Customer Segmentation Dataset

LA - 1 Exploratory Data Analysis

Submitted By:- Abhinandan S Vishwaroop,1NT21IS005, 5 A

Introduction

```
# Customer Segmentation is the process of splitting a customer base into multiple groups of i
ndividuals that share a similarity in ways a product is or can be marketed to them such as ge
nder, age, interests, demographics, economic status, geography, behavioral patterns, spending
habits and much more. Customer Segmentation is one the most important applications of unsuper
vised learning. Using clustering techniques, companies can identify the several segments of c
ustomers allowing them to target the potential user base. Companies use the clustering proces
s to foresee or map customer segments with similar behavior to identify and target potential 
user base.Here we seek to achieve Value-based segmentation, where it differentiates customers
by their economic value, grouping customers with the same value level into individual segment
s that can be distinctly targeted.
```

Let’s load the **Customer Segmentation Dataset** into R

```
# Load the Dataset

customer_data <- read.csv("C:\\Users\\abhiv\\OneDrive\\Desktop\\Mall_Customers.csv")



# Data Summary

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
```

```
#To find missing values
any(is.na(customer_data))
```

```
## [1] FALSE
```

It returned **False** means the dataset has no missing values.

```r
# Retrieves the names of all the columns present in the dataset

names(customer_data)
```

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

```r
# Gives the structure
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 ...
```

```r
# Outputs the initial rows of the dataset

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
# Outputs the summary of the dataset like max, min, Quartiles etc..

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
# Outputs the standard deviation of the dataset
sd(customer_data$Age)
```

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

```r
# Correlation Coefficient of Annual Income and Spending Score

cor(customer_data$Annual.Income..k.., customer_data$Spending.Score..1.100.)
```

```
## [1] 0.009902848
```

Let’s find the summary and standard deviation of the Annual Income and Age columns

```
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
```

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

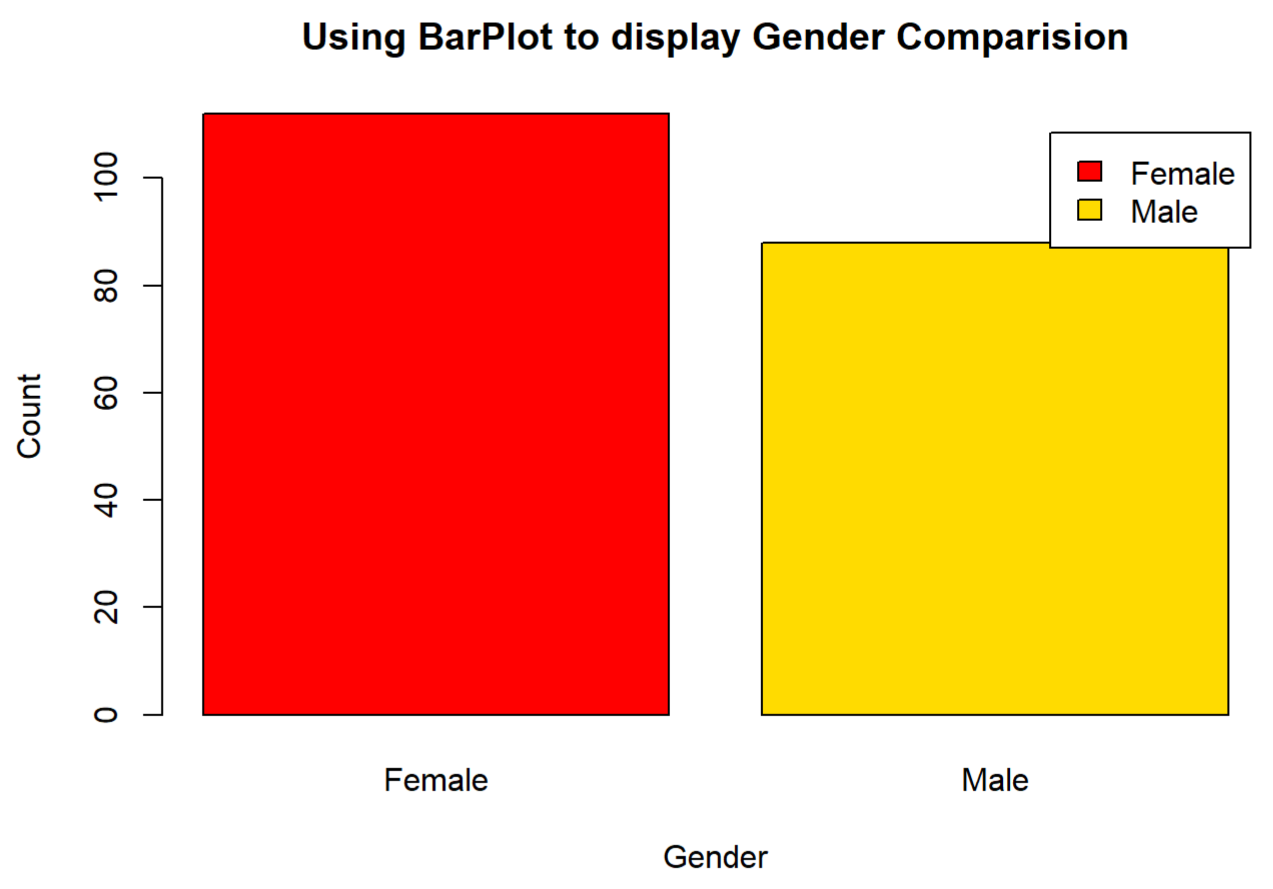
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   18.00   28.75   36.00   38.85   49.00   70.00
```

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

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

Customer Gender Visualization

```
a=table(customer_data$Gender)
barplot(a,main="Using BarPlot to display Gender Comparision",
        ylab="Count",
        xlab="Gender",
        col=rainbow(7),
        legend=rownames(a))
```

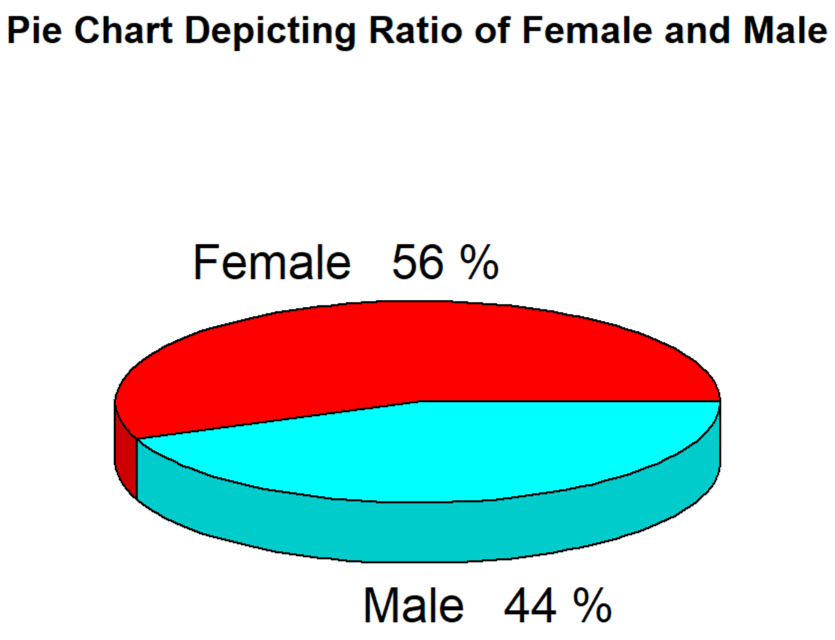


In R, the library **plotrix** is used for advanced plotting and visualization tasks, providing additional functionalities beyond the base plotting system. Plotrix offers a variety of specialized plotting functions and enhancements, making it particularly useful for creating complex and customized visualizations.

The library includes tools for creating 3D pie charts, polar plots, and color scales, among others. Additionally, it offers functionalities for labeling and annotating plots, which is valuable for improving the interpretability of visualizations.

The plotrix library extends R's plotting capabilities, making it a versatile tool for researchers and analysts who need more sophisticated and tailored visual representations of their data. Overall, the plotrix library enhances the flexibility and expressiveness of R's plotting capabilities, enabling users to create insightful and visually appealing plots for data analysis and presentation.

```
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")
```

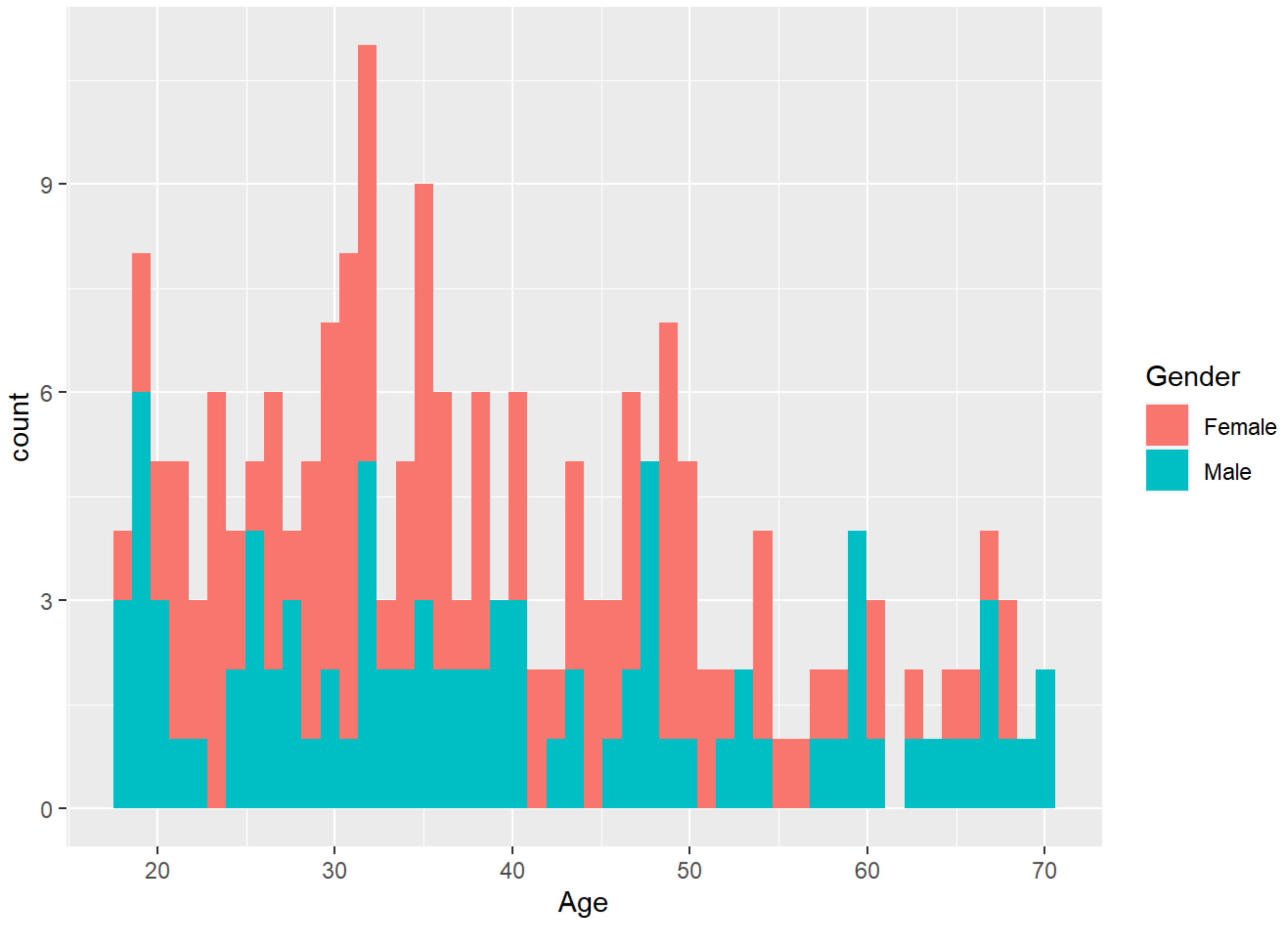


Data Visualizations

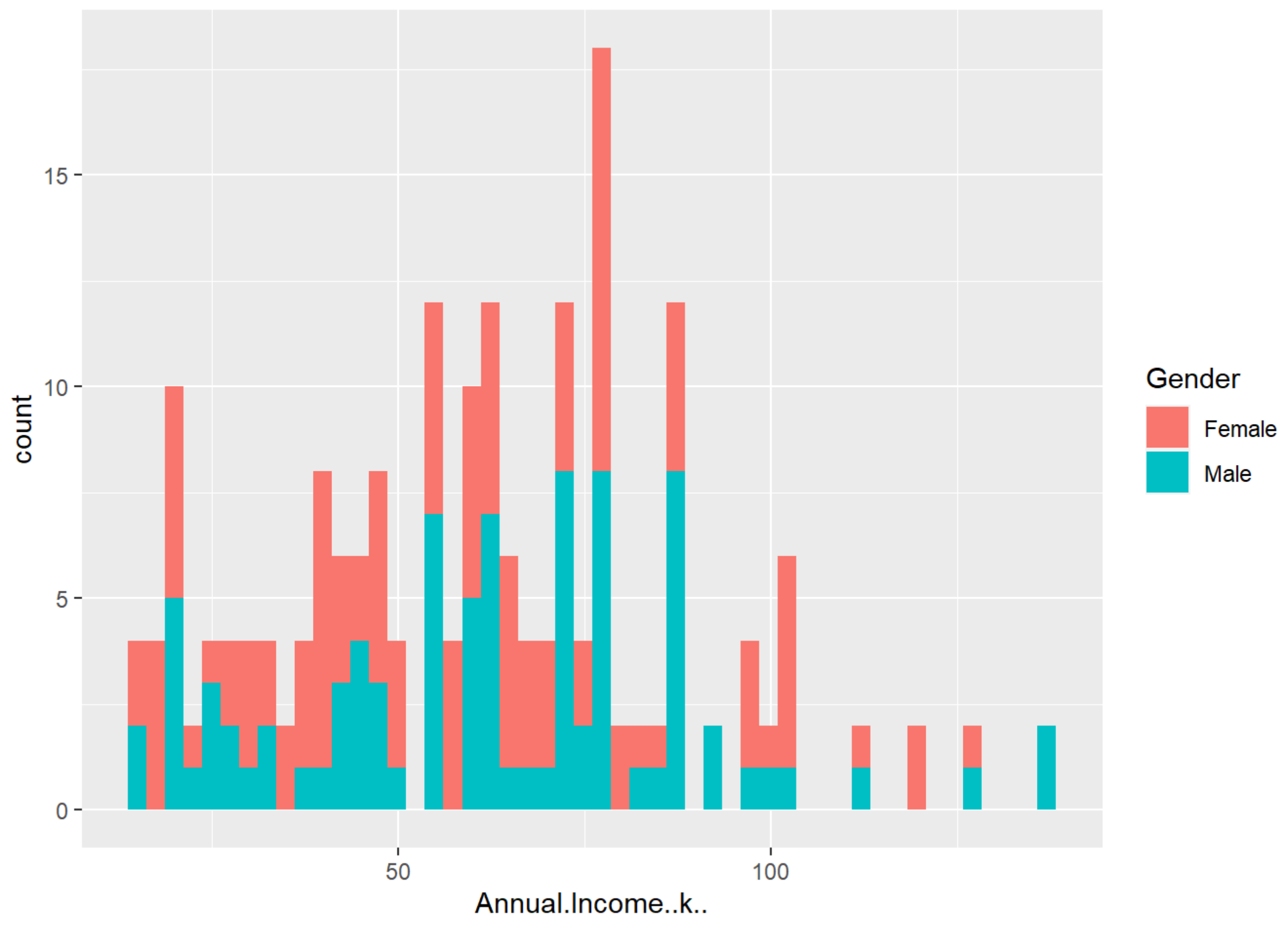
Plots

Histogram

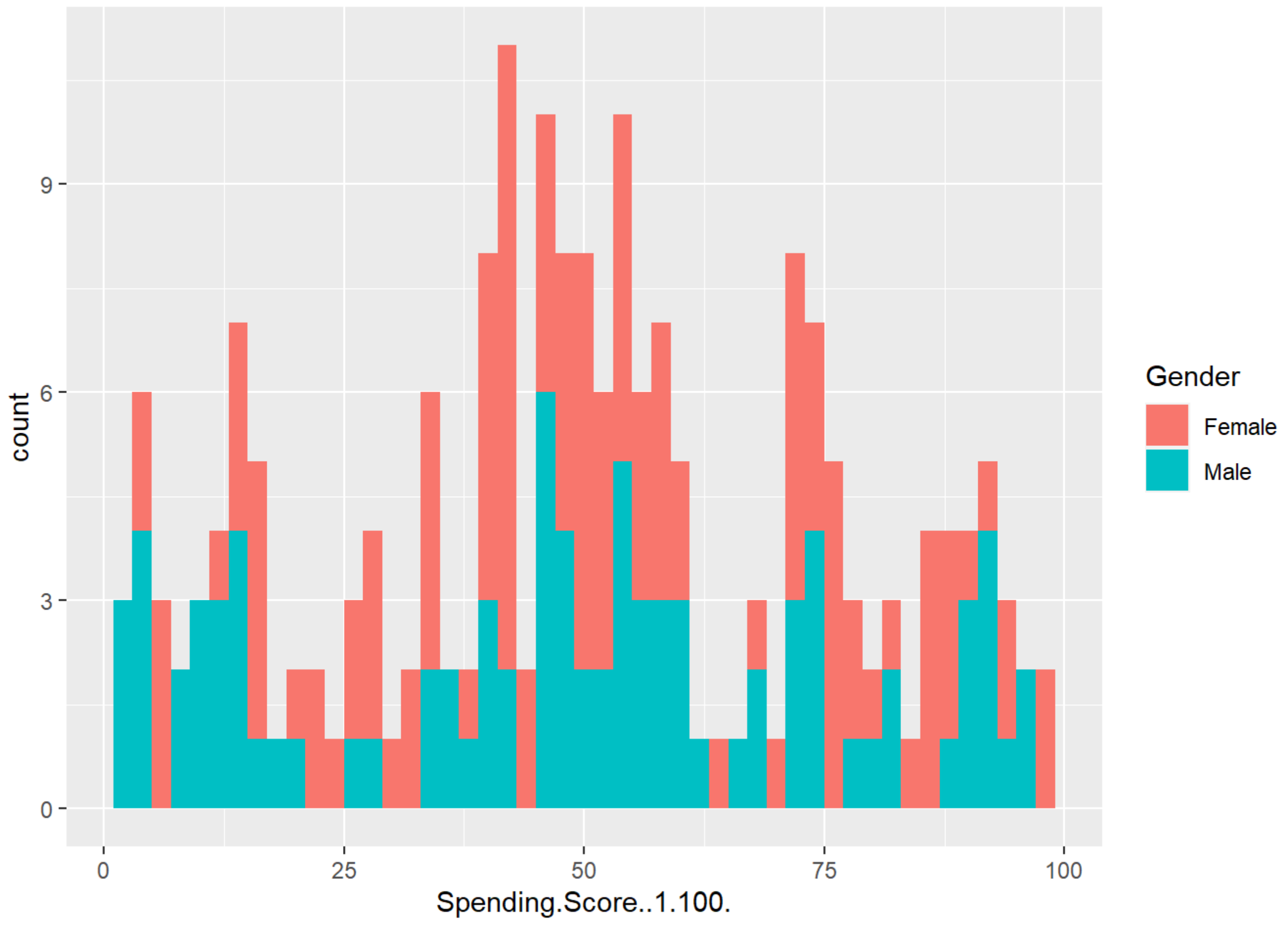
```
# Load the ggplot2 library

library(ggplot2)

# Histogram of Age filling Gender

ggplot(customer_data,aes(x= Age, fill=Gender))+geom_histogram(bins = 50)
```



```
# Histogram of Annual income filling Gender

ggplot(customer_data,aes(x= `Annual.Income..k..`,fill=Gender)) +geom_histogram(bins = 50)
```



```
# Histogram of Spending Score filling Gender

ggplot(customer_data,aes(x= `Spending.Score..1.100.`,fill=Gender)) +geom_histogram(bins=50)
```



Note:-

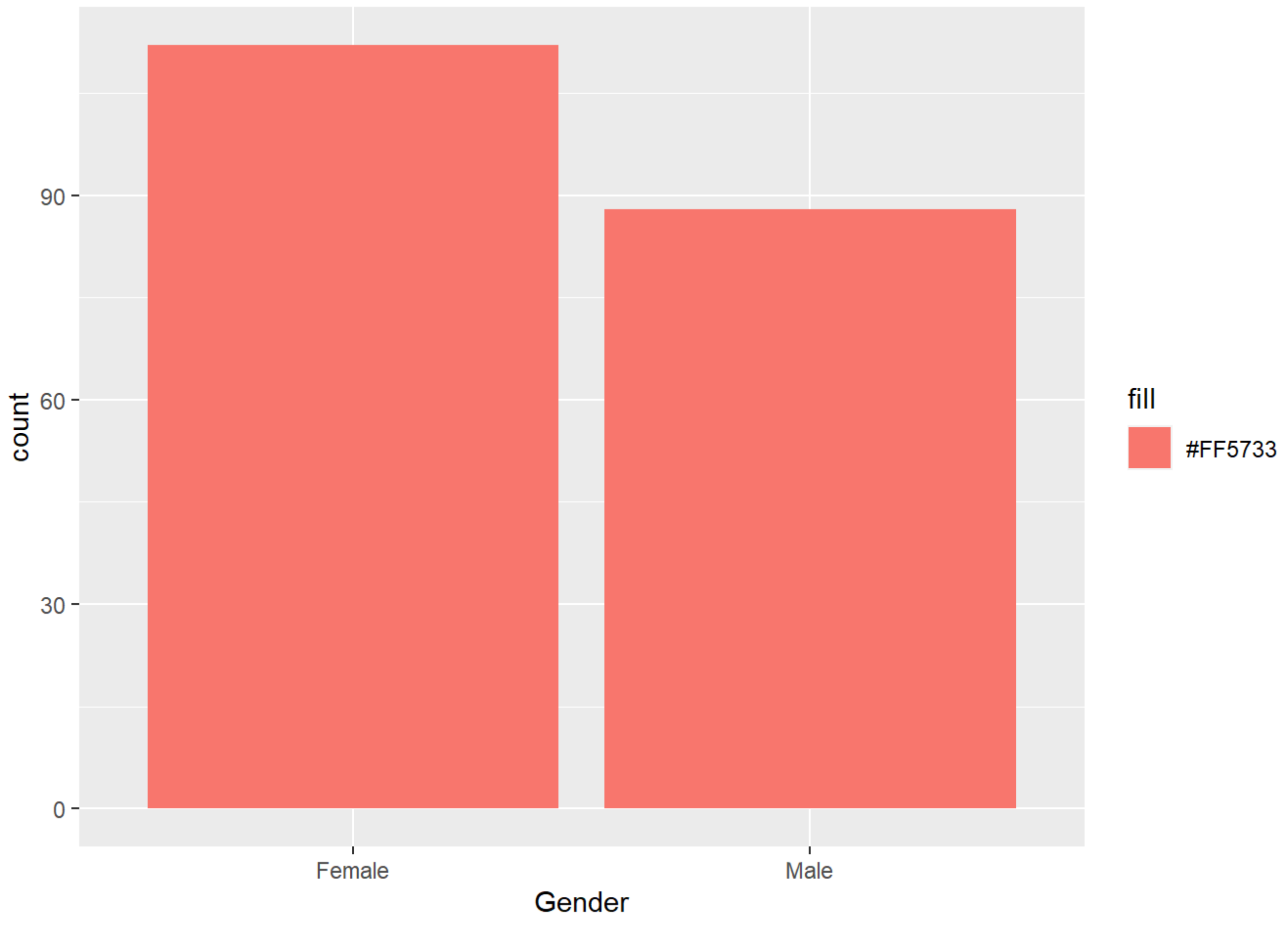
1. Most of the customers are between the age 30-35.
2. Minimum age of a customer is 18, whereas maximum age of a customer is 70.
3. The minimum spending score is 1, maximum is 99.

Analysis

From the histogram, we conclude that customers between class 40 and 50 have the highest spending score among all the classes.

Bar Graph / BarPlot

```
# generates a bar plot

ggplot(customer_data,aes(x= Gender, fill = "#FF5733")) + geom_bar ()
```



```
table(customer_data$Gender)
```

```
##
## Female   Male
##    112     88
```

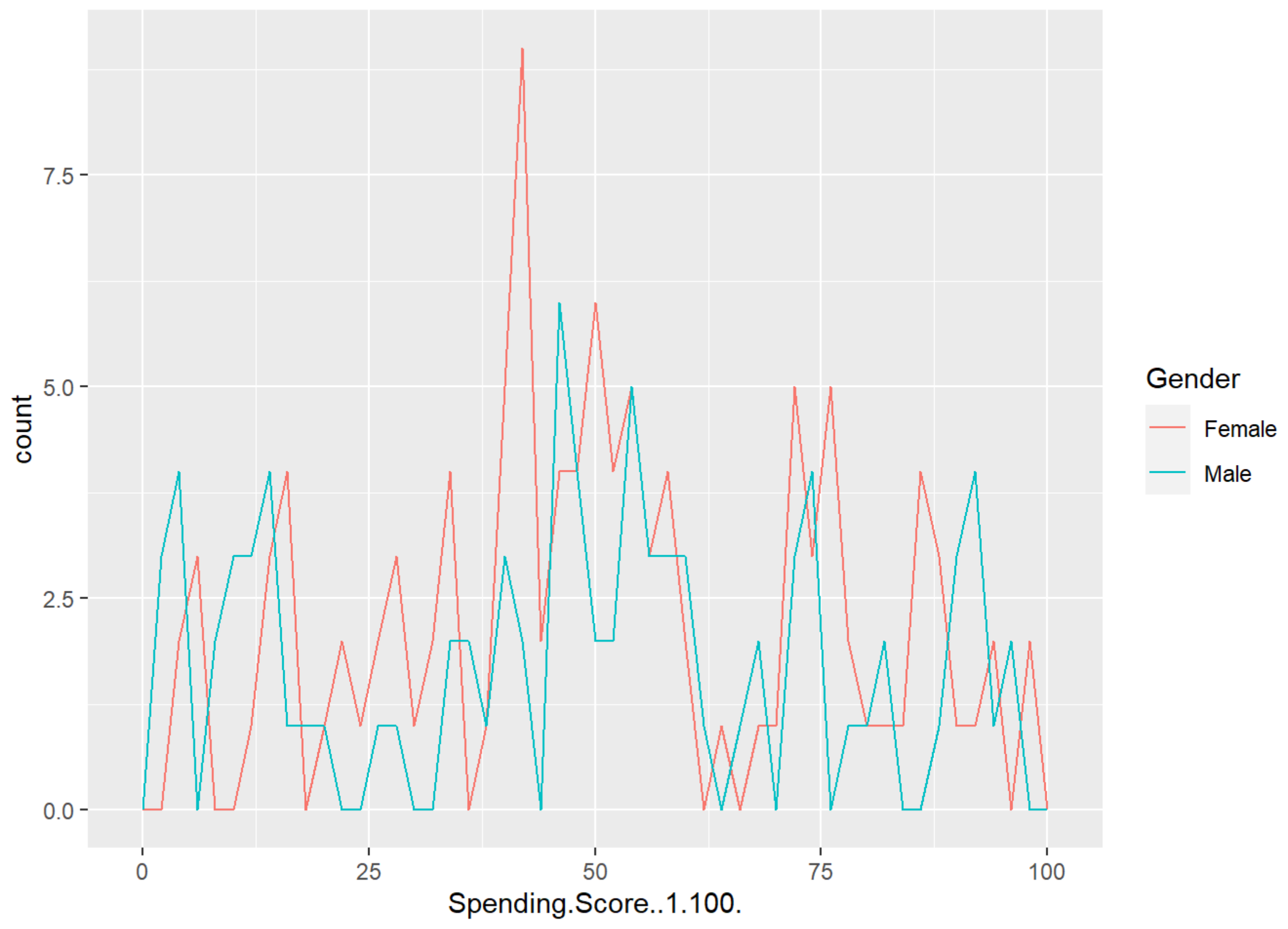
Analysis

From the BarPlot, we observe that there are **more** number of **Female** customers than ***Male** Customers.

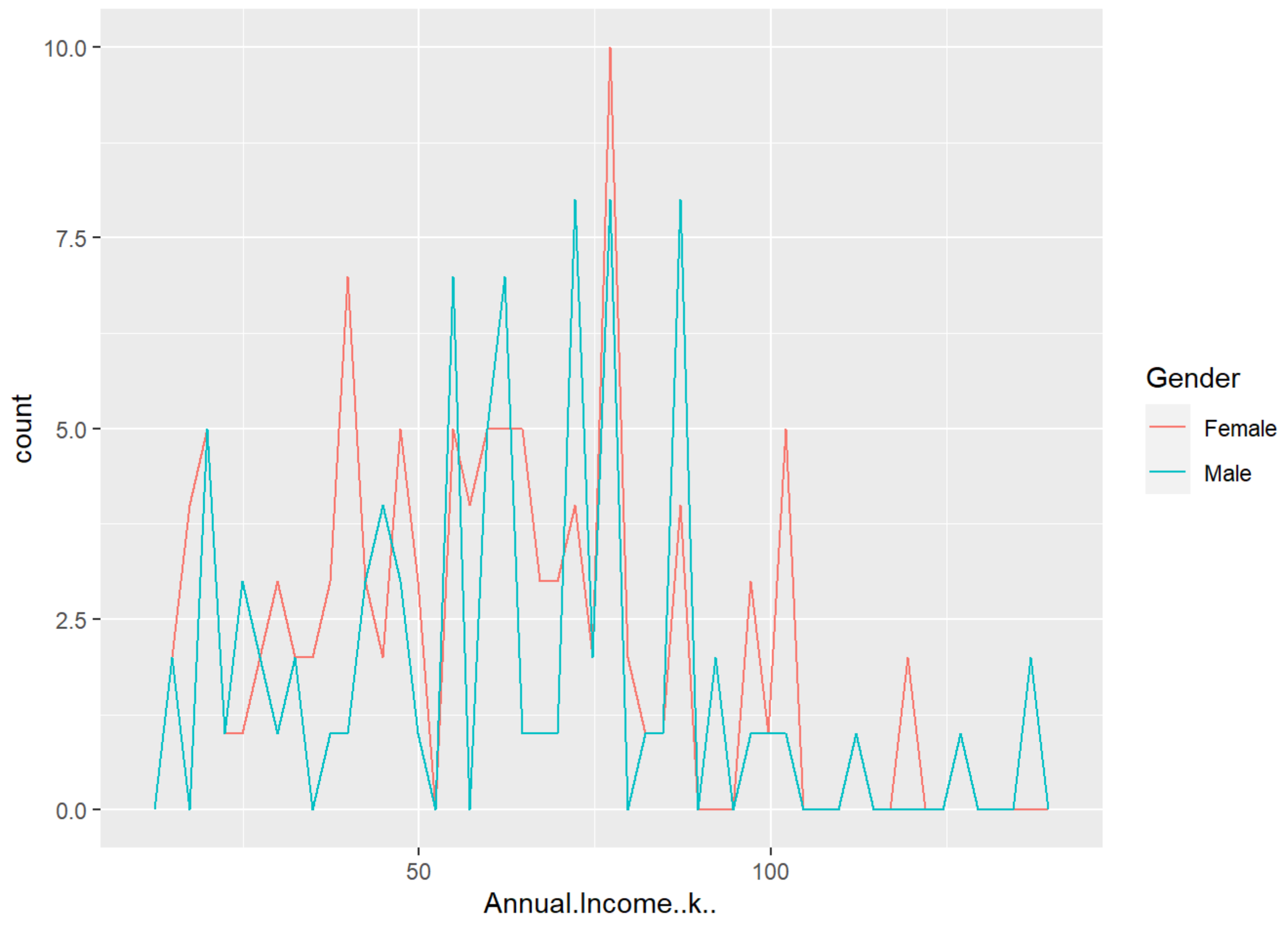
Frequency Ploygon

```
# Histograms and frequency polygons both depict data distributions, but differ in representat
ion. Histograms are bar charts, where bars represent the frequency of values within specified
bins. Adjacent bars emphasize data continuity.

# Frequency polygons, line graphs, connect midpoints of histogram bars, offering a smoothed c
urve for distribution trends. While histograms emphasize raw counts, frequency polygons provi
de a more continuous view of the underlying distribution, aiding in identifying patterns and
trends in datasets.

#The choice between them depends on the need for a detailed count-based representation (histo
grams) or a smoother visualization (frequency polygons).
```

```
ggplot(customer_data,aes(x= `Spending.Score..1.100.`, col=Gender)) + geom_freqpoly(bins=50)
```

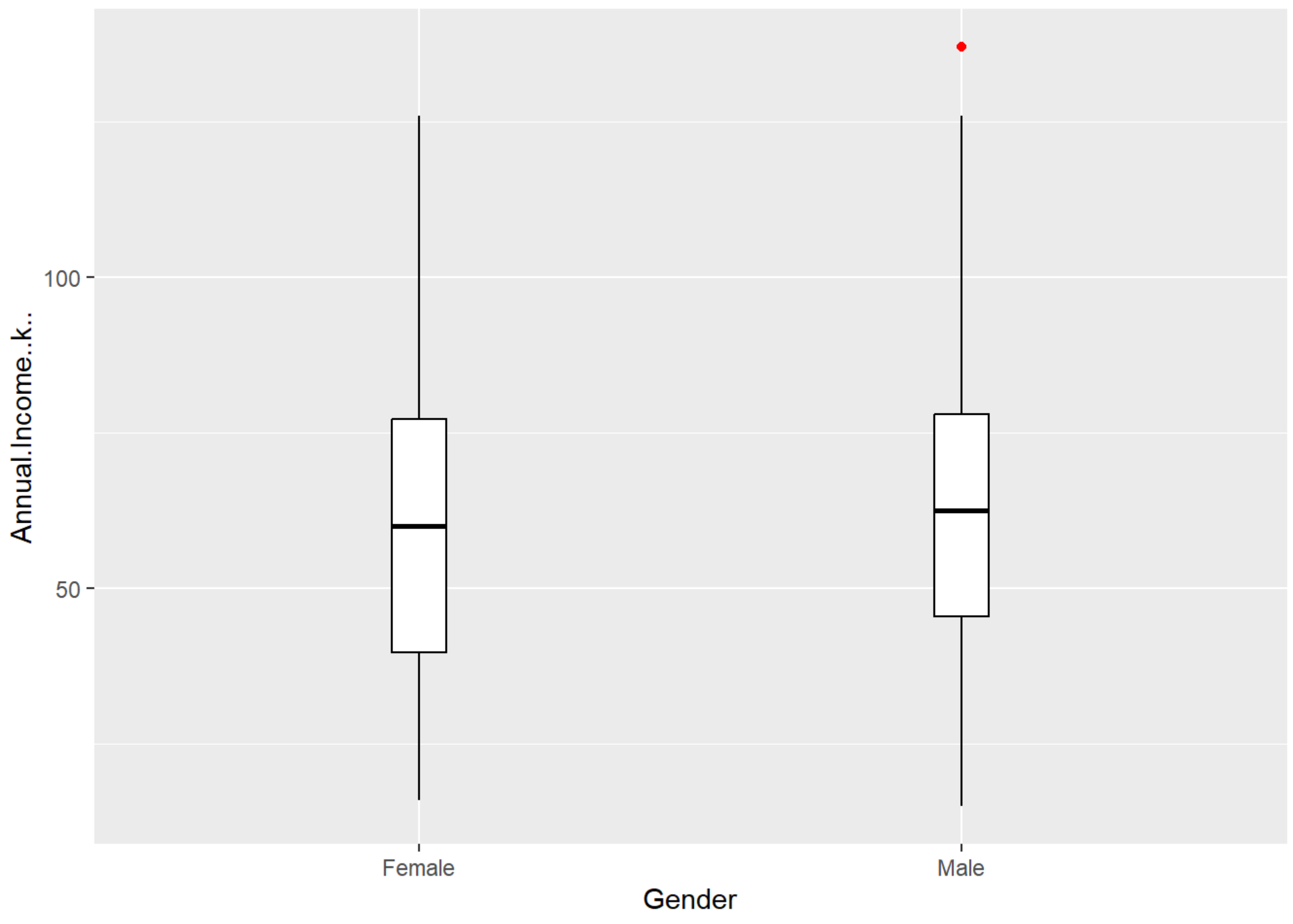


```
ggplot(customer_data,aes(x= `Annual.Income..k..`, col=Gender)) + geom_freqpoly(bins=50)
```

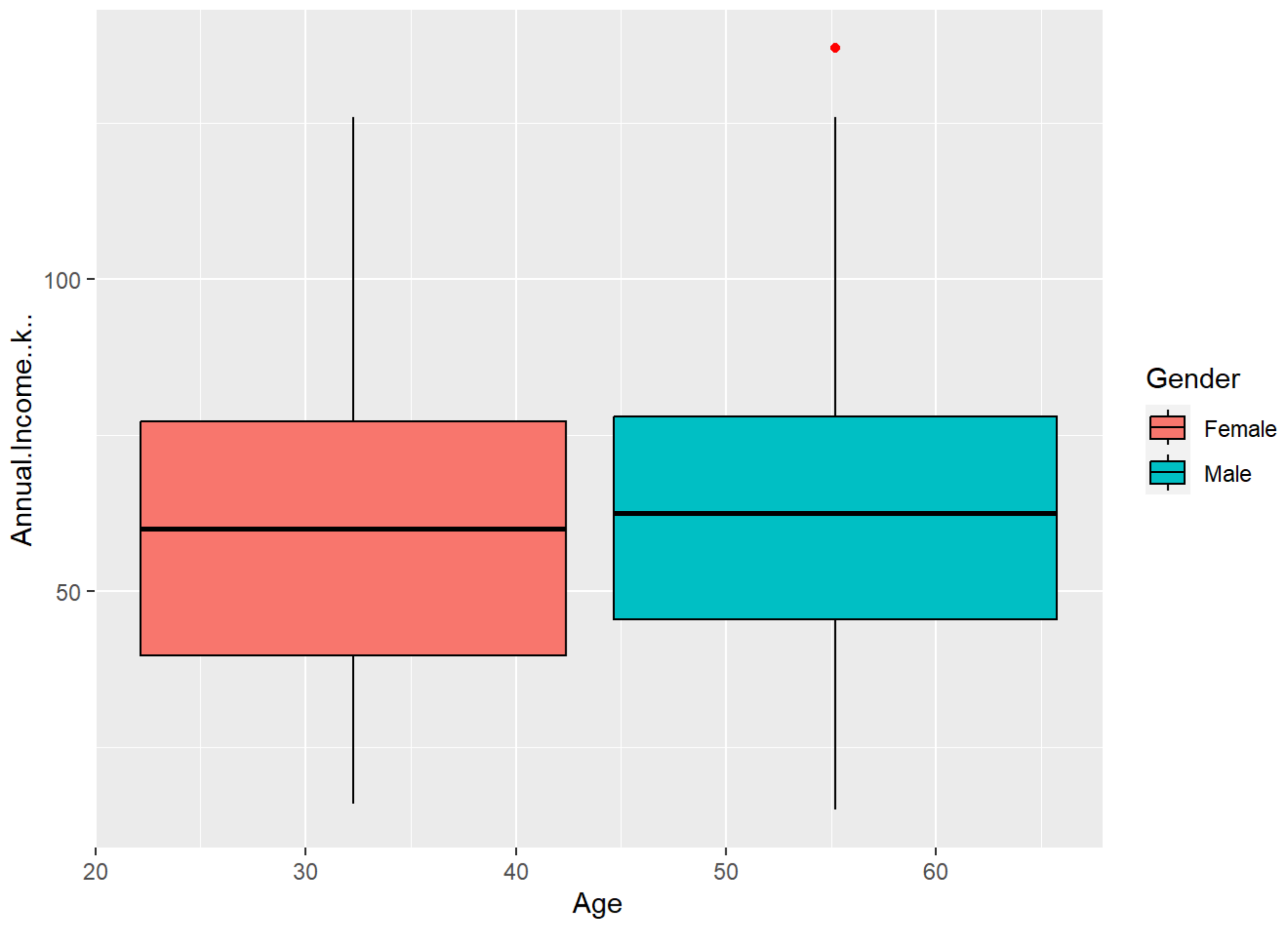


Box Plot

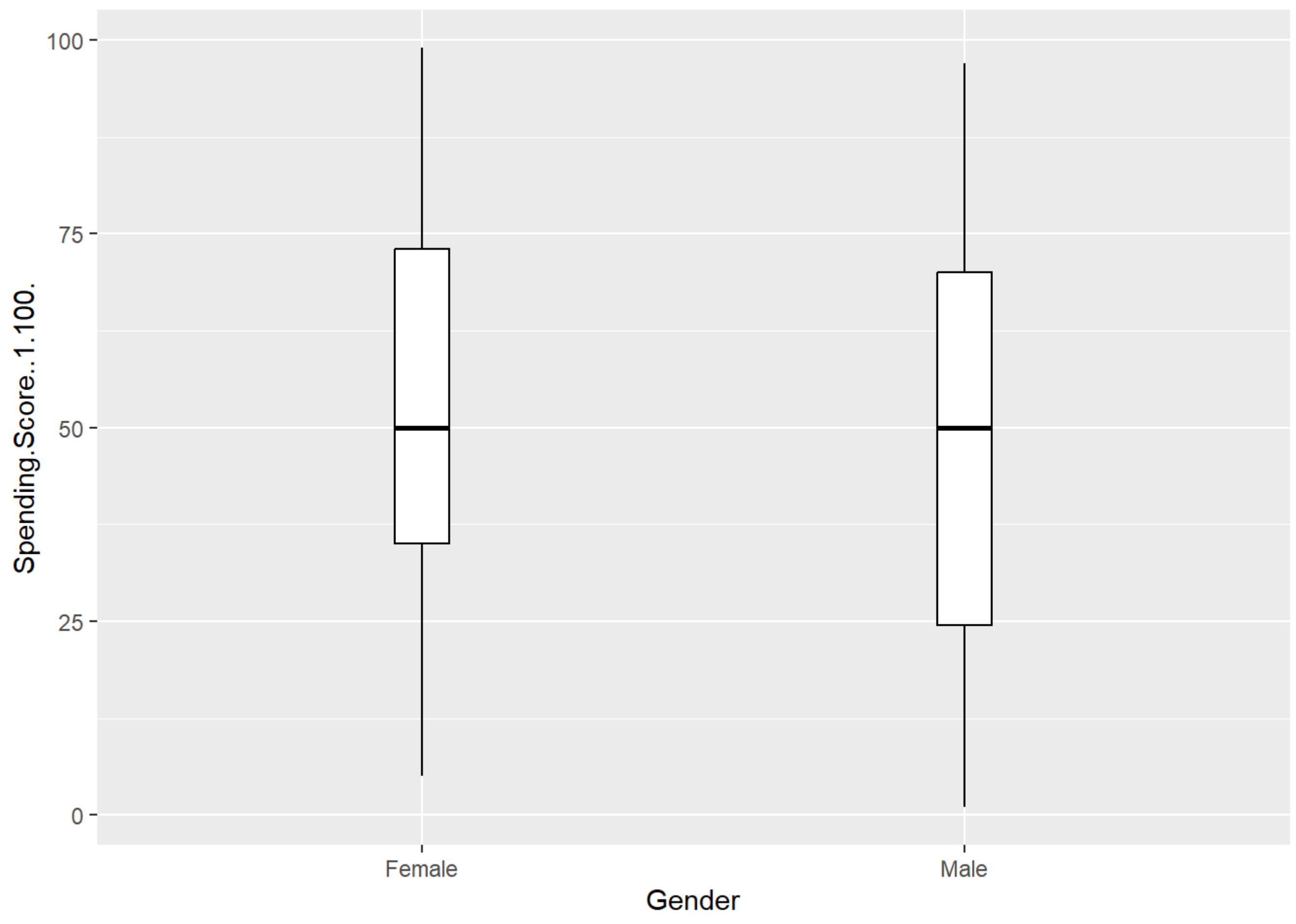
```
# Box Plot of Annual Income and Gender

ggplot(customer_data, aes(y =`Annual.Income..k..`, x= Gender)) +
  geom_boxplot(color = "black", outlier.color = "red", width = 0.1)
```



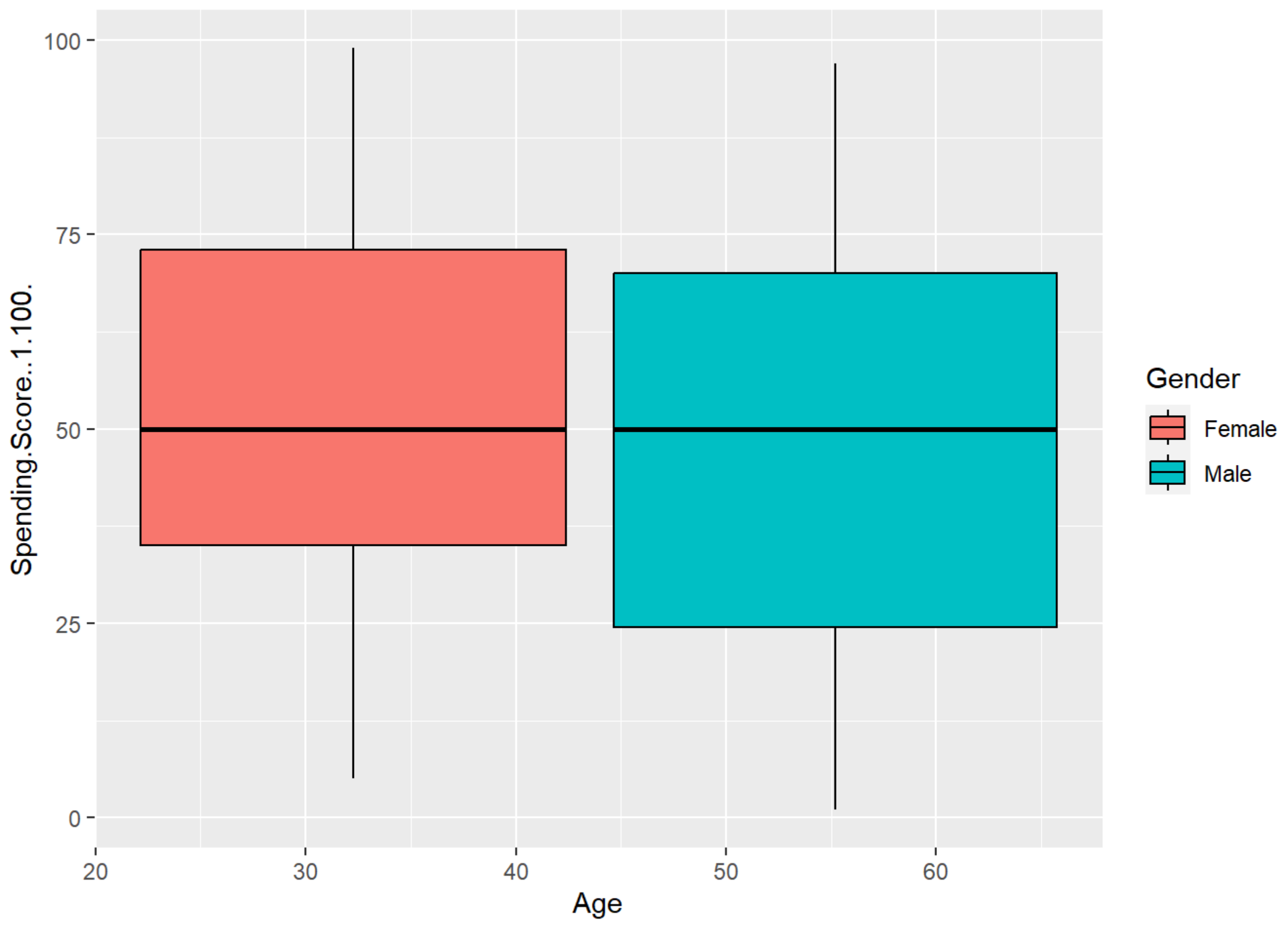
```
# Box Plot of Annual Income and Age differentiating with Gender

ggplot(customer_data, aes(y =`Annual.Income..k..`, x= Age, fill = Gender)) +
  geom_boxplot(color = "black", outlier.color = "red", width = 0.1)
```



```
# Box Plot of Spending Score and Gender

ggplot(customer_data, aes(y =`Spending.Score..1.100.`, x= Gender)) +
  geom_boxplot(color = "black", outlier.color = "red", width = 0.1)
```

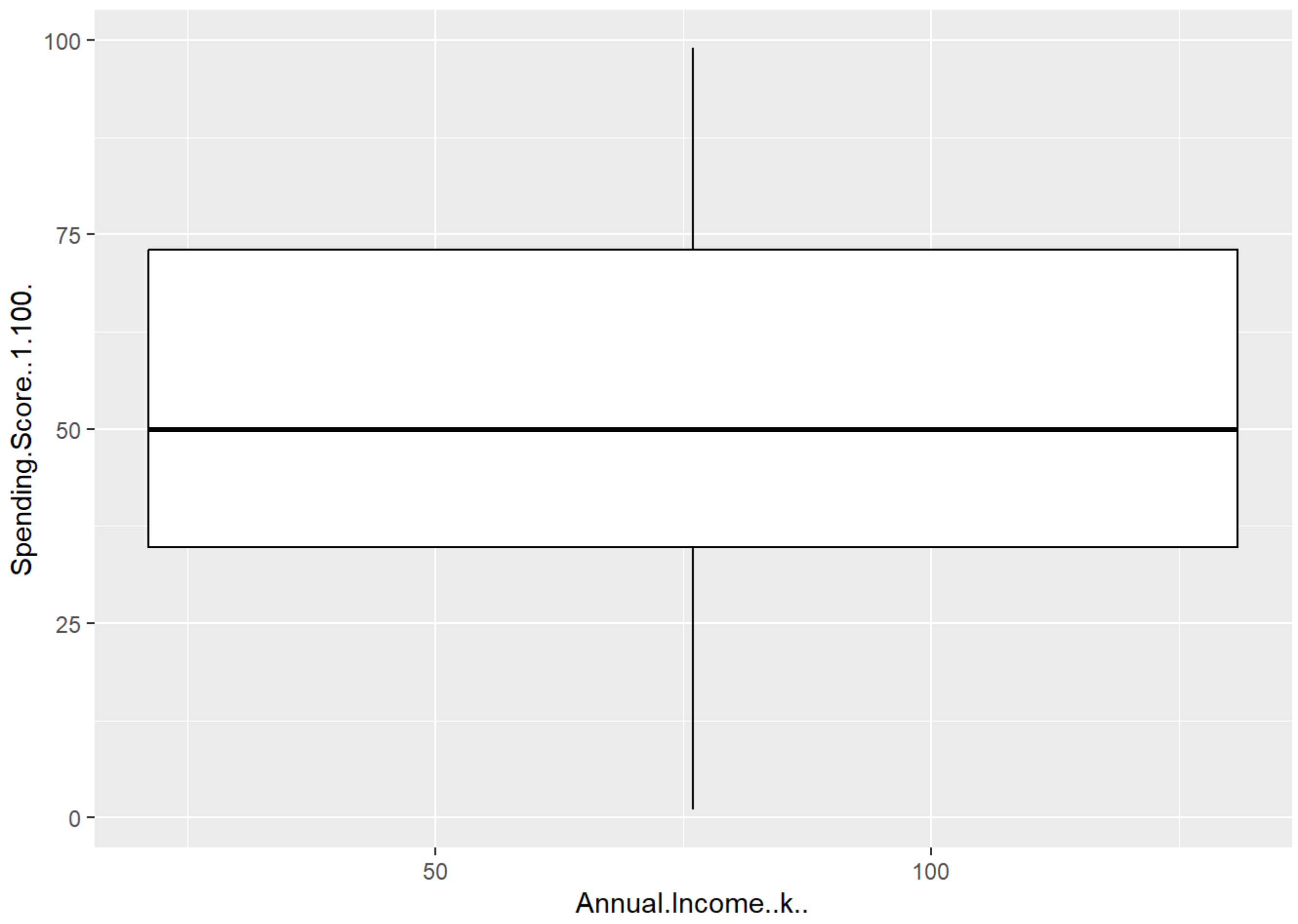


```
# Box Plot of Spending Score and Age differentiating with Gender

ggplot(customer_data, aes(y =`Spending.Score..1.100.`, x= Age, fill = Gender)) +
  geom_boxplot(color = "black", outlier.color = "red", width = 0.1)
```



```
# Box Plot of Spending Score and Annual Income

ggplot(customer_data, aes(y =`Spending.Score..1.100.`, x =`Annual.Income..k..` )) +
  geom_boxplot(color = "black", outlier.color = "red", width = 0.1)
```

```
## Warning: Continuous x aesthetic
## ℹ did you forget `aes(group = ...)`?
```



Correlation and Multi-Collinearity

Correlation is a statistical measure that indicates the extent to which two or more variables move together. A positive correlation indicates that the variables increase or decrease together. A negative correlation indicates that if one variable increases, the other decreases, and vice versa.

Collinearity is a linear association between two predictors. **Multicollinearity** is a situation where two or more predictors are highly linearly related. In general, an absolute correlation coefficient of >0.7 among two or more predictors indicates the presence of multicollinearity.

Though correlation talks about bivariate linear relationship whereas multicollinearity are multivariate, if not always, correlation matrix can be a good indicator of multicollinearity and indicate the need for further investigation.

```
# The GGally library in R is an extension to the popular ggplot2 package and is designed for
exploring and visualizing relationships in multivariate datasets. It provides a set of functi
ons and tools for creating scatterplot matrices, correlation plots, and other types of visual
izations to help analyze the patterns and associations between variables.
```

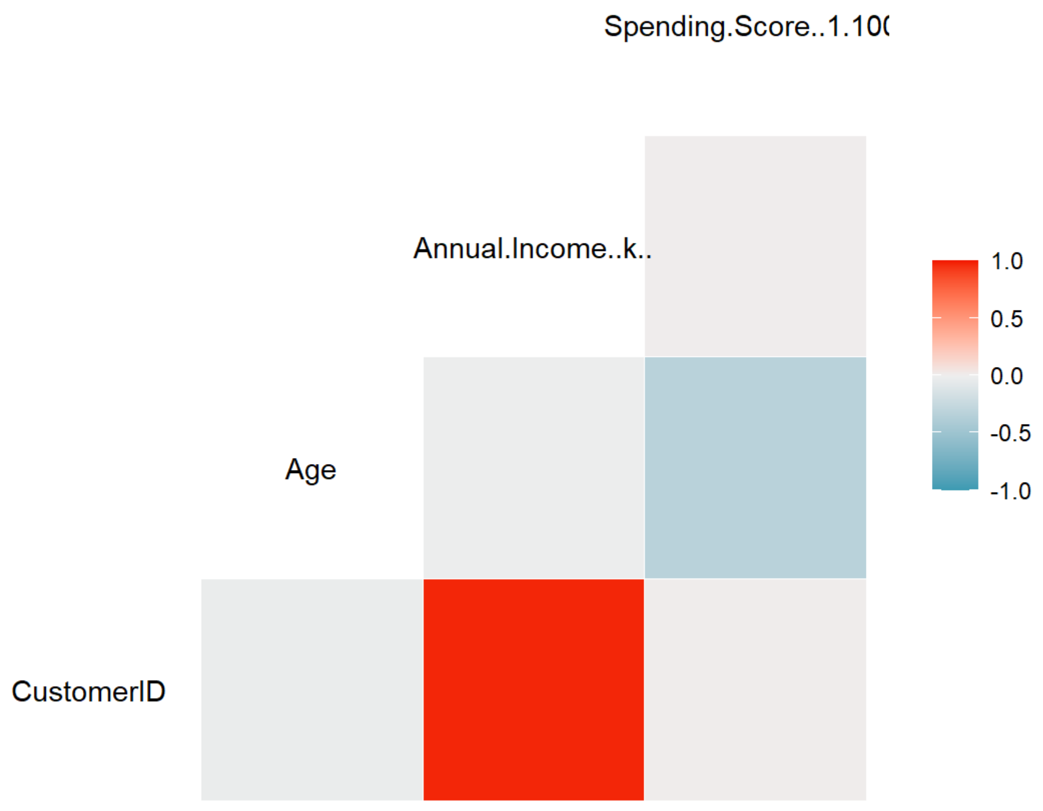
```
# Load GGally Library

library(GGally)
```

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

```
# The GGally library in R is an extension to the popular ggplot2 package and is designed for
exploring and visualizing relationships in multivariate datasets. It provides a set of functi
ons and tools for creating scatterplot matrices, correlation plots, and other types of visual
izations to help analyze the patterns and associations between variables.
```

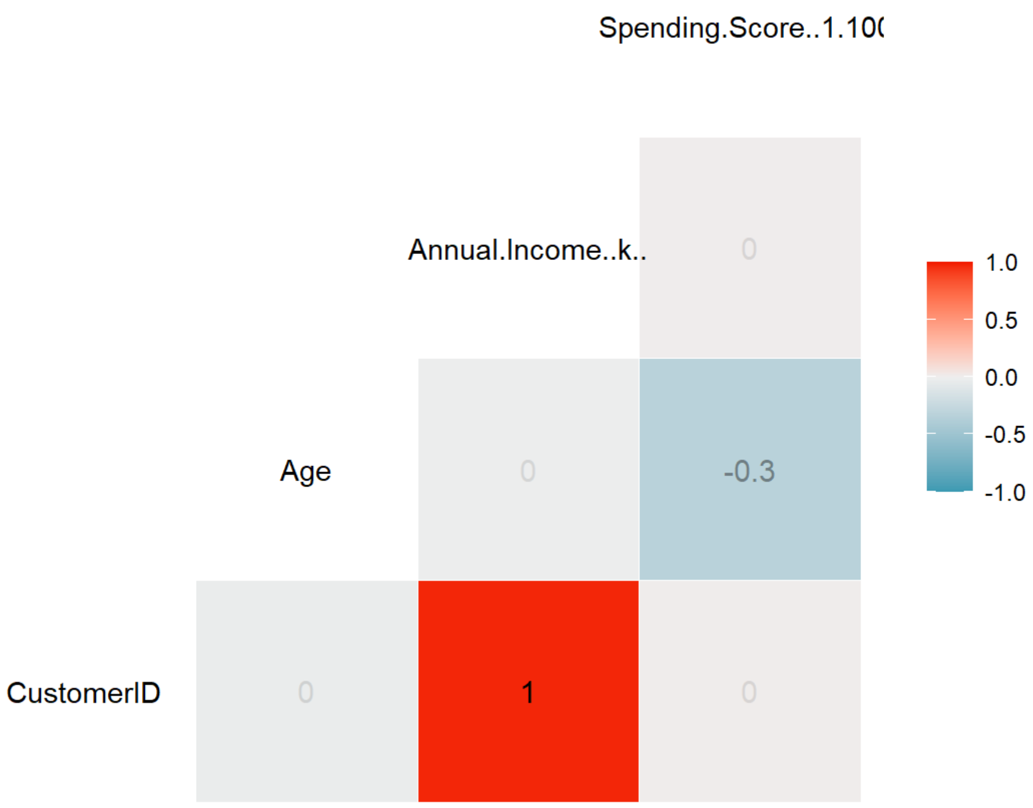
```
# ggcorr() function generates a visual representation of the correlation matrix, allowing for
a quick assessment of variable relationships.

ggcorr(customer_data)
```

```
## Warning in ggcorr(customer_data): data in column(s) 'Gender' are not numeric
## and were ignored
```



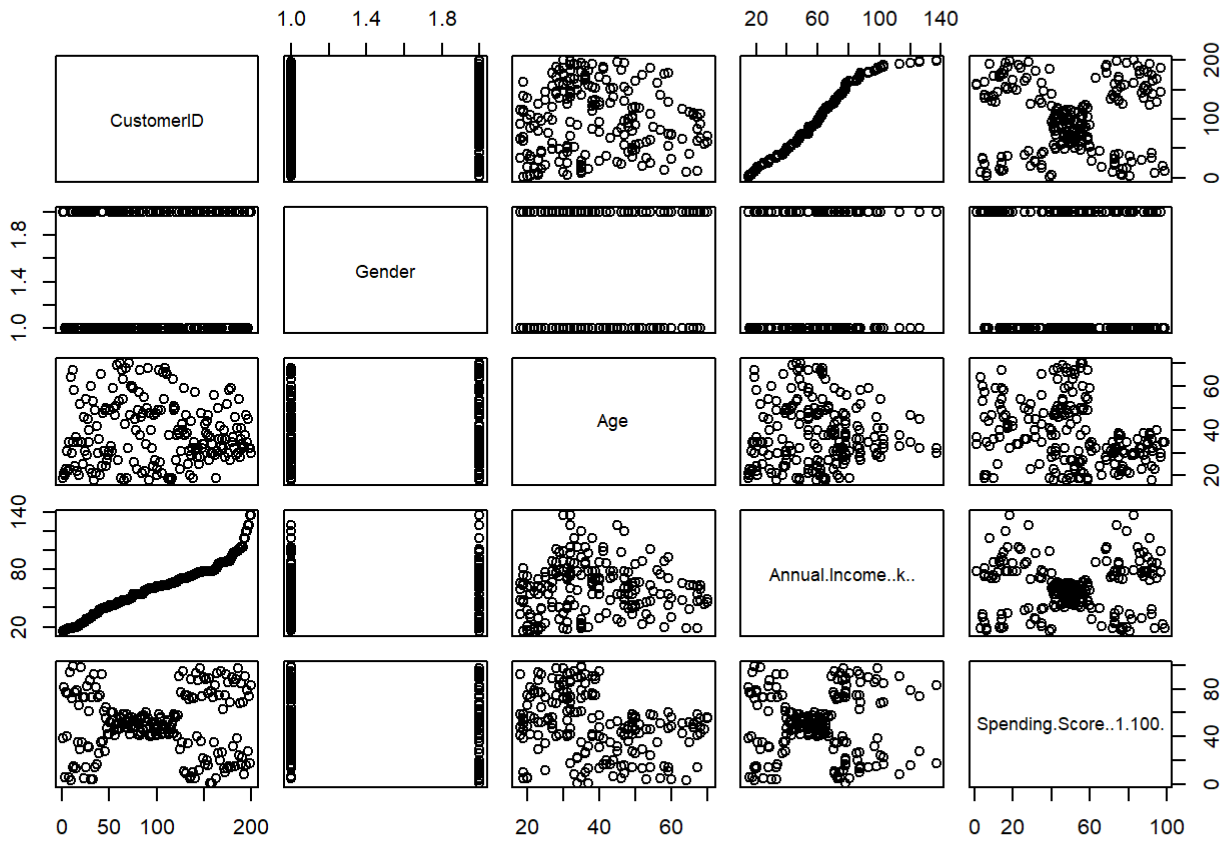
```
ggcorr(customer_data, label = TRUE, label_alpha = TRUE)
```

```
## Warning in ggcorr(customer_data, label = TRUE, label_alpha = TRUE): data in
## column(s) 'Gender' are not numeric and were ignored
```



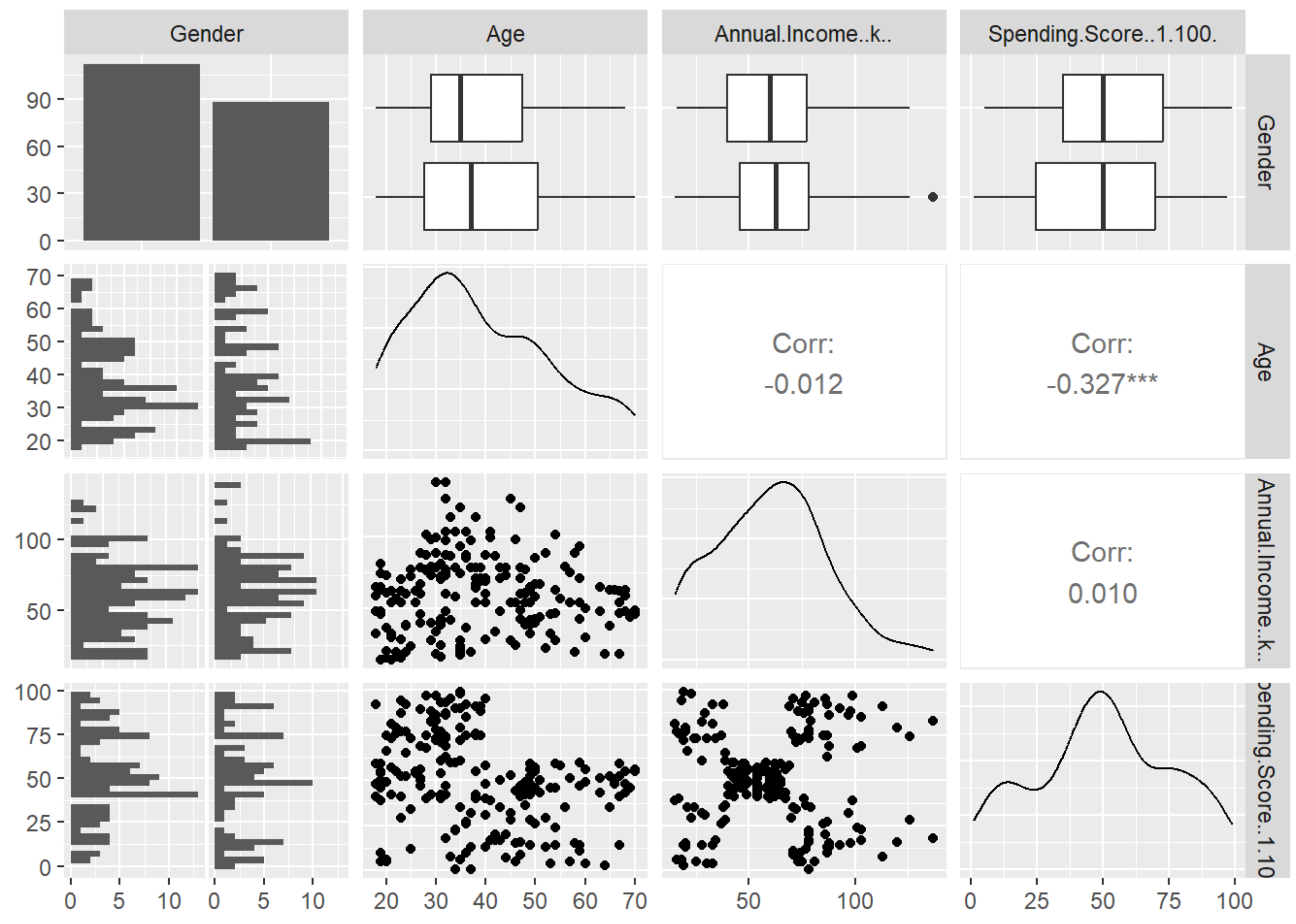
```
# label_alpha :- Display significance level (alpha) on the plot
# label :- Display Correlation coefficients on the plot
```

```
#par sets up a plotting layout of 2 rows and 2 columns.
#plot Generates scatterplots for each pair of variables in the customer_data dataset

par(mfrow=c(2,2))
plot(customer_data)
```



```
X <- customer_data[,2:5]
ggpairs(X)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# ggpairs() function creates a matrix of scatterplots for exploring relationships between mul
tiple variables. It can include scatterplots, density plots, and correlation coefficients.
```

Let’s apply an unsupervised ML algorithm on the dataset.

K-Means Algorithm

```
# The objective of K-means is to partition the data into 'K' clusters based on similarity, wh
ere 'K' is a predefined number.

# The algorithm iteratively assigns data points to clusters and updates the cluster centroids
until convergence.

# Unlike supervised learning, where the algorithm is trained on a labeled dataset to predict
output labels, K-means does not require labeled data. It identifies patterns solely based on
the similarity of data points.

# K-means is commonly used for clustering analysis, where the goal is to group similar data p
oints into clusters, making it a valuable tool for exploratory data analysis and pattern disc
overy in unsupervised scenarios.
```

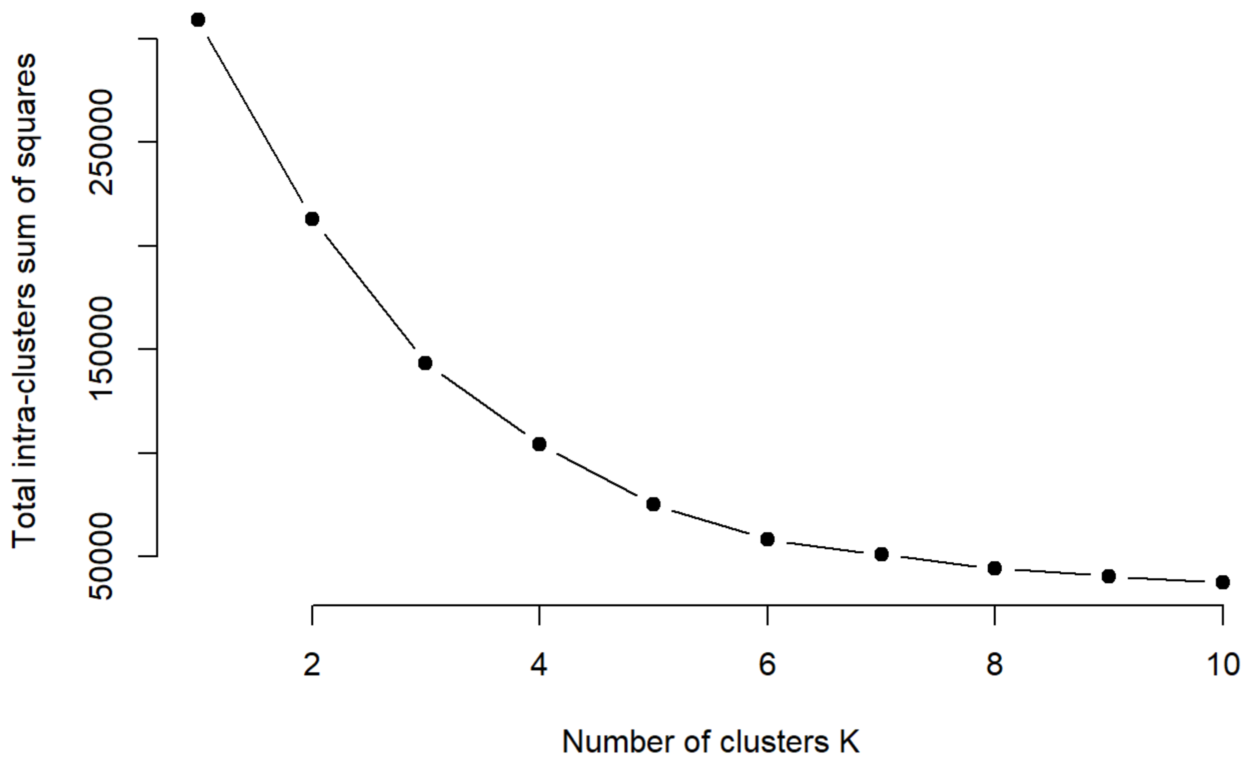
The main goal behind cluster partitioning methods like k-means is to define the clusters such that the intra-cluster variation stays minimum.

Minimize (sum W(Ck)), k=1…k

Where Ck represents the kth cluster and W(Ck) denotes the intra-cluster variation (this is the distance between data points in the same cluster). With the measurement of the total intra-cluster variation, one can evaluate the compactness of the clustering boundary. We can then proceed to define the optimal clusters as follows –

1. Calculate the clustering algorithm for several values of k. This can be done by creating a variation within k from 1 to 10 clusters.
2. Calculate the total intra-cluster sum of square (iss).
3. Plot iss based on the number of k clusters. This plot denotes the appropriate number of clusters required in our model.
4. In the plot, the location of a bend or a knee is the indication of the optimum number of clusters

```
library(purrr)

# purrr a package in R that provides a set of tools for functional programming, specifically
designed to work seamlessly with the principles of the tidyverse. The purrr package makes it
easier to work with and manipulate data in a functional programming paradigm.

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.value<-1:10
iss_value<- map_dbl(k.value, iss)
plot(k.value, iss_value,type="b", pch = 19, frame = FALSE,xlab="Number of clusters K",ylab="T
otal intra-clusters sum of squares")
```



Average Silhouette Method

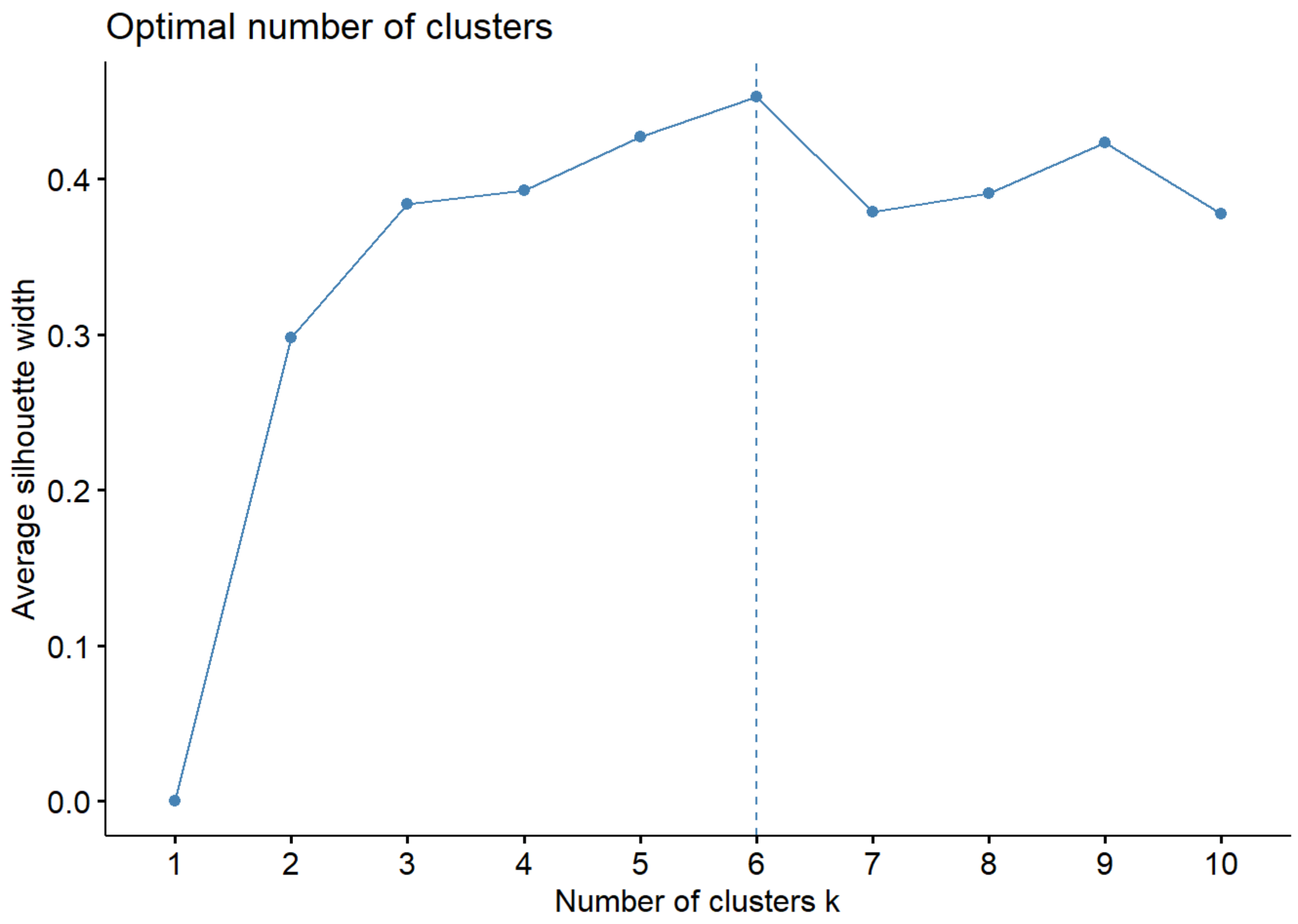
With the help of the average silhouette method, we can measure the quality of our clustering operation. With this, we can determine how well within the cluster is the data object. If we obtain a high average silhouette width, it means that we have good clustering. The average silhouette method calculates the mean of silhouette observations for different k values. With the optimal number of k clusters, one can maximize the average silhouette over significant values for k clusters.

Using the average silhouette widths (Distance between one cluster and the next clusters) for every k from 1 to 10, and plotted an optimal number of clusters to find the k with the highest average width.

```
library(NbClust)

# The NbClust package in R is designed for determining the optimal number of clusters in a da
taset. It provides a comprehensive set of indices and statistical tests for assessing the num
ber of clusters in a clustering analysis. The package is particularly useful when the user is
unsure about the appropriate number of clusters and wants to choose the best one based on var
ious criteria.

# The factoextra package in R is an extension to the popular factoMineR package and is design
ed to provide a set of functions for extracting and visualizing the results of multivariate d
ata analyses, particularly those obtained from factor analysis and clustering methods.

library(factoextra)
```

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

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



We can observe that the optimal number of clusters is 6. It’s time to plot those 6 clusters

Plotting of 6 clusters

We use PCA i.e, Principal Component Analysis

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

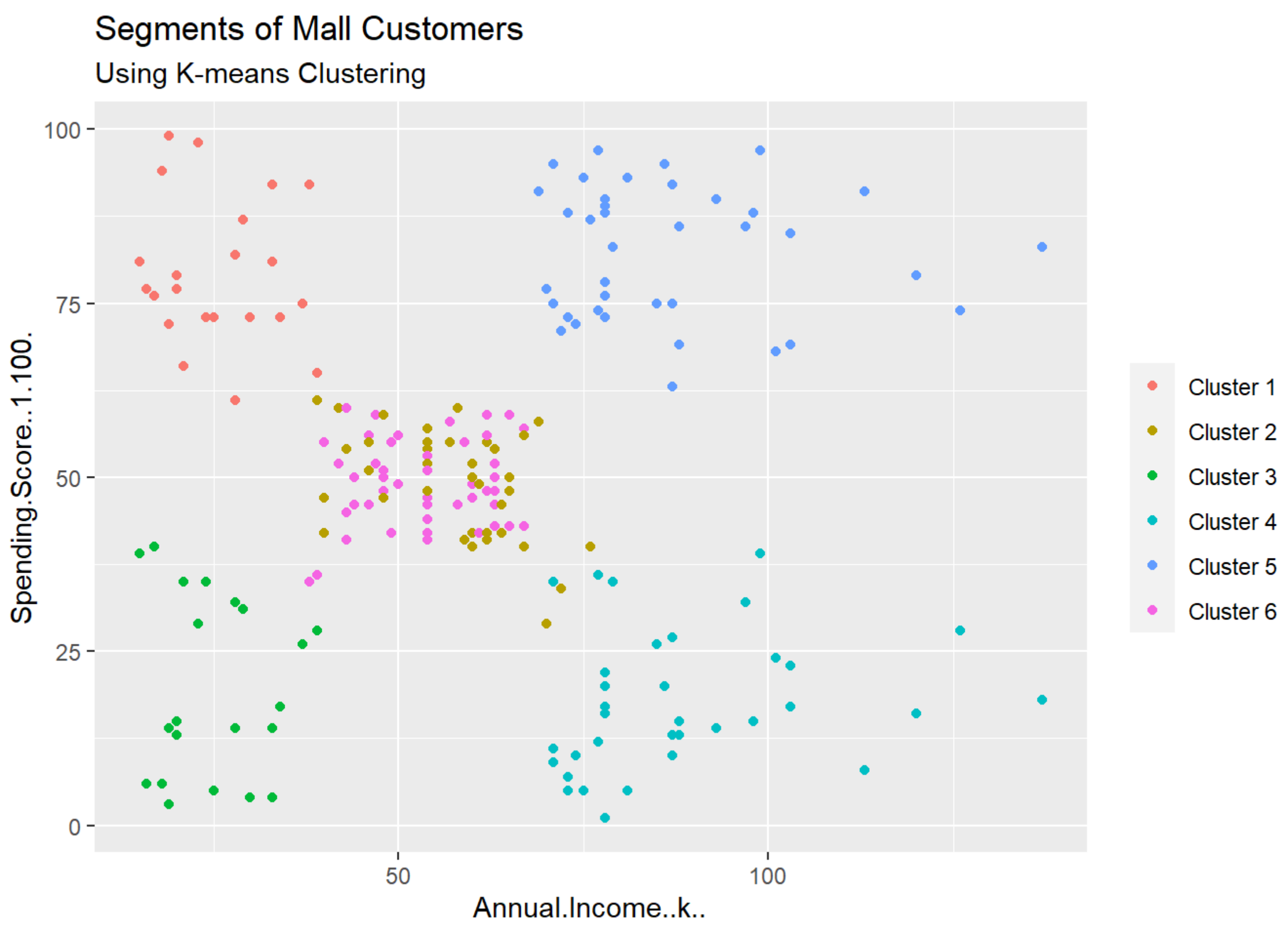
```
## Importance of components:
##                            PC1     PC2     PC3
## Standard deviation     26.4625 26.1597 12.9317
## Proportion of Variance  0.4512  0.4410  0.1078
## Cumulative Proportion   0.4512  0.8922  1.0000
```

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

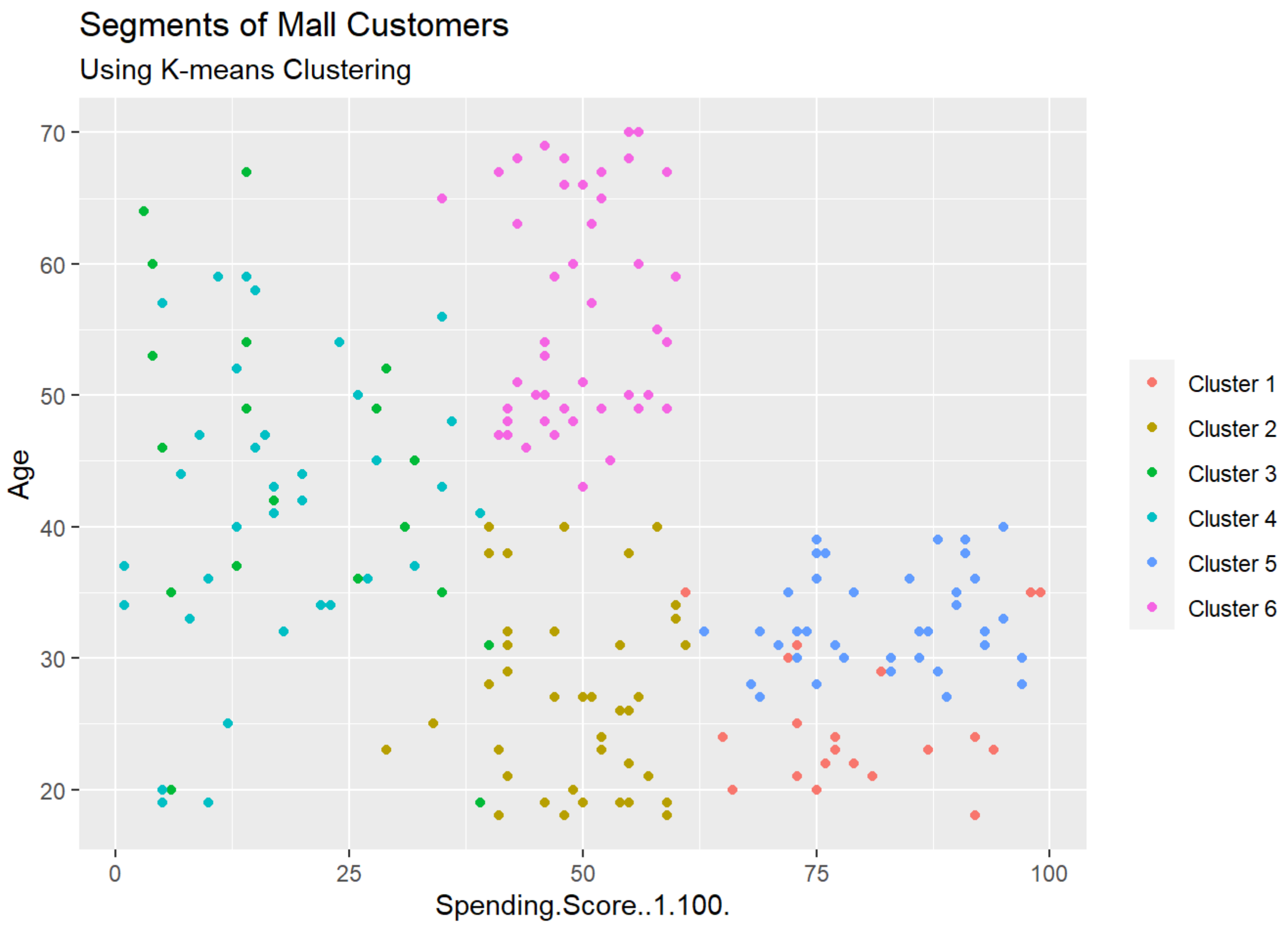
```
##                               PC1        PC2
## Age                     0.1889742 -0.1309652
## Annual.Income..k..     -0.5886410 -0.8083757
## Spending.Score..1.100. -0.7859965  0.5739136
```

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

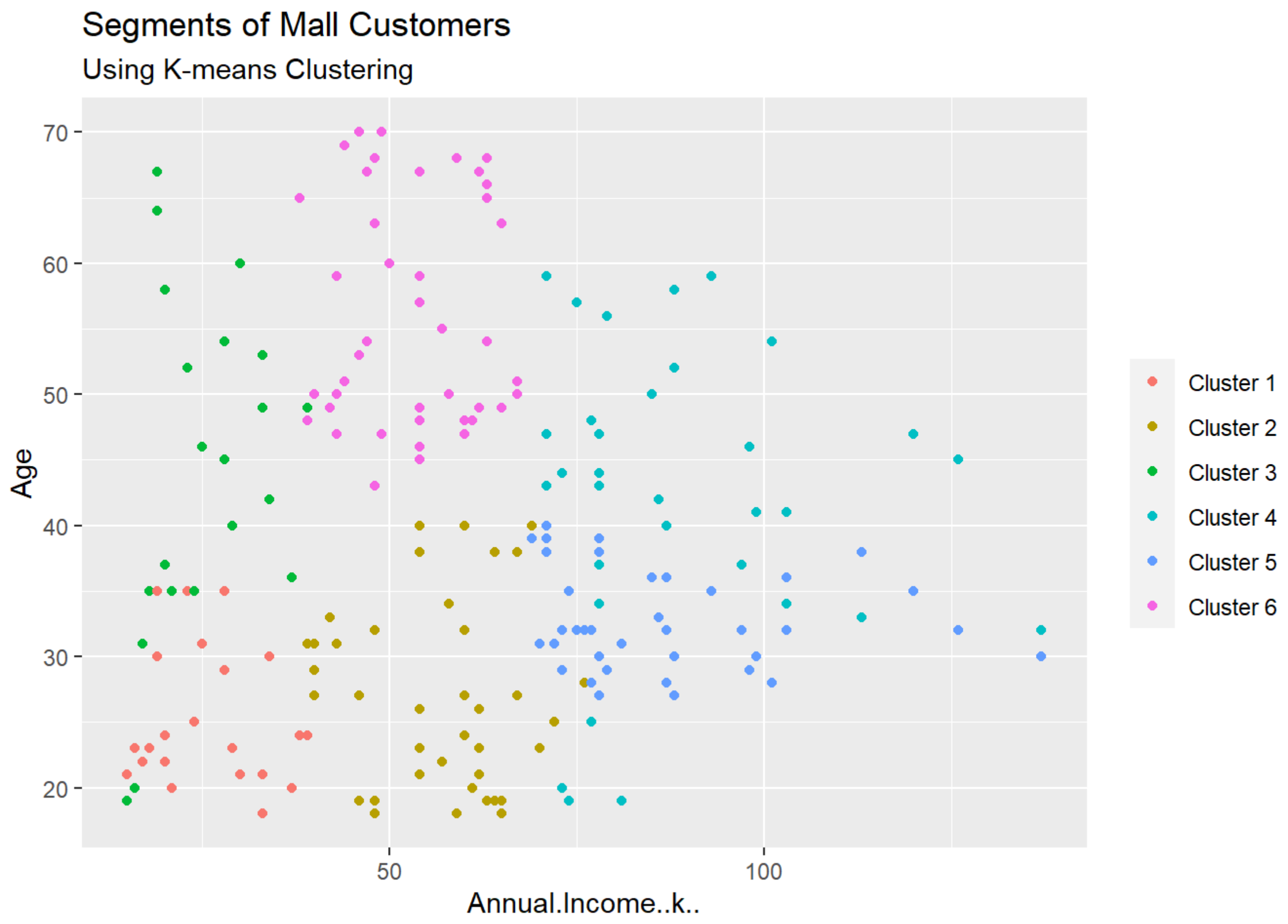
```r
# annual income vs spending score clusters
library(ggplot2)
ggplot(customer_data, aes(x =`Annual.Income..k..`, y = `Spending.Score..1.100.`)) +
  geom_point(stat = "identity", aes(color = as.factor(k6$cluster))) +
  scale_color_discrete(name=" ",breaks=c("1", "2", "3", "4", "5","6"), labels=c("Cluster 1",
"Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5","Cluster 6")) +
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
```



```r
#spending score vs age clusters
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 6")) +
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
```



```
#annual income vs age clusters
ggplot(customer_data, aes(x =`Annual.Income..k..`, 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 6")) +
  ggtitle("Segments of Mall Customers", subtitle = "Using K-means Clustering")
```



Analysis

Cluster 1 – This cluster represents the customers having a high annual income as well as a high annual spend with an age between age 28 and 40. Cluster 2 - This clusters represents customers with a average annual income(above 25 below 60), average spending score(above 25 and below 60) and ages above 40 years. Cluster 3 – This cluster denotes the customers with low annual income as well as low yearly spend of income with ages spreading through out the spectrum. Cluster 4 – This cluster denotes a high annual income and low yearly spending score with with ages spreading through out the spectrum. Cluster 5 – This cluster represents a low annual income but its high yearly expenditure. Cluster 6 - This clusters represents customers with a average annual income(above 30 below 80), average spending score(above 25 and below 62) and ages of 40 and below.

Mapping clusters back to the data to see which customer belongs to which cluster

Let’s Map all the 6 Clusters to the data set.

```
# In R, the order() function is used to obtain the order or permutation of the indices that w
ould sort a vector or a set of vectors. This function is often used to sort data or reorganiz
e it based on specific criteria.

o = order(k6$cluster)

# Using only customer ID for easy tracking instead of the whole dataset.
data.frame(customer_data$CustomerID[o],k6$cluster[o])
```

```
##     customer_data.CustomerID.o. k6.cluster.o.
## 1                             2             1
## 2                             4             1
## 3                             6             1
## 4                             8             1
## 5                            10             1
## 6                            12             1
## 7                            14             1
## 8                            16             1
## 9                            18             1
## 10                           20             1
## 11                           22             1
## 12                           24             1
## 13                           26             1
## 14                           28             1
## 15                           30             1
## 16                           32             1
## 17                           34             1
## 18                           36             1
## 19                           38             1
## 20                           40             1
## 21                           42             1
## 22                           46             1
## 23                           44             2
## 24                           48             2
## 25                           49             2
## 26                           50             2
## 27                           52             2
## 28                           53             2
## 29                           59             2
## 30                           62             2
## 31                           66             2
## 32                           69             2
## 33                           70             2
## 34                           76             2
## 35                           78             2
## 36                           79             2
## 37                           82             2
## 38                           85             2
## 39                           88             2
## 40                           89             2
## 41                           92             2
## 42                           94             2
## 43                           95             2
## 44                           96             2
## 45                           98             2
## 46                          100             2
## 47                          101             2
## 48                          104             2
## 49                          106             2
## 50                          112             2
## 51                          113             2
## 52                          114             2
## 53                          115             2
## 54                          116             2
## 55                       121            2
## 56                       122            2
## 57                       123            2
## 58                       125            2
## 59                       133            2
## 60                       143            2
## 61                         1            3
## 62                         3            3
## 63                         5            3
## 64                         7            3
## 65                         9            3
## 66                        11            3
## 67                        13            3
## 68                        15            3
## 69                        17            3
## 70                        19            3
## 71                        21            3
## 72                        23            3
## 73                        25            3
## 74                        27            3
## 75                        29            3
## 76                        31            3
## 77                        33            3
## 78                        35            3
## 79                        37            3
## 80                        39            3
## 81                        45            3
## 82                       127            4
## 83                       129            4
## 84                       131            4
## 85                       135            4
## 86                       137            4
## 87                       139            4
## 88                       141            4
## 89                       145            4
## 90                       147            4
## 91                       149            4
## 92                       151            4
## 93                       153            4
## 94                       155            4
## 95                       157            4
## 96                       159            4
## 97                       161            4
## 98                       163            4
## 99                       165            4
## 100                      167            4
## 101                      169            4
## 102                      171            4
## 103                      173            4
## 104                      175            4
## 105                      177            4
## 106                      179            4
## 107                      181            4
## 108                      183            4
## 109                      185            4
## 110                      187            4
## 111                         189          4
## 112                         191          4
## 113                         193          4
## 114                         195          4
## 115                         197          4
## 116                         199          4
## 117                         124          5
## 118                         126          5
## 119                         128          5
## 120                         130          5
## 121                         132          5
## 122                         134          5
## 123                         136          5
## 124                         138          5
## 125                         140          5
## 126                         142          5
## 127                         144          5
## 128                         146          5
## 129                         148          5
## 130                         150          5
## 131                         152          5
## 132                         154          5
## 133                         156          5
## 134                         158          5
## 135                         160          5
## 136                         162          5
## 137                         164          5
## 138                         166          5
## 139                         168          5
## 140                         170          5
## 141                         172          5
## 142                         174          5
## 143                         176          5
## 144                         178          5
## 145                         180          5
## 146                         182          5
## 147                         184          5
## 148                         186          5
## 149                         188          5
## 150                         190          5
## 151                         192          5
## 152                         194          5
## 153                         196          5
## 154                         198          5
## 155                         200          5
## 156                          41          6
## 157                          43          6
## 158                          47          6
## 159                          51          6
## 160                          54          6
## 161                          55          6
## 162                          56          6
## 163                          57          6
## 164                          58          6
## 165                          60          6
## 166                          61          6
## 167                          63            6
## 168                          64            6
## 169                          65            6
## 170                          67            6
## 171                          68            6
## 172                          71            6
## 173                          72            6
## 174                          73            6
## 175                          74            6
## 176                          75            6
## 177                          77            6
## 178                          80            6
## 179                          81            6
## 180                          83            6
## 181                          84            6
## 182                          86            6
## 183                          87            6
## 184                          90            6
## 185                          91            6
## 186                          93            6
## 187                          97            6
## 188                          99            6
## 189                         102            6
## 190                         103            6
## 191                         105            6
## 192                         107            6
## 193                         108            6
## 194                         109            6
## 195                         110            6
## 196                         111            6
## 197                         117            6
## 198                         118            6
## 199                         119            6
## 200                         120            6
```

```
# mapping using all the other columns i.e, to the entire dataset

x = data.frame(customer_data$CustomerID[o], customer_data$Gender[o], customer_data$Age[o], cu
stomer_data$`Annual.Income..k..`[o], customer_data$`Spending.Score..1.100.`[o],k6$cluster[o])
```

Conclusion

1. The dataset contains information about customers, including gender, age, annual income, and spending score.
2. No missing values were found in the dataset, ensuring completeness for analysis.
3. Summary statistics and visualizations were generated for key variables like age, annual income, and spending score.
4. Histograms and box plots provided insights into the distribution and central tendencies of these variables.
5. Bar plots and pie charts were used to visualize the gender distribution among customers.
6. It was observed that there were more female customers compared to male customers.
7. The K-means algorithm was applied to cluster customers into segments based on annual income, spending score, and age.

8. The silhouette method and the elbow method were used to determine the optimal number of clusters, which was found to be 6.
9. Visualization of clusters using scatter plots revealed distinct segments with different characteristics.
10. Each cluster was analyzed to understand the characteristics of customers within them.
11. Clusters were differentiated based on factors such as high/low annual income, high/low spending score, and age groups.
12. Interpretation of clusters allowed for targeted marketing strategies and personalized customer engagement.
13. Customer segmentation is crucial for tailoring marketing strategies to specific customer groups.
14. The identified clusters provide a basis for targeted advertising, promotions, and services.
15. Understanding customer segments can lead to improved customer satisfaction and increased business revenue.