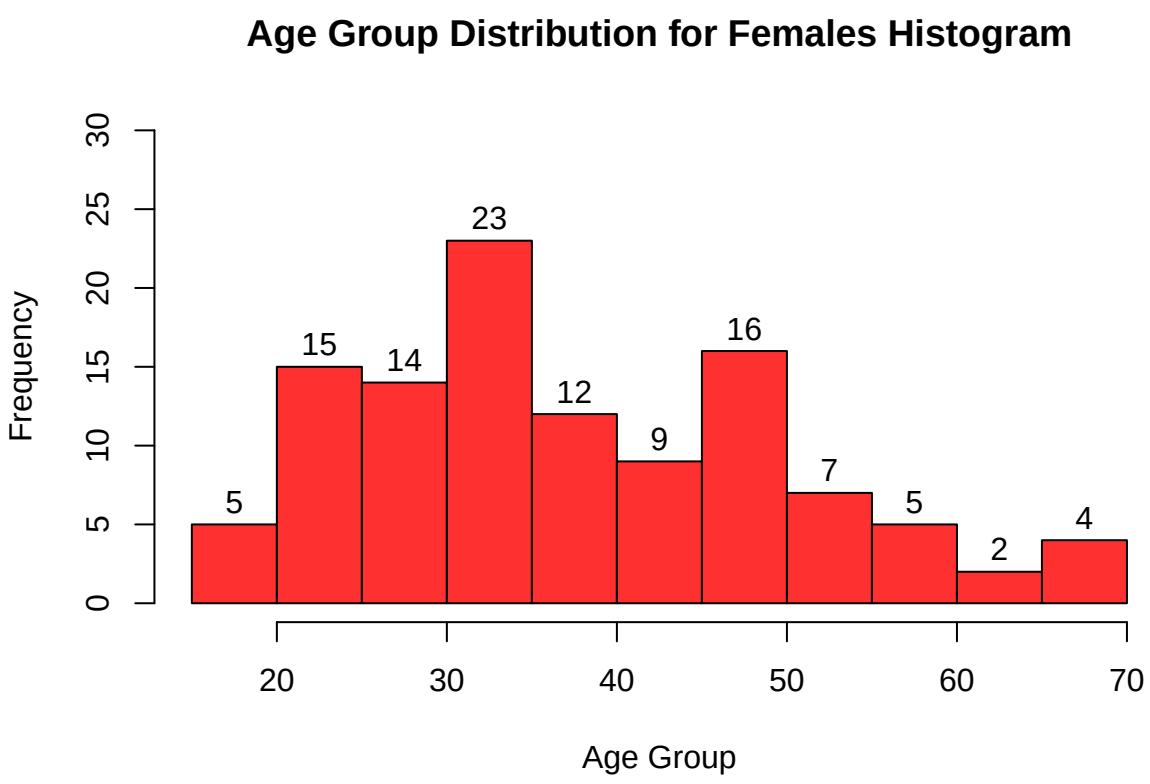


Primarily, there are many individuals in their 20s-40s, with the most number of individuals in the 30s-35s category, more specifically 36 individuals.

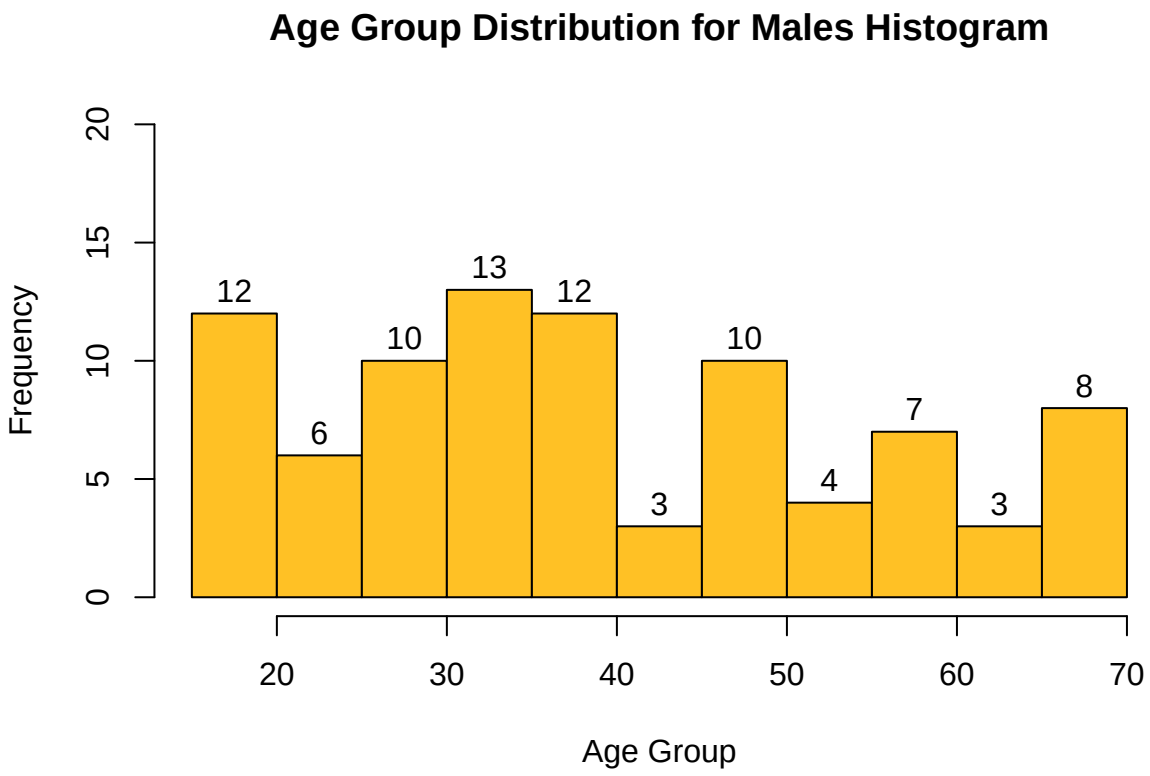
2C. Gender and Age Distribution Visualization

Furthermore, now that we have information on gender and age respectively, it would be interesting to note if the most common age groups were to be different across gender for future generalization.

```
#age grouping for females
females <- filter(dataset, Gender == "Female")
hist(females$Age,
     main = "Age Group Distribution for Females Histogram",
     xlab = "Age Group",
     ylab = "Frequency",
     ylim = c(0, 30),
     labels = TRUE,
     col = "firebrick1"
     )
```




```r
#age grouping for males
males <- filter(dataset, Gender == "Male")
hist(males$Age,
     main = "Age Group Distribution for Males Histogram",
     xlab = "Age Group",
     ylab = "Frequency",
     ylim = c(0, 20),
     labels = TRUE,
     col = "goldenrod1"
     )
```



In females, we can see that there the two most common age groups are 30s-35s and 45s-50s, whereas in males, the common age groups are in the 40s and under. This difference can emerge from the concept that there are noticeably less male than female; however, it is important to know the discrepancy in age groups for each of the two genders.

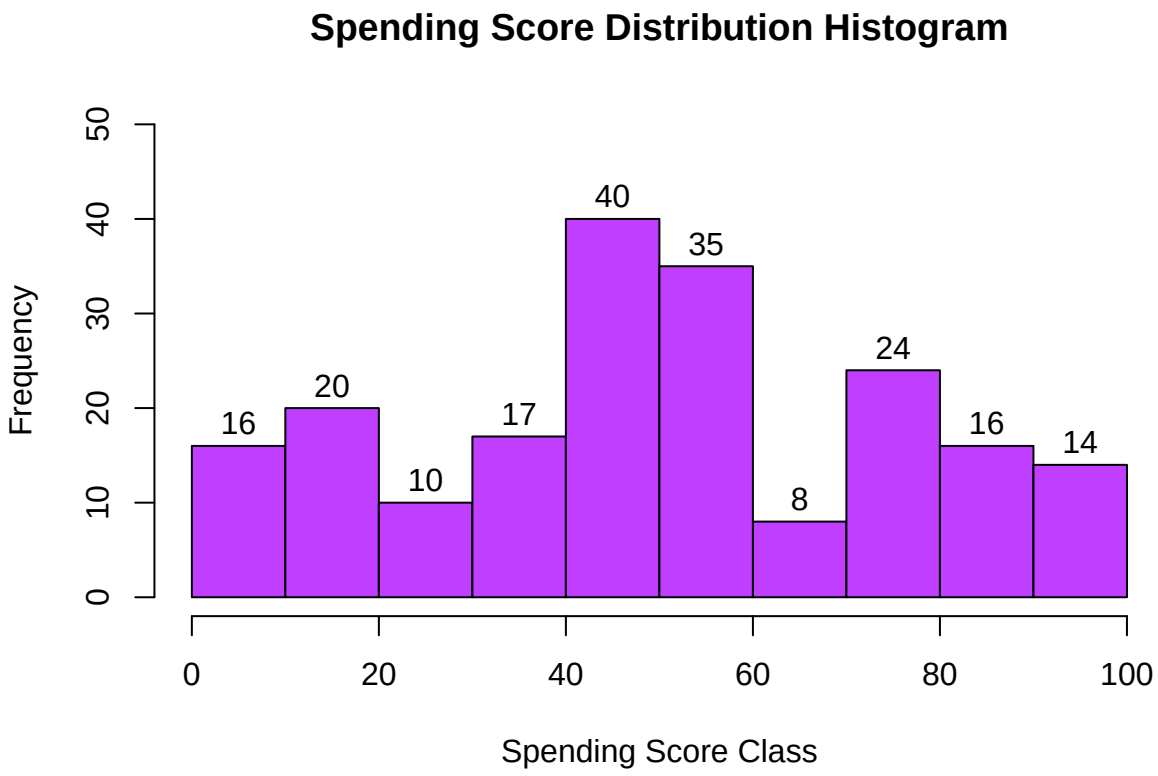
2D. Customer Spending Score Analysis

To dive into the customer tailored analysis, looking first at the spending scores of the customers through visual representations such as a histogram and a box plot is necessary.

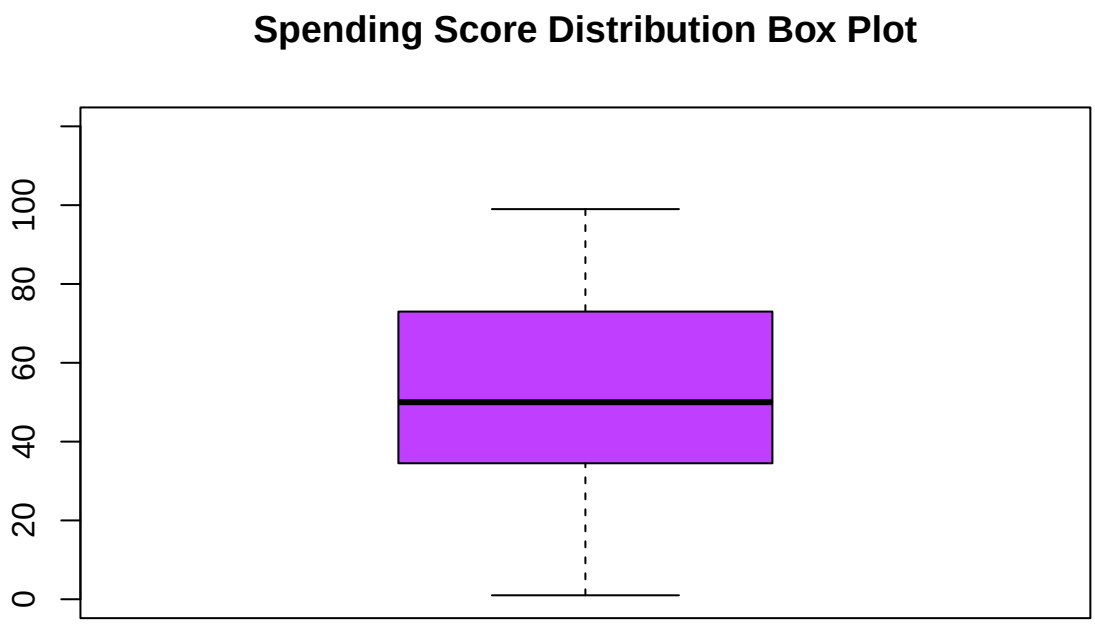
```
#summary of spending scores
summary(dataset$Spending.Score..1.100.)
```

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

```
hist(dataset$Spending.Score..1.100.,
     main = "Spending Score Distribution Histogram",
     xlab = "Spending Score Class",
     ylab = "Frequency",
     ylim = c(0, 50),
     labels = TRUE,
     col = "darkorchid1")
```




```
boxplot(dataset$Spending.Score..1.100.,
        main = "Spending Score Distribution Box Plot",
        col = "darkorchid1",
        ylim = c(0, 120))
```



We can conclude that for the spending scores, there was a minimum score of 1, a maximum score of 99, and the mean being 50.20. Furthermore, we can conclude from the histogram that the spending score class from 40 to 60 had the highest frequencies.

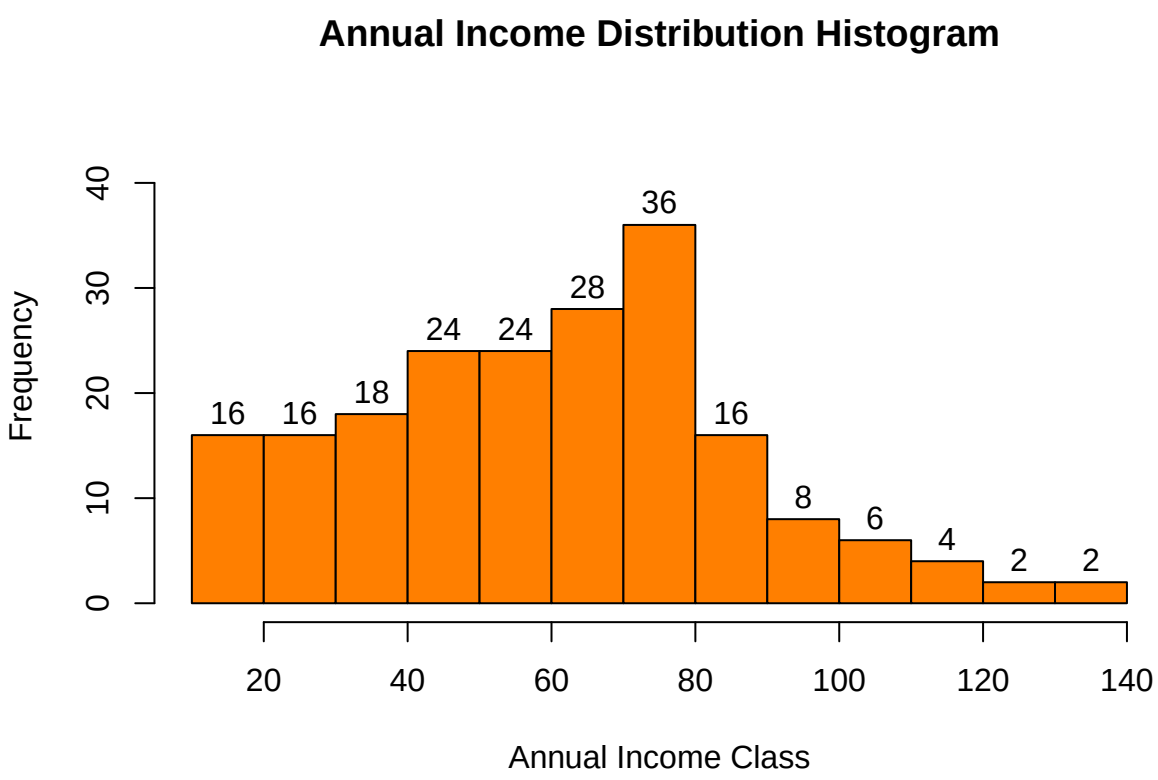
2E. Customer Annual Income Analysis

To dive into the customer tailored analysis, looking second at the annual income of the customers through visual representations such as a histogram and a density plot is necessary.

```
#summary of annual income
summary(dataset$Annual.Income..k..)
```

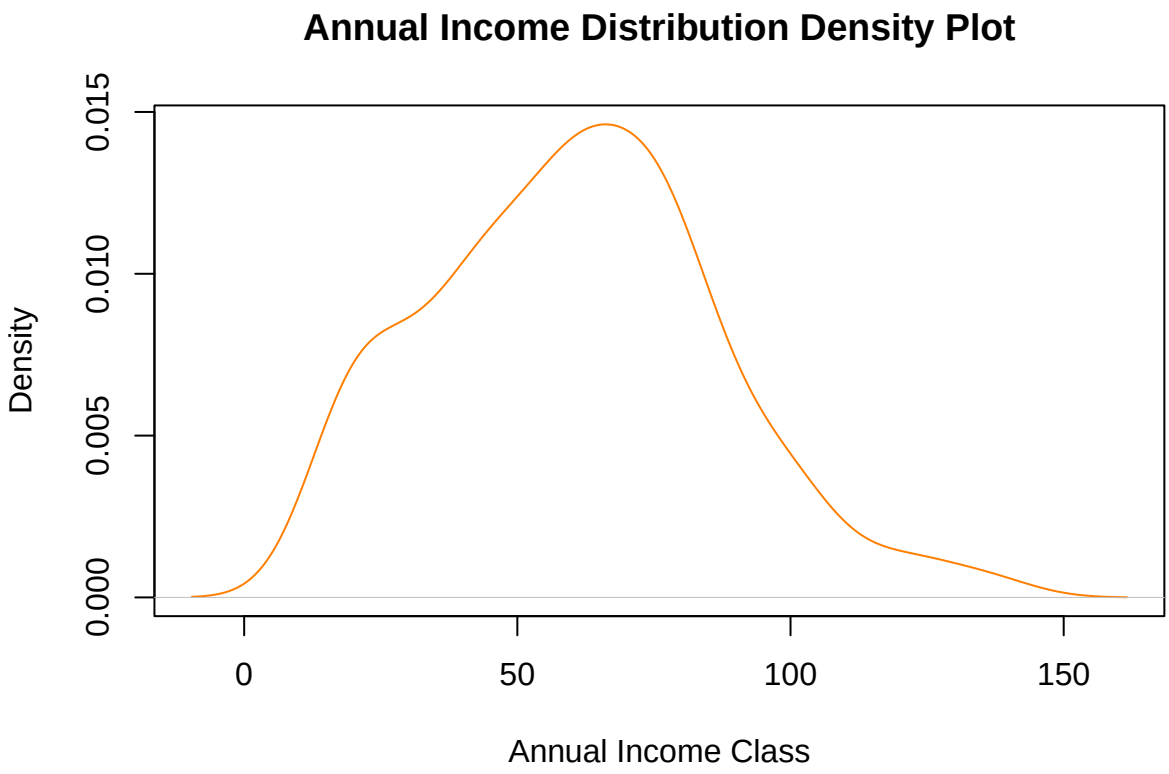
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   41.50   61.50   60.56   78.00  137.00
```

```
hist(dataset$Annual.Income..k..,
     main = "Annual Income Distribution Histogram",
     xlab = "Annual Income Class",
     ylab = "Frequency",
     ylim = c(0, 45),
     labels = TRUE,
     col = "darkorange1")
```



From the summary and the histogram, we see that the annual income summaries are as follows: a minimum value of 15, a maximum value of 137, and a mean of 60.56. Furthermore, the income class of 70-80 inclusive had the highest frequency of 36 individuals.

```
plot(density(dataset$Annual.Income..k..),
     main = "Annual Income Distribution Density Plot",
     xlab = "Annual Income Class",
     ylab = "Density",
     col = "darkorange1")
```



The density plot above demonstrates that the data set follows a close normal distribution, with the plot being centered closely around the mean. While there exists a slight right skew, seeing as how the curve is almost symmetric, we can conclude that the annual income is a normal distribution.

3. K-Means Algorithm

With this K-Means Clustering Algorithm, there are a few steps and methods in order to **iterate and determine the optimal clusters**. We must indicate the number of k cluster we want in order randomize the centroids (initial cluster centers). Utilizing the Euclidean Distance, we can calculate the mean value of its distance between the object and cluster mean through maximum iteration. The program will stop running once maximum iteration is achieved through the total sum of squares.

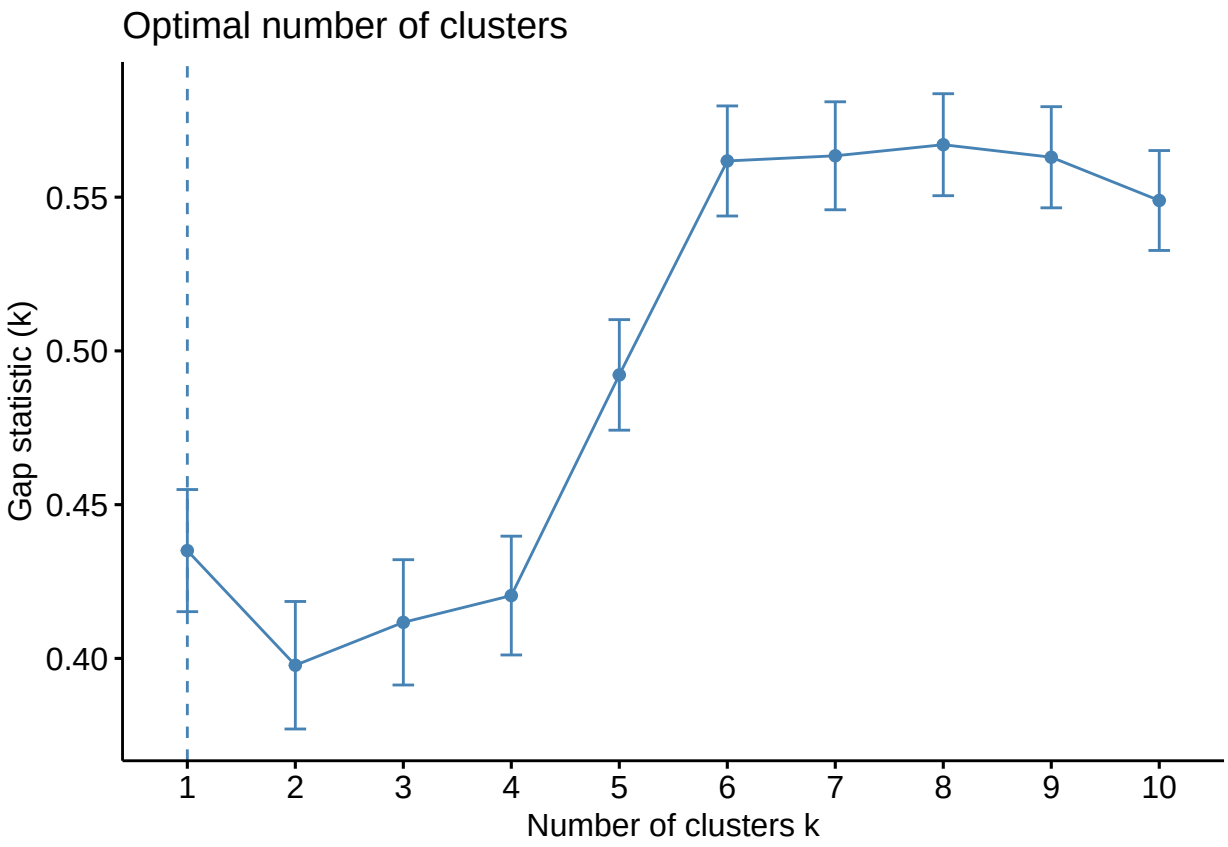
In order to determine the optimal number of clusters, this project makes use of the **gap statistic method** and the **silhouette method**.

3A. Gap Statistic Method

This method, referenced by the famous Monte Carlo simulations, we can calculate the minimum-maximum range, creating units for both the lower and upper interval bound. For each different value of k, comparison of their expected value (under the null distribution) with variation can be determined.

In the "clusGap" function, incorporation of K = 20 (a maximum of 20 clusters to be considered) and B = 100 (100 bootstrap samples run) will be coded.

```
set.seed(2022)
gap_statistic <- clusGap(dataset[, 3:5],
                         FUN = kmeans,
                         nstart = 25,
                         K.max = 10,
                         B = 100)
fviz_gap_stat(gap_statistic)
```



Taking k = 6 as being our optimal cluster, we can determine the output of K-means operations.

```
k_6 <- kmeans(dataset[, 3:5], 6,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
k_6
```

```
## 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 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"
```

```
#cluster number for each observation
k_6$cluster
```

```
##   [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 4 3
## [186] 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
```

```
#cluster means
k_6$centers
```

```
##        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
```

```
#cluster size
k_6$size
```

```
## [1] 45 21 35 39 38 22
```

From the data above, we can conclude components such as but not limited to the vector of the 6 clusters that have a significant allocation at each point, cluster means, and cluster size.

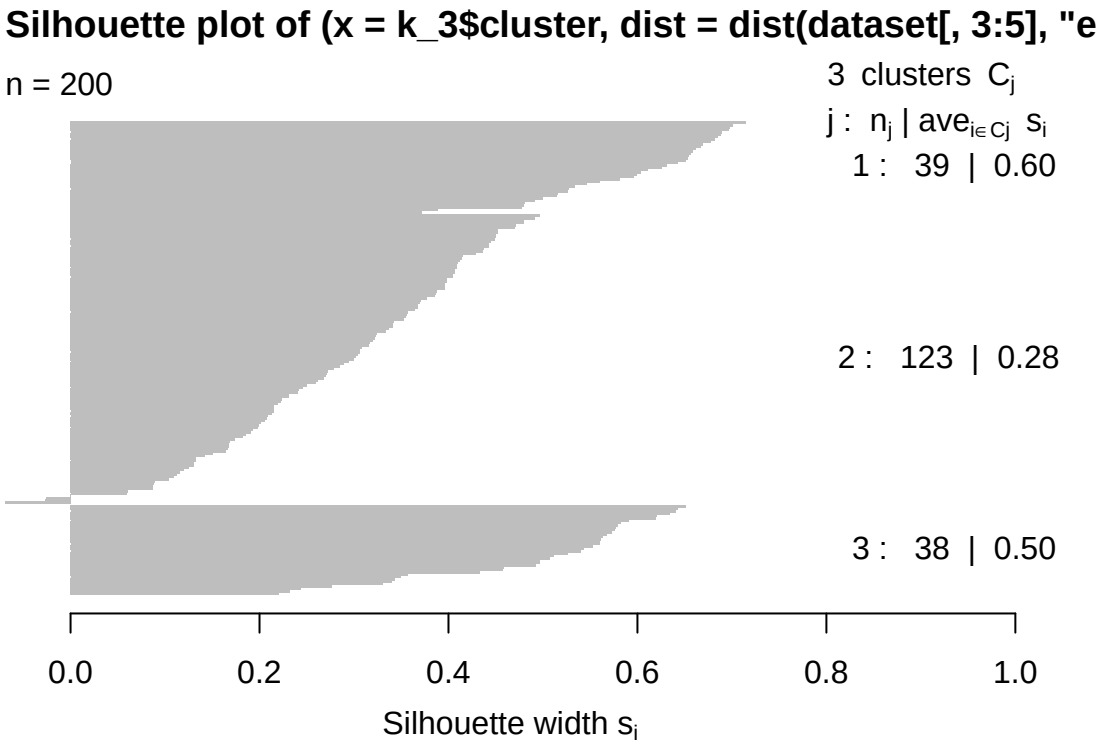
3B. Silhouette Method

This method allows us to measure our **optimized clustering quality** by figuring out how well our data associates with the cluster. Using the optimal number of k clusters, we can determine the maximum average

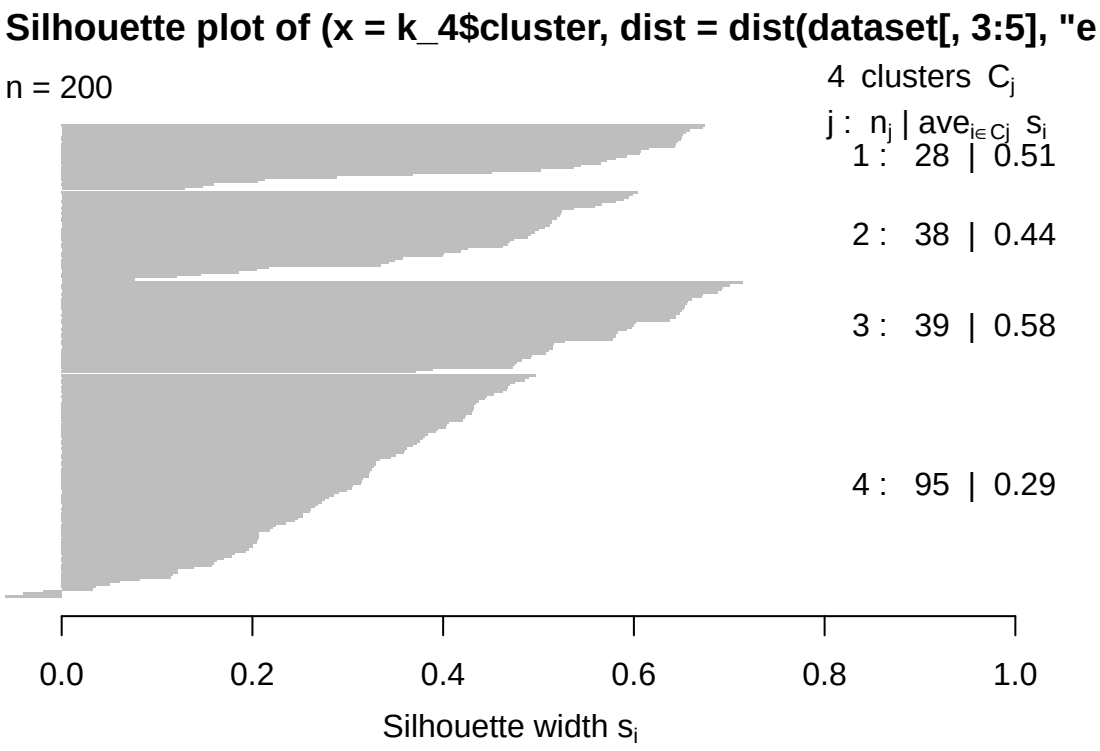
silhouette for the significant k clusters. The mean resulted from the method calculates the silhouette mean for the various k values we have predetermined. An indicator of a good clustering would result from a high average silhouette width.

To make sure that the optimal cluster is k = 6, we will work our way up from 3 to 9 to determine the average silhouette width for each.

```
#optimal cluster of k = 3
k_3 <- kmeans(dataset[, 3:5], 3,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette3 <- plot(silhouette(k_3$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```

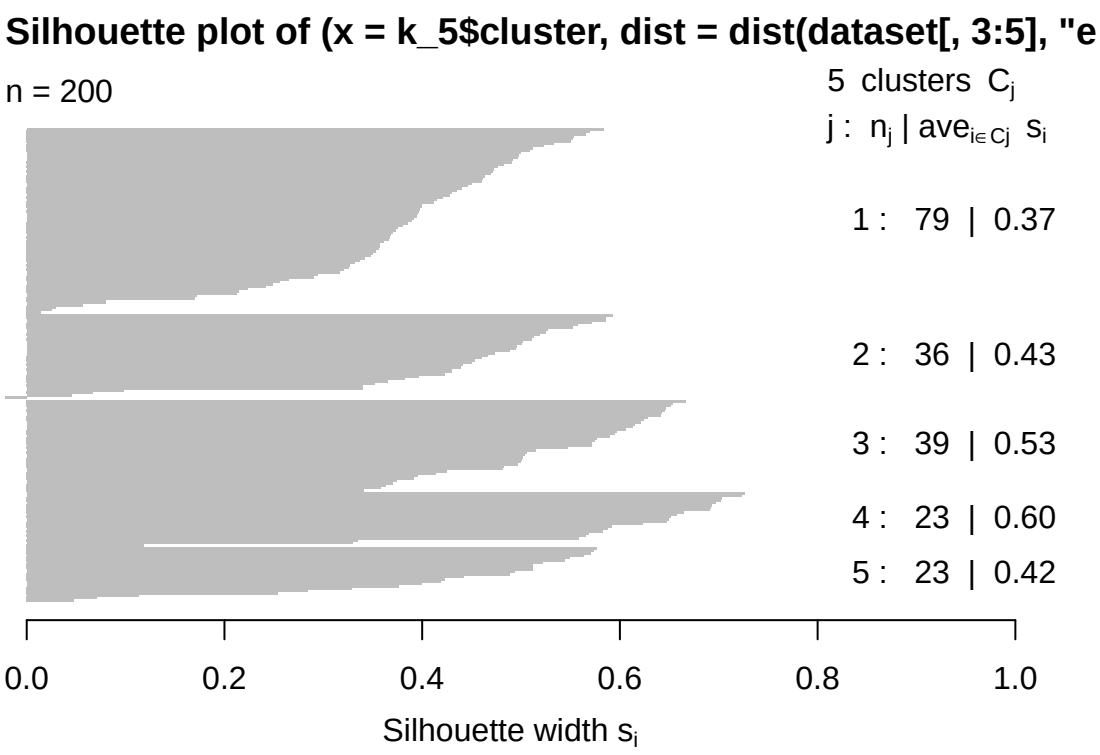



```
#optimal cluster of k = 4
k_4 <- kmeans(dataset[, 3:5], 4,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette4 <- plot(silhouette(k_4$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```

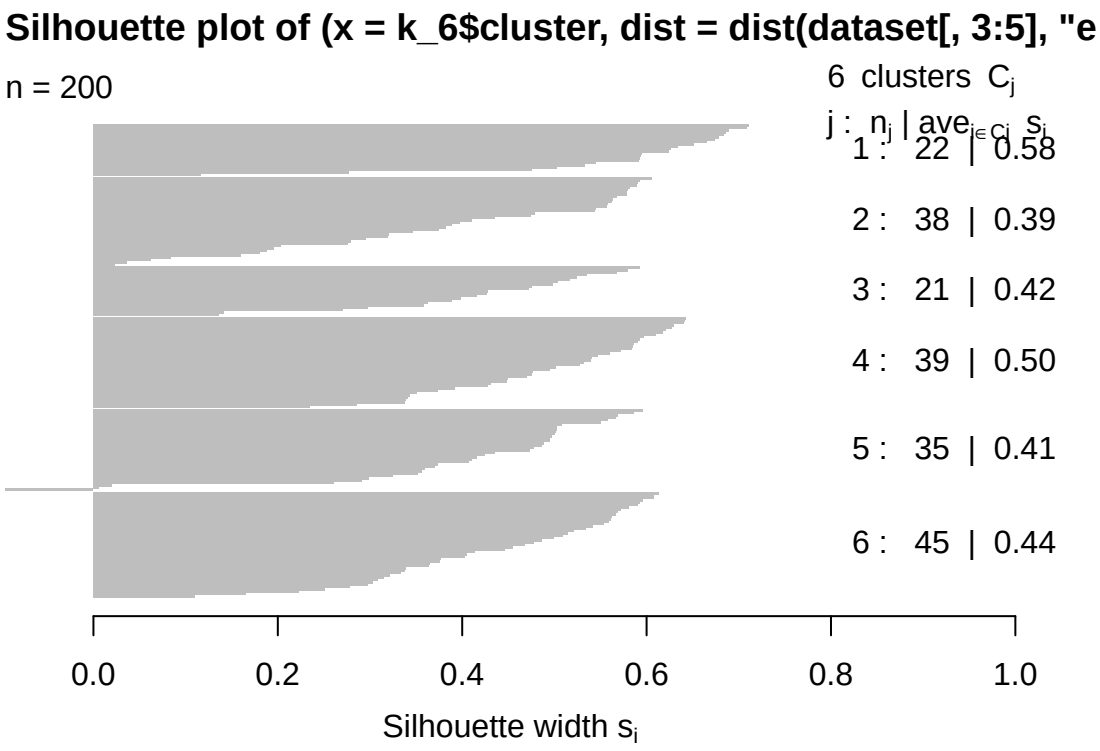


Average silhouette width : 0.41

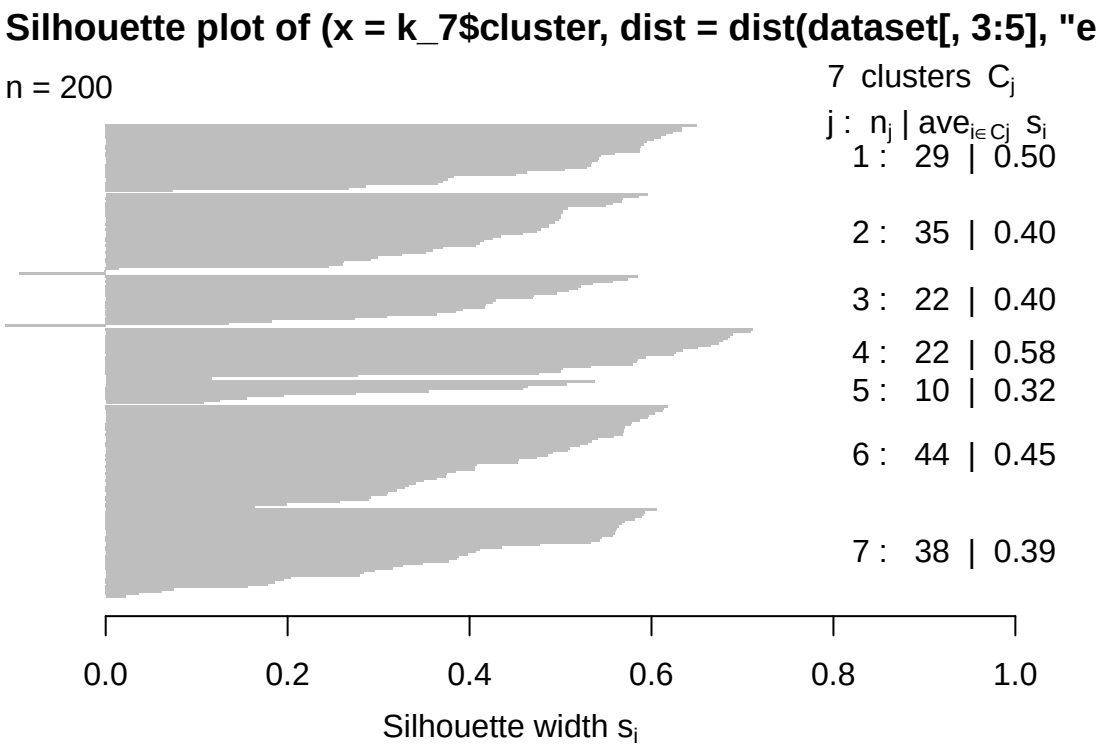
```
#optimal cluster of k = 5
k_5 <- kmeans(dataset[, 3:5], 5,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette5 <- plot(silhouette(k_5$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```



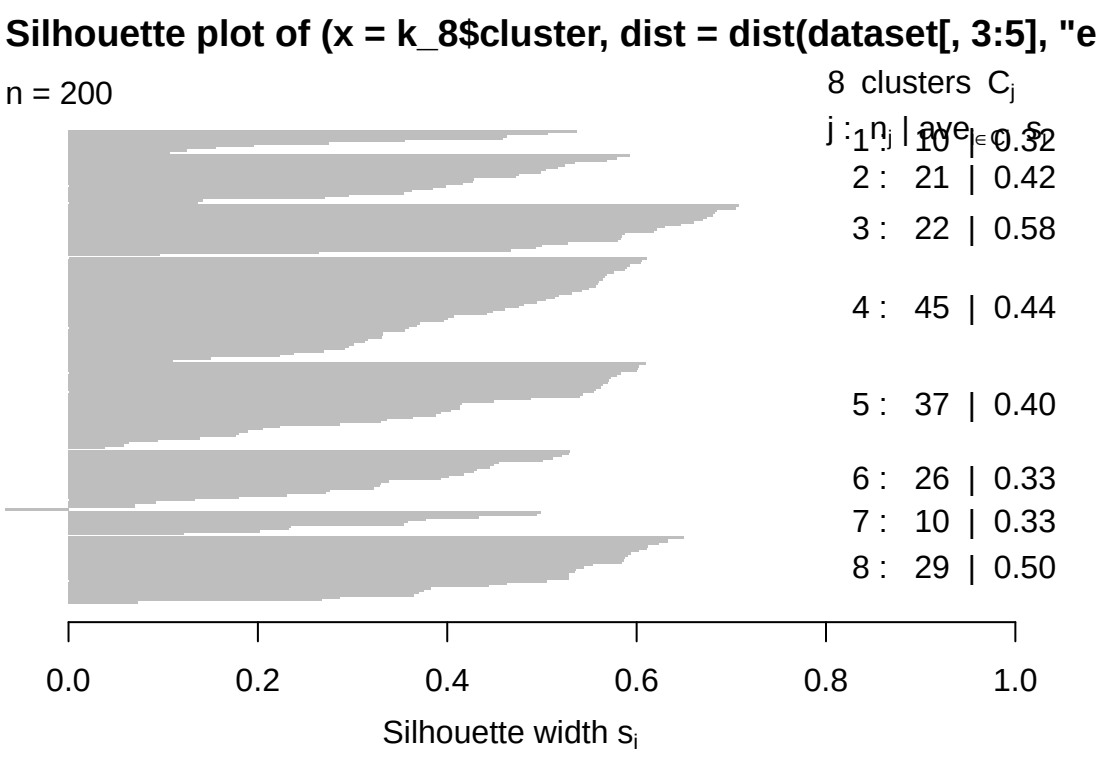

```
#optimal cluster of k = 6
k_6 <- kmeans(dataset[, 3:5], 6,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette6 <- plot(silhouette(k_6$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```




```
#optimal cluster of k = 7
k_7 <- kmeans(dataset[, 3:5], 7,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette7 <- plot(silhouette(k_7$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```




```
#optimal cluster of k = 8
k_8 <- kmeans(dataset[, 3:5], 8,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette8 <- plot(silhouette(k_8$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```

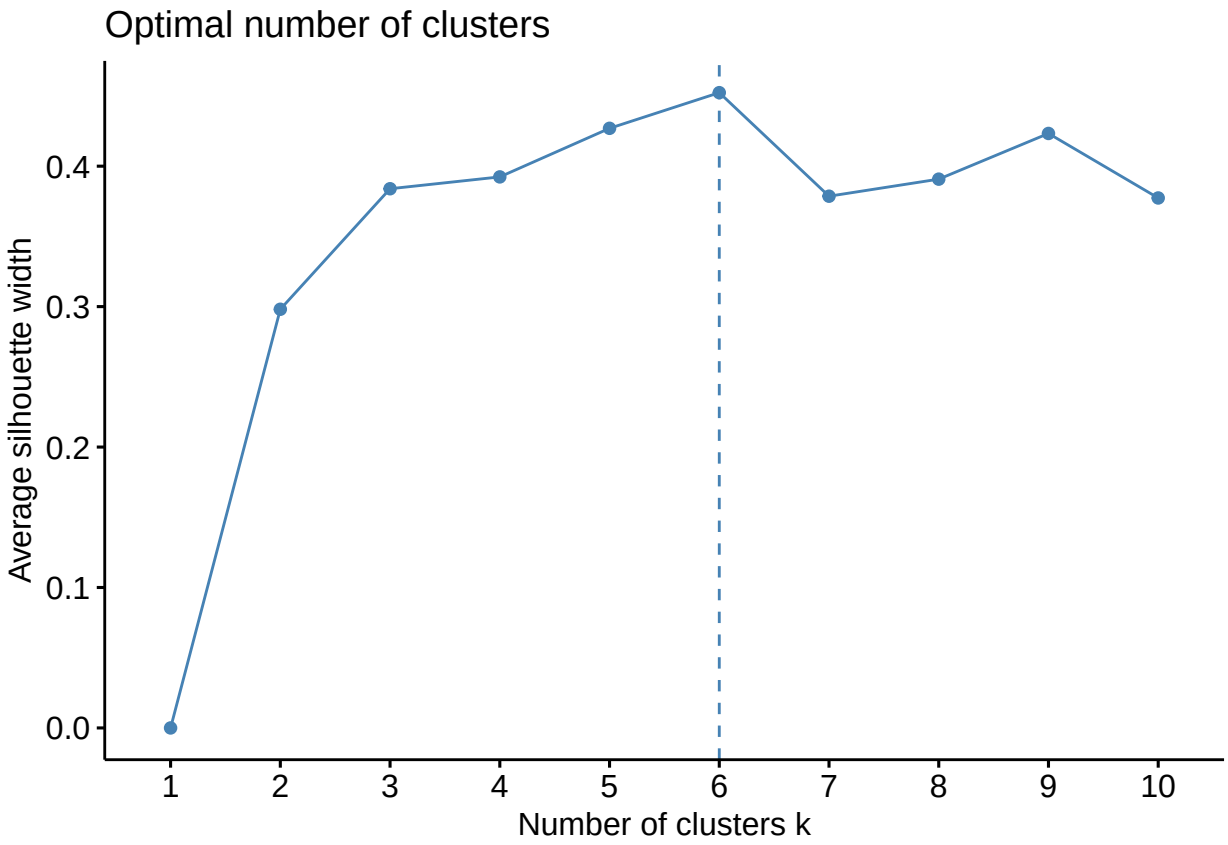



```
#optimal cluster of k = 9
k_9 <- kmeans(dataset[, 3:5], 9,
              iter.max = 100,
              nstart = 50,
              algorithm = "Lloyd")
silhouette9 <- plot(silhouette(k_9$cluster,
                               dist(dataset[, 3:5],
                                    "euclidean")))
```

Knowing that the highest average silhouette width indicates the best clustering, we can conclude that with a cluster k = 6, it produces the highest average silhouette width of 0.45.

Additionally, using the function "fvis_nbclust" allows visualization of optimal cluster numbers. This will visually strengthen the statistical claim that k = 6 is the optimal cluster.

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



4. Analyzing Clustering Data

In this final section, Principal Component Analysis (PCA) is utilized to uncover the low dimensional features existing in the rather large data set we are working with. This can help us narrow dimensionality and gain **deeper insight to the features of the data set**.

```
#principle component analysis
pca_kmeans <- prcomp(dataset[, 3:5],
                     scale = FALSE)
pca_kmeans
```

```
## Standard deviations (1, .., p=3):
## [1] 26.46251 26.15970 12.93169
##
## Rotation (n x k) = (3 x 3):
##                                PC1        PC2         PC3
## Age                      0.1889742 -0.1309652 0.973209570
## Annual.Income..k..      -0.5886410 -0.8083757 0.005516668
## Spending.Score..1.100. -0.7859965  0.5739136 0.229853647
```

```
#summary of pca
summary(pca_kmeans)
```

```
## 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
```

```
#principle components coordinates of observations
head(pca_kmeans$x[, 1:2])
```

```
##             PC1      PC2
## [1,] 31.8705078 33.00143
## [2,] -0.7633969 56.84387
## [3,] 57.4087256 13.12294
## [4,]  2.1698965 53.47790
## [5,] 32.1749197 30.38700
## [6,]  2.1782778 52.22658
```

```
#matrix of lodaings for variables
pca_kmeans$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
```

```
#cluster graph visualization
set.seed(12345)
ggplot(dataset,
       aes(x = Annual.Income..k.., y = Spending.Score..1.100.)) +
  geom_point(stat = "identity",
             aes(color = as.factor(k_6$cluster))) +
  ggtitle("Mall Customers Segmentation", subtitle = "K-Means Clustering") +
  xlab("Annual Income") +
  ylab("Spending Score") +
  labs(fill = "Cluster Number") +
  scale_color_discrete(name = "",
                       breaks = c("1", "2", "3", "4", "5", "6"),
                       labels = c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5", "Clus
```

4A. Cluster Analysis from Income-Spending Visualization

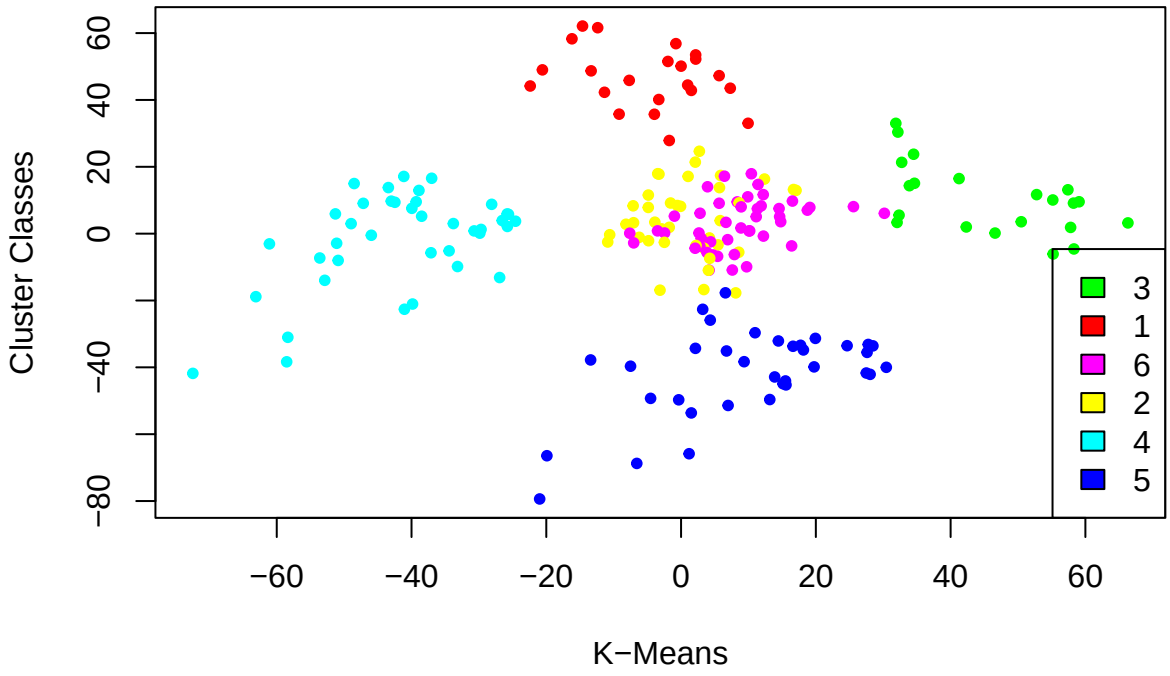
1. **Cluster 1:** this cluster forms the individuals in the customer data that have a low annual income and a high annual spending score
2. **Cluster 3:** this cluster forms the individuals in the customer data that have both a low annual income and a low yearly spending income
3. **Cluster 4:** this cluster forms the individuals in the customer data that have both a high yearly spending amount and a high annual income (opposite of Cluster 3)
4. **Cluster 5:** this cluster forms the individuals in the customer data that have a high annual income yet a low annual spending amount
5. **Cluster 2 and Cluster 6:** this cluster forms the individuals in the customer data that are in the middle range of income salary and annual spend

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

clust <- k_6$cluster
dig <- as.character(clust)

plot(pca_kmeans$x[, 1:2], col = kCols(clust),
     xlab = "K-Means",
     ylab = "Cluster Classes",
```

```
     pch = 20)
legend("bottomright", unique(dig), fill = unique(kCols(clust)))
```



4B. Cluster Analysis from PCA1/PCA2 Visualization

1. **Cluster 1:** this cluster forms the individuals in the customer data that have a high PCA2 and a medium range of annual spending income
2. **Cluster 3:** this cluster forms the individuals in the customer data that have both a high PCA1 and high PCA2 income
3. **Cluster 4:** this cluster forms the individuals in the customer data that have a low PCA1 and high PCA2
4. **Cluster 5:** this cluster forms the individuals in the customer data that have a low PCA2 score and a medium-range score of PCA1
5. **Cluster 2 and Cluster 6:** this cluster forms the individuals in the customer data that are in the middle range of both PCA1 and PCA2 scores

5. Final Notes

Throughout this project, we were able to analyze a given set of customer data through various means of statistical resources. Starting from basic data set analysis of customer information to PCA analysis and K-Means Clustering, we can understand the basis of the variables at a much deeper level. The **identification of customers and optimal cluster number** can have benefits to the real application of future companies as well.

While this is a model representing a set defined group of individuals, taken in a global context, companies can utilize this method to **tailor their products and brand to specific customers**. Parameters such as but not limited to age, gender, income, and spending patters can aid in the respective fitting group of customers. In addition, the PCA and K-Means Clustering methods will further help in segmentation and weaving of product reviews in future usage.