Customer Segmentation Analysis

Ahomagnon Romuald

2025-02-05

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
library(ggplot2)
library(cluster)
library(factoextra)
```

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

```
library(NbClust)
library(plotrix)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dbscan)
```

```
##
## Attaching package: 'dbscan'
```

```
## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

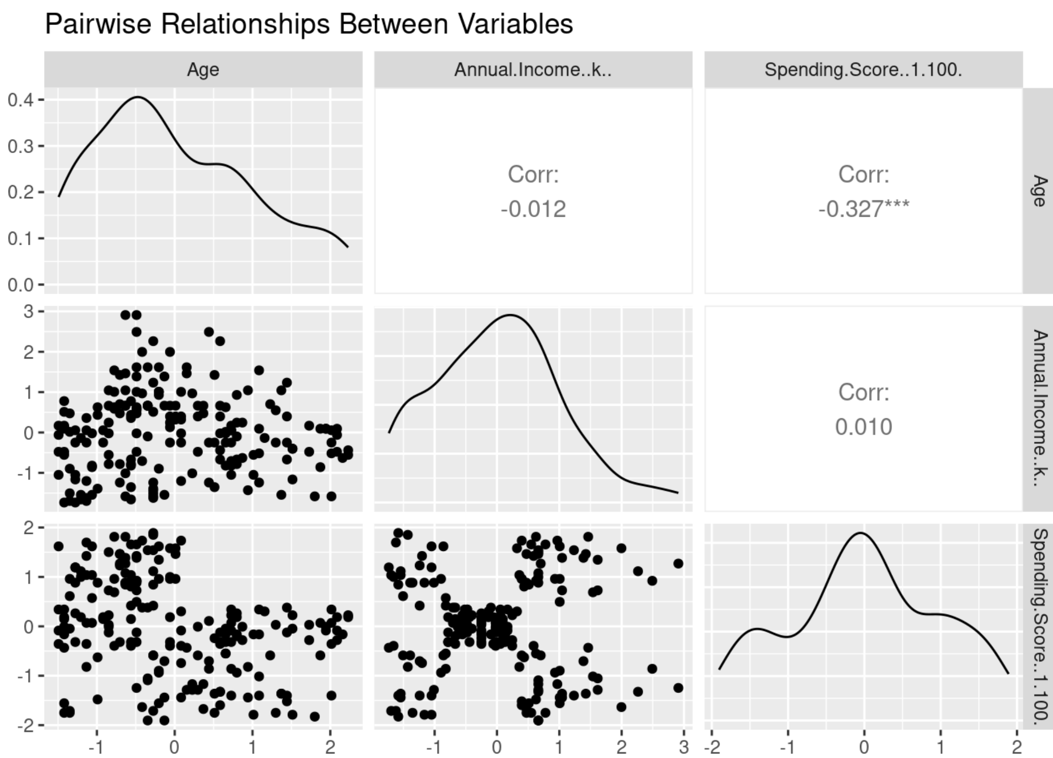
1. Data Loading and Preprocessing

```
# Load the dataset
Data_custumer <- read.csv("~/Downloads/Mes projets/customer-segmentation-dataset/Mall_Customers.csv")

# Normalize the data (scale Age, Annual Income, and Spending Score)
Data_custumer[, 3:5] <- scale(Data_custumer[, 3:5])

# Display the first few rows of the dataset
head(Data_custumer)
```

```
##   CustomerID Gender        Age Annual.Income..k.. Spending.Score..1.100.
## 1          1   Male -1.4210029          -1.734646             -0.4337131
## 2          2   Male -1.2778288          -1.734646              1.1927111
## 3          3 Female -1.3494159          -1.696572             -1.7116178
## 4          4 Female -1.1346547          -1.696572              1.0378135
## 5          5 Female -0.5619583          -1.658498             -0.3949887
## 6          6 Female -1.2062418          -1.658498              0.9990891
```

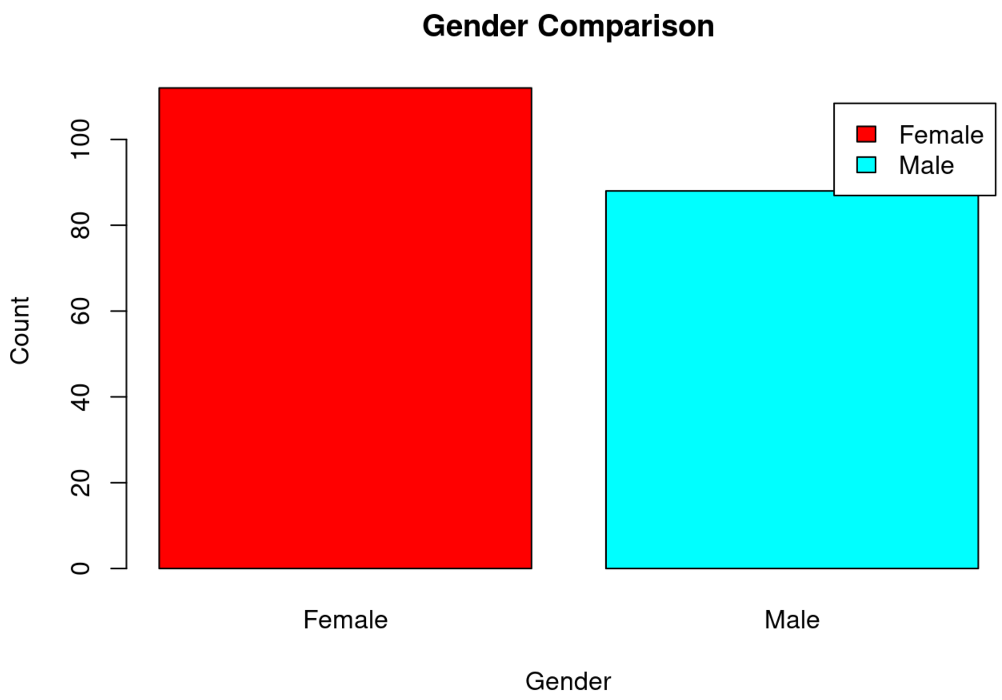
2. Data Exploration and Visualization Pairwise Relationships

```
# Visualize pairwise relationships between variables
ggpairs(Data_custumer[, 3:5], title = "Pairwise Relationships Between Variables")
```

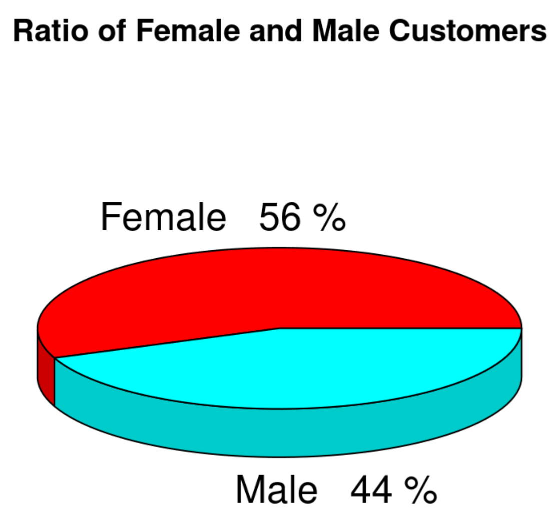


Gender Distribution

```
# Bar plot for gender distribution
a <- table(Data_custumer$Gender)
barplot(a, main = "Gender Comparison", ylab = "Count", xlab = "Gender", col = rainbow(2), legend = rownames
(a))
```



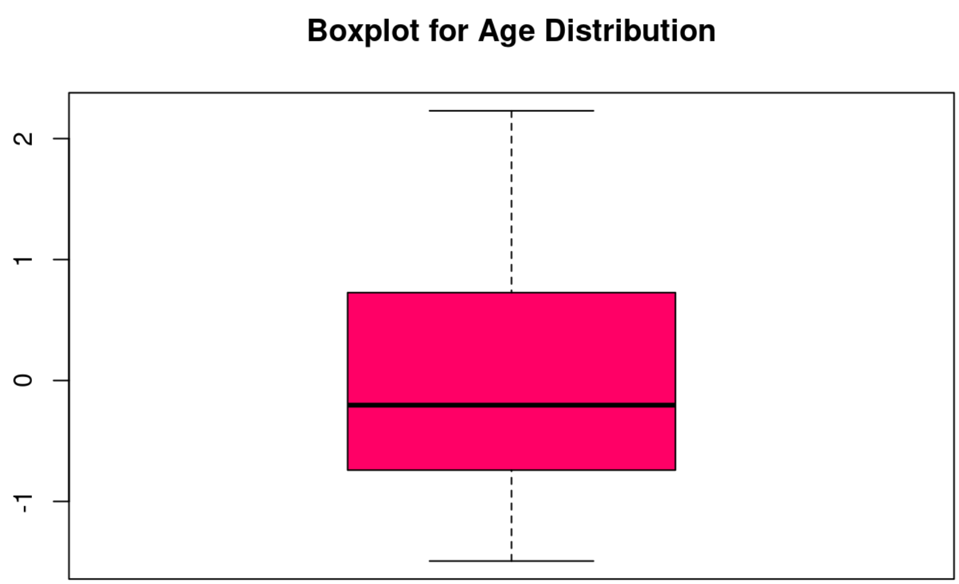

```
# 3D Pie Chart for gender distribution
pct <- round(a/sum(a)*100)
lbs <- paste(c("Female", "Male"), " ", pct, "%", sep = " ")
pie3D(a, labels = lbs, main = "Ratio of Female and Male Customers")
```



Age Distribution

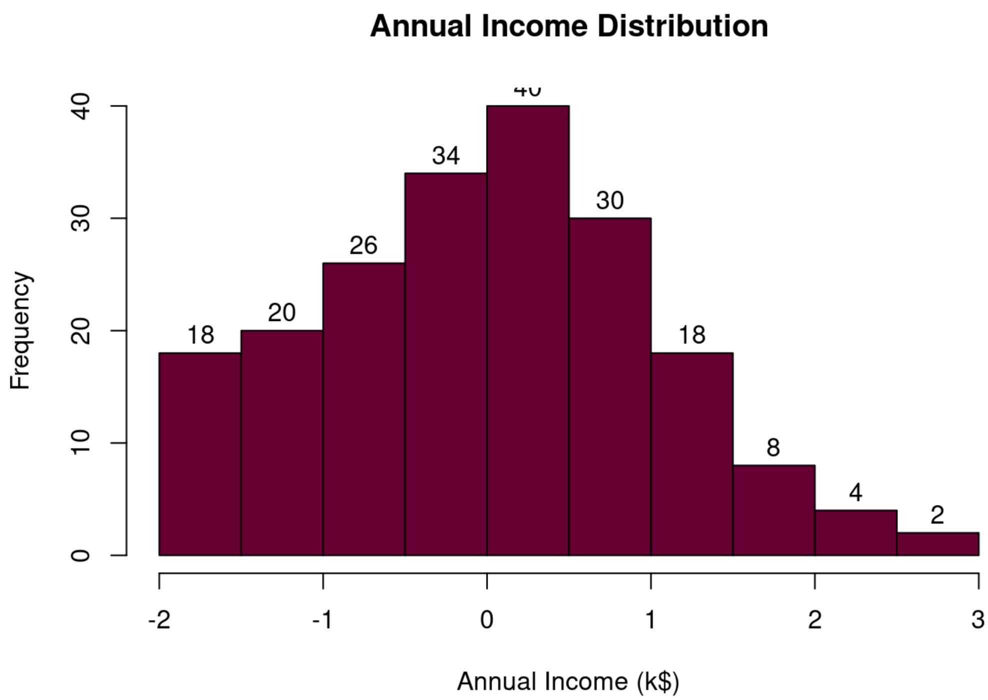
```
# Histogram for age distribution
hist(Data_custumer$Age, col = "blue", main = "Age Distribution", xlab = "Age Class", ylab = "Frequency", lab
els = TRUE)
```


```
# Boxplot for age distribution
boxplot(Data_custumer$Age, col = "#ff0066", main = "Boxplot for Age Distribution")
```

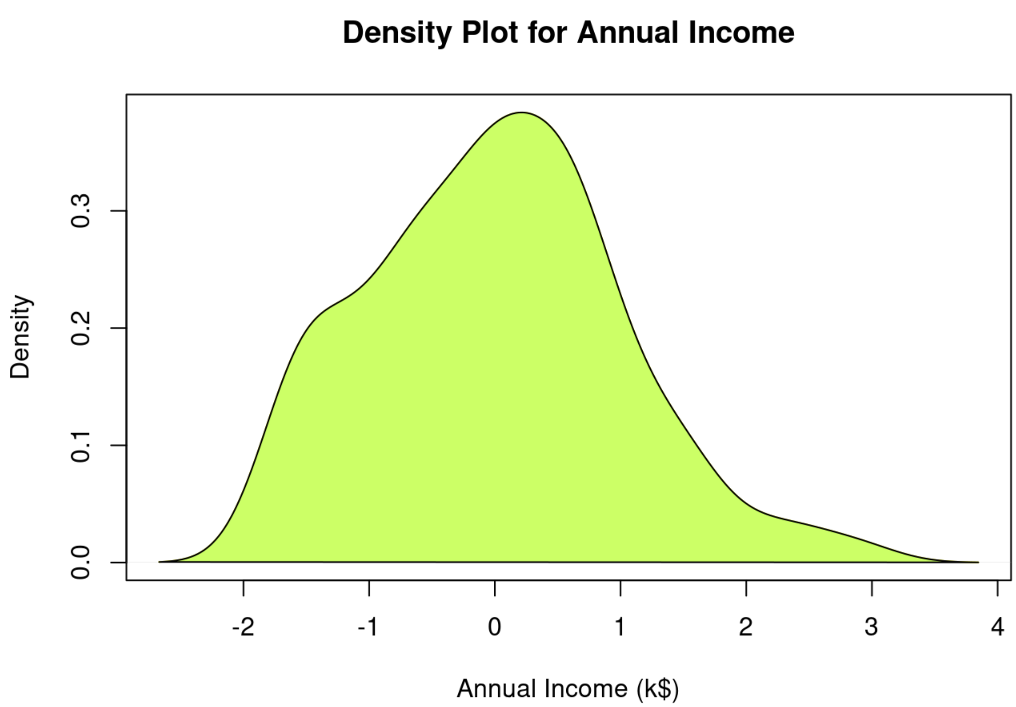


Annual Income Analysis

```
# Histogram for annual income
hist(Data_custumer$Annual.Income..k.., col = "#660033", main = "Annual Income Distribution", xlab = "Annual
Income (k$)", ylab = "Frequency", labels = TRUE)
```

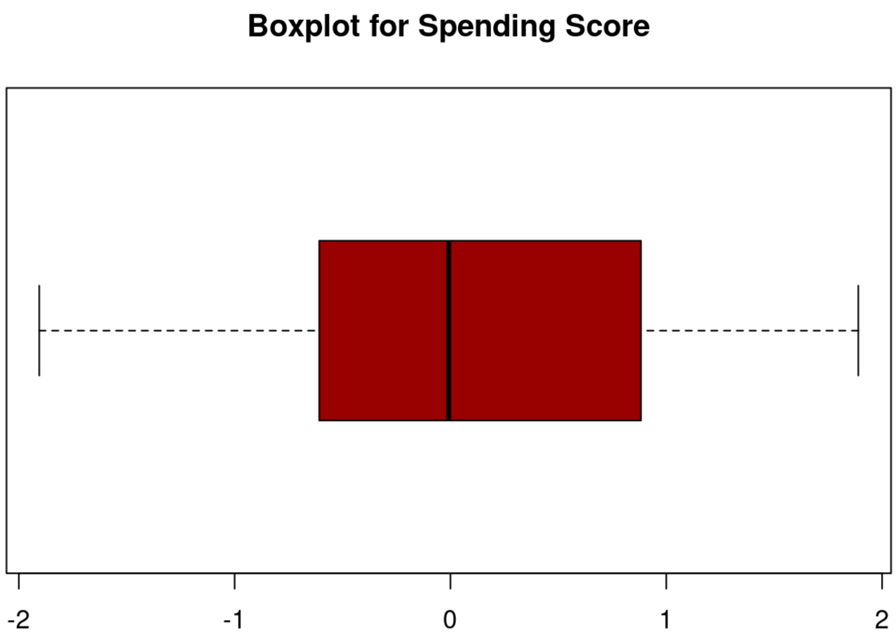



```
# Density plot for annual income
plot(density(Data_custumer$Annual.Income..k..), col = "yellow", main = "Density Plot for Annual Income", xla
b = "Annual Income (k$)", ylab = "Density")
polygon(density(Data_custumer$Annual.Income..k..), col = "#ccff66")
```

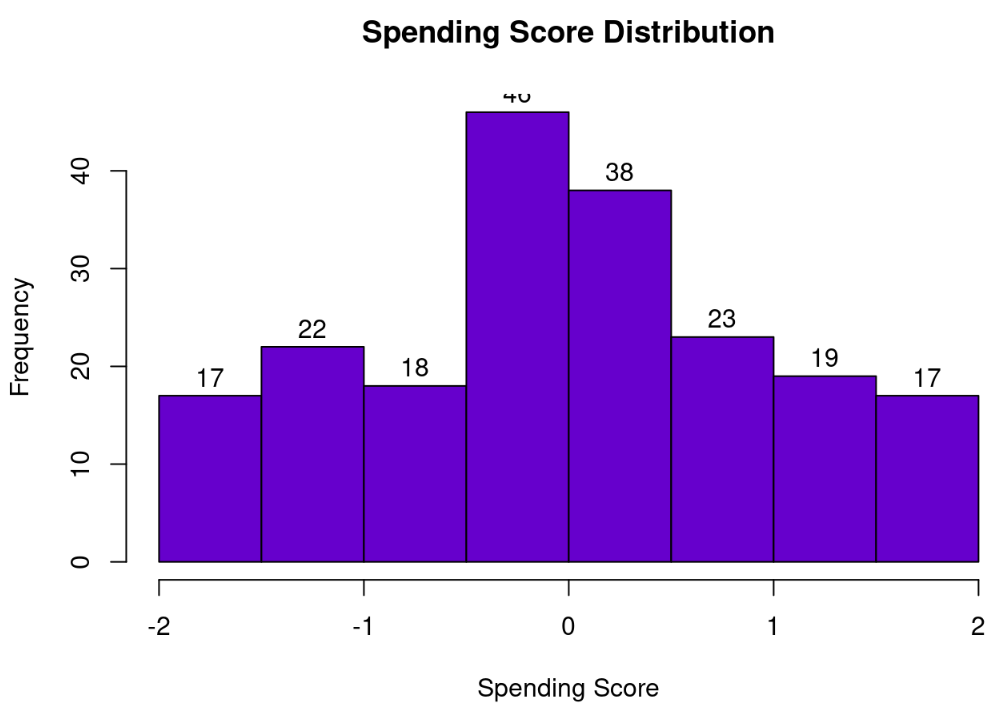


Spending Score Analysis

```
# Boxplot for spending score
boxplot(Data_custumer$Spending.Score..1.100., horizontal = TRUE, col = "#990000", main = "Boxplot for Spendi
ng Score")
```

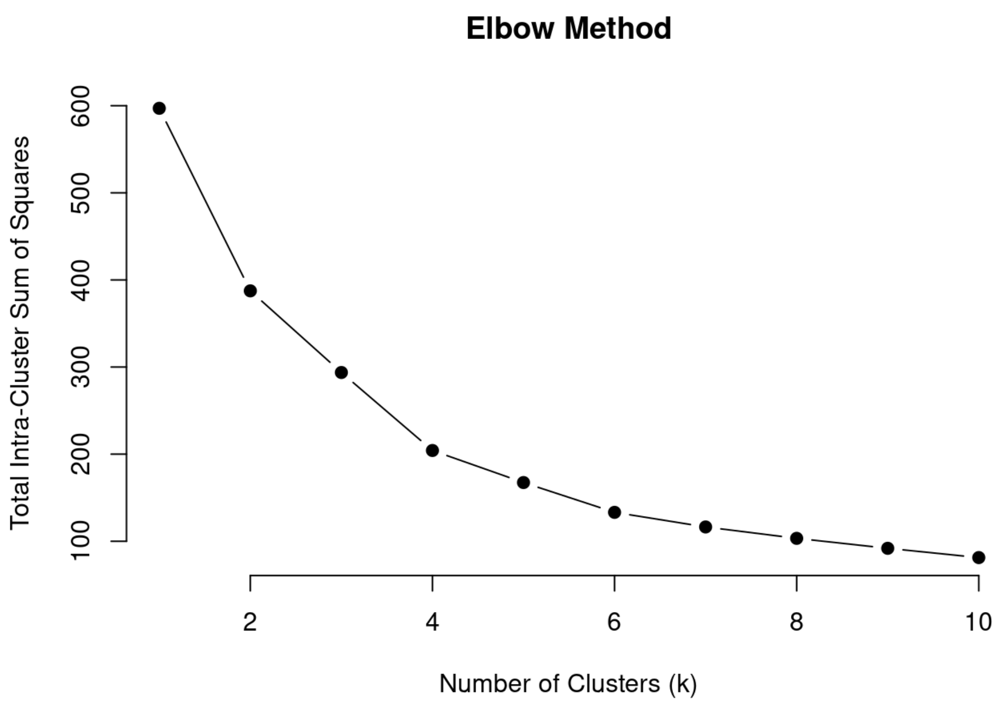



```
# Histogram for spending score
hist(Data_custumer$Spending.Score..1.100., main = "Spending Score Distribution", xlab = "Spending Score", yl
ab = "Frequency", col = "#6600cc", labels = TRUE)
```



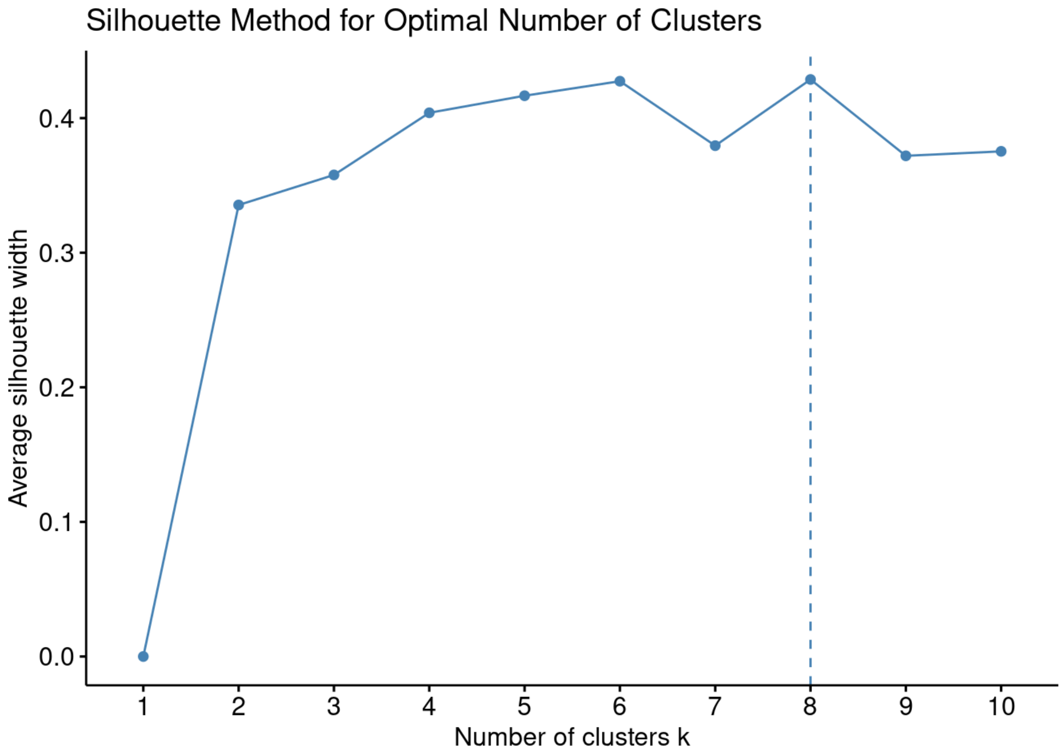
3. Clustering Analysis Elbow Method

```
# Function to calculate total intra-cluster sum of squares (WCSS)
iss <- function(k) {
  kmeans(Data_custumer[, 3:5], k, iter.max = 100, nstart = 100, algorithm = "Lloyd")$tot.withinss
}

# Compute WCSS for k = 1 to 10
k.values <- 1:10
iss_values <- sapply(k.values, iss)

# Plot the elbow curve
plot(k.values, iss_values, type = "b", pch = 19, frame = FALSE, xlab = "Number of Clusters (k)", ylab = "Tot
al Intra-Cluster Sum of Squares", main = "Elbow Method")
```



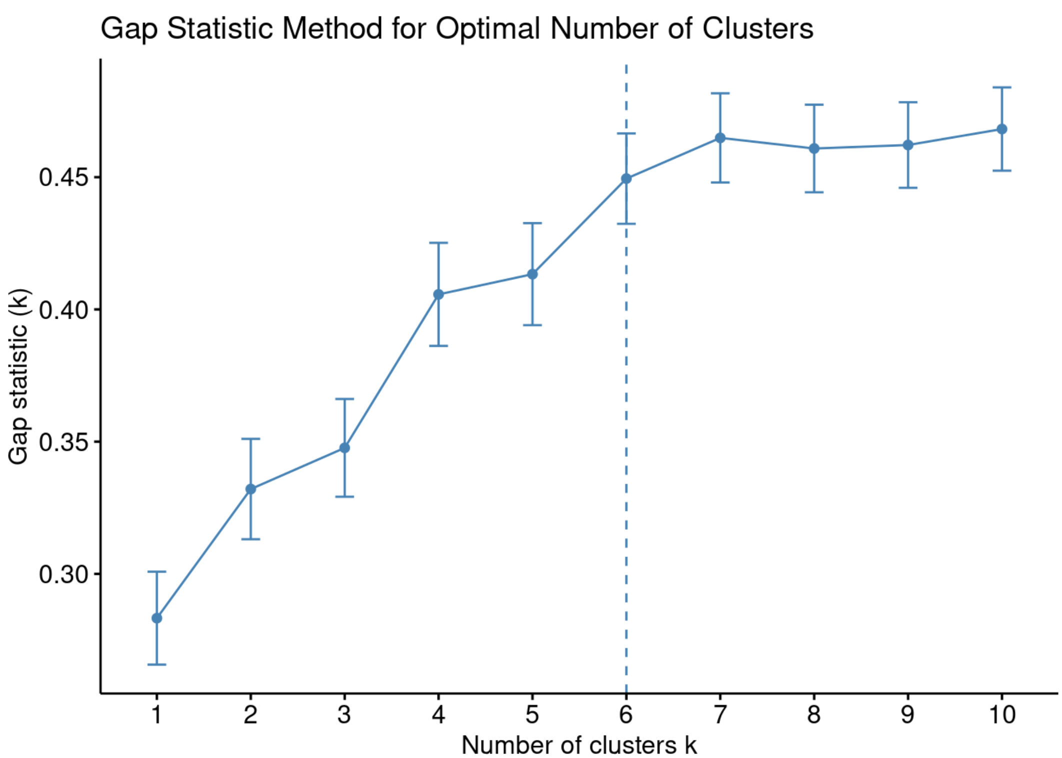
Silhouette Method

```
# Silhouette Method
fviz_nbclust(Data_custumer[, 3:5], kmeans, method = "silhouette") +
  ggtitle("Silhouette Method for Optimal Number of Clusters")
```



Gap Statistic Method

```
# Gap Statistic Method
set.seed(125)
stat_gap <- clusGap(Data_custumer[, 3:5], FUN = kmeans, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(stat_gap) +
  ggtitle("Gap Statistic Method for Optimal Number of Clusters")
```



Final K-means Clustering

```r
# Perform K-means clustering with the optimal number of clusters (k = 6)
k6 <- kmeans(Data_custumer[, 3:5], centers = 6, iter.max = 100, nstart = 50, algorithm = "Lloyd")
k6
```

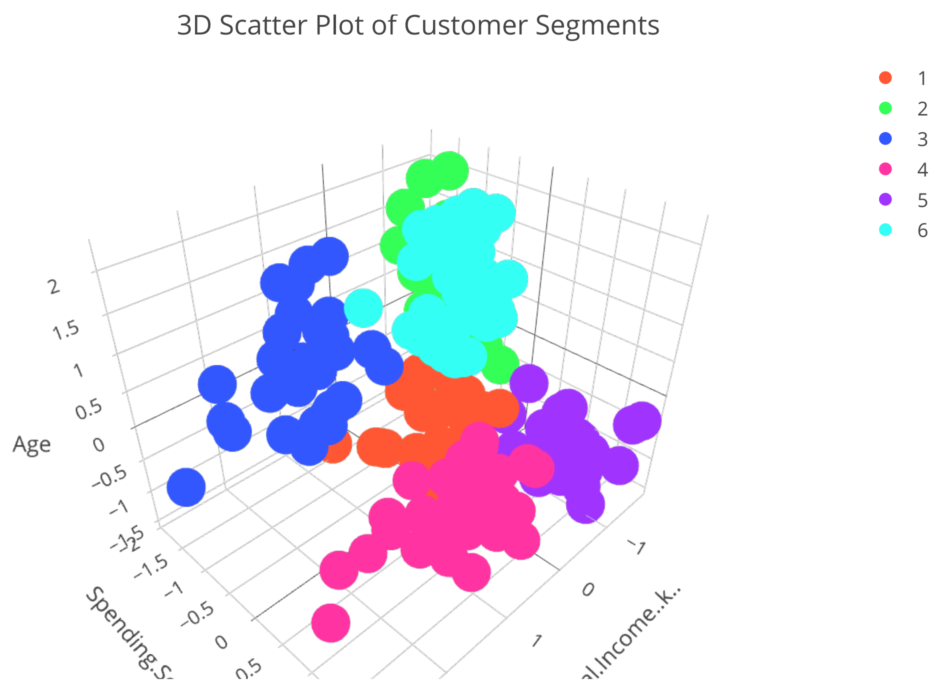
```
## K-means clustering with 6 clusters of sizes 38, 21, 33, 39, 24, 45
##
## Cluster means:
##          Age Annual.Income..k.. Spending.Score..1.100.
## 1 -0.8709130         -0.1135003            -0.09334615
## 2  0.4777583         -1.3049552            -1.19344867
## 3  0.2211606          1.0805138            -1.28682305
## 4 -0.4408110          0.9891010             1.23640011
## 5 -0.9735839         -1.3221791             1.03458649
## 6  1.2515802         -0.2396117            -0.04388764
##
## Clustering vector:
##   [1] 5 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2 5 2
##  [38] 5 2 5 6 5 2 5 2 5 6 1 1 1 6 1 1 6 6 6 6 6 1 6 6 1 6 6 6 1 6 6 1 1 6 6 6 6
##  [75] 6 1 6 1 1 6 6 1 6 6 1 6 6 1 1 6 6 1 6 1 1 1 6 1 6 1 1 6 6 1 6 1 6 6 6 6 6
## [112] 1 1 1 1 1 6 6 6 6 1 1 1 4 1 4 3 4 3 4 3 4 1 4 3 4 3 4 1 4 3 4 1 4 3 4 3 4
## [149] 3 4 3 4 3 4 3 4 3 4 3 4 6 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] 20.20990 20.52332 34.51630 22.36267 11.71664 23.87015
##  (between_SS / total_SS =  77.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

4. Cluster Validation Davies-Bouldin Index

```r
# Install and load the clusterSim package
install.packages("clusterSim")
library(clusterSim)

# Compute Davies-Bouldin Index for cluster validation
davies_bouldin <- index.DB(Data_custumer[, 3:5], k6$cluster)
davies_bouldin
```

```
## $DB
## [1] 0.9007268
##
## $r
## [1] 0.8713583 0.9822516 0.8713583 0.8350444 0.8620968 0.9822516
##
## $R
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]       Inf 0.8144616 0.8713583 0.8350444 0.8620968 0.6853413
## [2,] 0.8144616       Inf 0.8376771 0.5037513 0.6345348 0.9822516
## [3,] 0.8713583 0.8376771       Inf 0.6819170 0.4851627 0.8396163
## [4,] 0.8350444 0.5037513 0.6819170       Inf 0.6116218 0.6058149
## [5,] 0.8620968 0.6345348 0.4851627 0.6116218       Inf 0.5286585
## [6,] 0.6853413 0.9822516 0.8396163 0.6058149 0.5286585       Inf
##
## $d
##          1        2        3        4        5        6
## 1 0.000000 2.109195 2.010642 1.780152 1.656405 2.126812
## 2 2.109195 0.000000 2.401046 3.465635 2.659104 1.747927
## 3 2.010642 2.401046 0.000000 2.610214 3.548139 2.085518
## 4 1.780152 3.465635 2.610214 0.000000 2.380460 2.452154
## 5 1.656405 2.659104 3.548139 2.380460 0.000000 2.699336
## 6 2.126812 1.747927 2.085518 2.452154 2.699336 0.000000
##
## $S
## [1] 0.7292733 0.9885854 1.0227163 0.7572330 0.6987083 0.7283185
##
## $centers
##            [,1]       [,2]        [,3]
## [1,] -0.8709130 -0.1135003 -0.09334615
## [2,]  0.4777583 -1.3049552 -1.19344867
## [3,]  0.2211606  1.0805138 -1.28682305
## [4,] -0.4408110  0.9891010  1.23640011
## [5,] -0.9735839 -1.3221791  1.03458649
## [6,]  1.2515802 -0.2396117 -0.04388764
```

```
# Create a 3D scatter plot of clusters
plot_ly(Data_custumer, x = ~Annual.Income..k.., y = ~Spending.Score..1.100., z = ~Age, color = ~as.factor(k6
$cluster), colors = c("#FF5733", "#33FF57", "#3357FF", "#FF33A1", "#A133FF", "#33FFF5")) %>%
  layout(title = "3D Scatter Plot of Customer Segments")
```



Cluster Profiles

```
# Summarize cluster profiles
cluster_profiles <- aggregate(Data_custumer[, 3:5], by = list(k6$cluster), FUN = mean)
cluster_profiles
```

```
##   Group.1        Age Annual.Income..k.. Spending.Score..1.100.
## 1       1 -0.8709130         -0.1135003            -0.09334615
## 2       2  0.4777583         -1.3049552            -1.19344867
## 3       3  0.2211606          1.0805138            -1.28682305
## 4       4 -0.4408110          0.9891010             1.23640011
## 5       5 -0.9735839         -1.3221791             1.03458649
## 6       6  1.2515802         -0.2396117            -0.04388764
```

6. Conclusion

   Key Findings: Identified 6 distinct customer segments based on income, spending score, and age.

   Business Implications: Tailor marketing strategies to target each segment effectively.

Copy