shopping_eda.R

Andrew Ruiz

2024-06-26

```r
# Importing necessary libraries for the project
library(tidyverse)  # For data manipulation and visualization

## ── 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   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors

library(psych)      # For descriptive statistics

##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

library(ggplot2)    # For data visualization
library(corrplot)   # For correlation plot

## corrplot 0.92 loaded

library(ggcorrplot) # For enhanced correlation plot
library(cluster)    # For clustering

# Function to load and inspect data
load_and_inspect_data <- function(file_path) {
  data <- read.csv(file_path)
  print("Data Summary:")
  print(summary(data))
  print("Data Structure:")
  print(str(data))
  print("Descriptive Statistics:")
  print(describe(data))
```

```r
  print("Data Head:")
  print(head(data))
  missing_values <- colSums(is.na(data))
  print("Missing Values:")
  print(missing_values)
  return(data)
}

# Function to clean data
clean_data <- function(data) {
  data <- na.omit(data)
  print("Cleaned Data Summary:")
  print(summary(data))
  return(data)
}

# Function to transform data
transform_data <- function(data) {
  names(data) <- c('ID', 'Gender', 'Age', 'Income', 'Spending_Score')
  data$Gender <- ifelse(data$Gender == 'Male', 1, ifelse(data$Gender ==
'Female', 0, NA))
  data$Rating <- cut(data$Spending_Score, breaks = c(0, 35, 70, 100), labels
= c("Bad", "Normal", "Good"))
  data$Rating <- as.factor(data$Rating)
  data$Gender <- as.factor(data$Gender)  # Convert Gender to factor for
plotting
  levels(data$Gender) <- c("Female", "Male")  # Set levels to meaningful
names
  print("Transformed Data Head:")
  print(head(data))
  return(data)
}

# Function for correlation analysis
correlation_analysis <- function(data) {
  cor_matrix <- cor(data %>% select(Age, Income, Spending_Score))
  print("Correlation Matrix:")
  print(cor_matrix)
  corrplot(cor_matrix, method = 'circle')
}

# Function to create histograms
create_histograms <- function(data) {
  cols <- c('Age', 'Income', 'Spending_Score')
  for (i in cols) {
    print(ggplot(data, aes_string(x = i)) +
            geom_histogram(fill = 'darkblue', binwidth = 2) +
            xlab(i) + theme_bw() + ggtitle(paste0(i, ' Histogram')))
  }
```

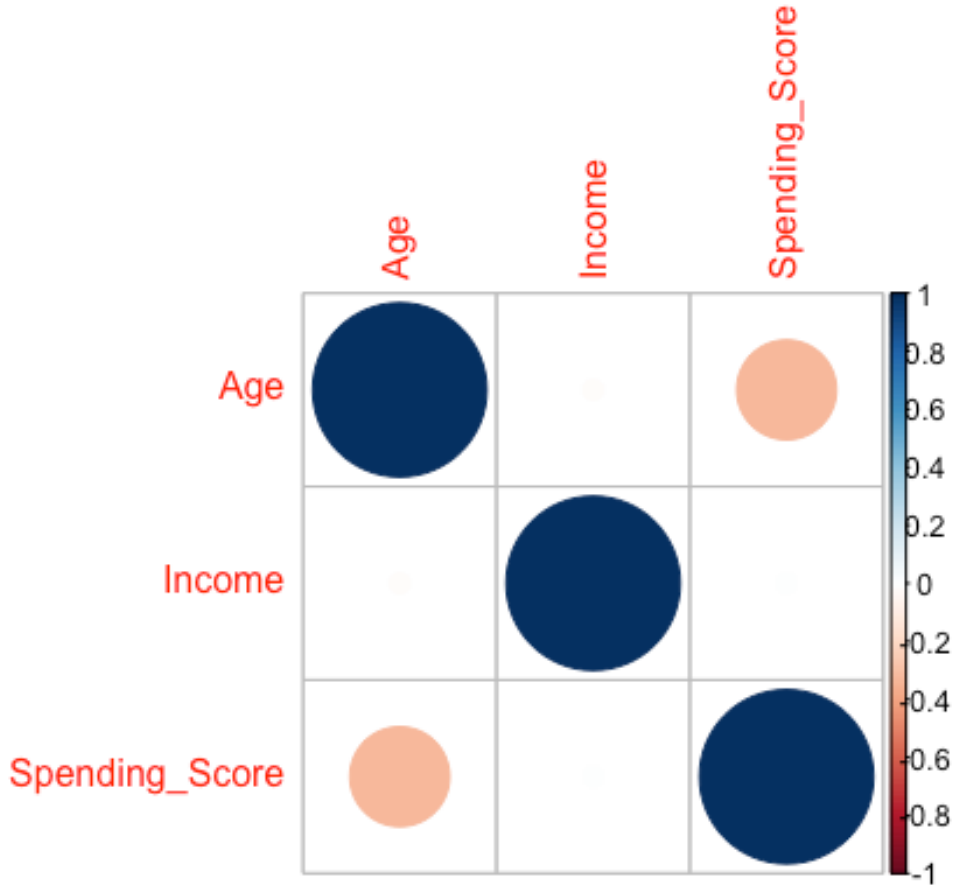
```r
}

# Function to create boxplots
create_boxplots <- function(data, by_var) {
  cols <- c('Age', 'Income', 'Spending_Score')
  for (i in cols) {
    print(ggplot(data, aes_string(x = by_var, y = i, fill = by_var)) +
            geom_boxplot() + ylab(i) + theme_bw() + ggtitle(paste0(i, '
Boxplot by ', by_var)))
  }
}

# Function to calculate and plot averages
calculate_and_plot_averages <- function(data, group_var, value_var,
fill_color, plot_title) {
  avg_data <- data %>% group_by_at(group_var) %>% summarize(N = n(),
avg_value = mean(get(value_var), na.rm = TRUE))
  print(paste0("Average ", value_var, " by ", group_var, ":"))
  print(avg_data)
  ggplot(avg_data, aes_string(x = group_var, y = 'avg_value')) +
    geom_col(fill = fill_color) + theme_bw() + ggtitle(plot_title)
}

# Function to create scatter plots
create_scatter_plots <- function(data, x_var, y_var, color_var, plot_title) {
  ggplot(data, aes_string(x = x_var, y = y_var, color = color_var)) +
    geom_point() + ggtitle(plot_title) + theme_bw()
}

# Function for k-means clustering
perform_kmeans_clustering <- function(data, num_clusters) {
  set.seed(123)  # Set seed for reproducibility
  data_scaled <- scale(data %>% select(Age, Income, Spending_Score))  # Scale
the data for clustering
  kmeans_result <- kmeans(data_scaled, centers = num_clusters)  # Apply k-
means clustering
  data$Cluster <- as.factor(kmeans_result$cluster)
  print("K-means Clustering Results:")
  print(kmeans_result)
  ggplot(data, aes(x = Income, y = Spending_Score, color = Cluster)) +
    geom_point() + ggtitle('Income vs Spending Score by Cluster') +
theme_bw()
}

# Main function to execute the analysis
main <- function() {
  file_path <- '/Users/andrew/Downloads/Shopping_data.csv'
  data <- load_and_inspect_data(file_path)
  data <- clean_data(data)
```

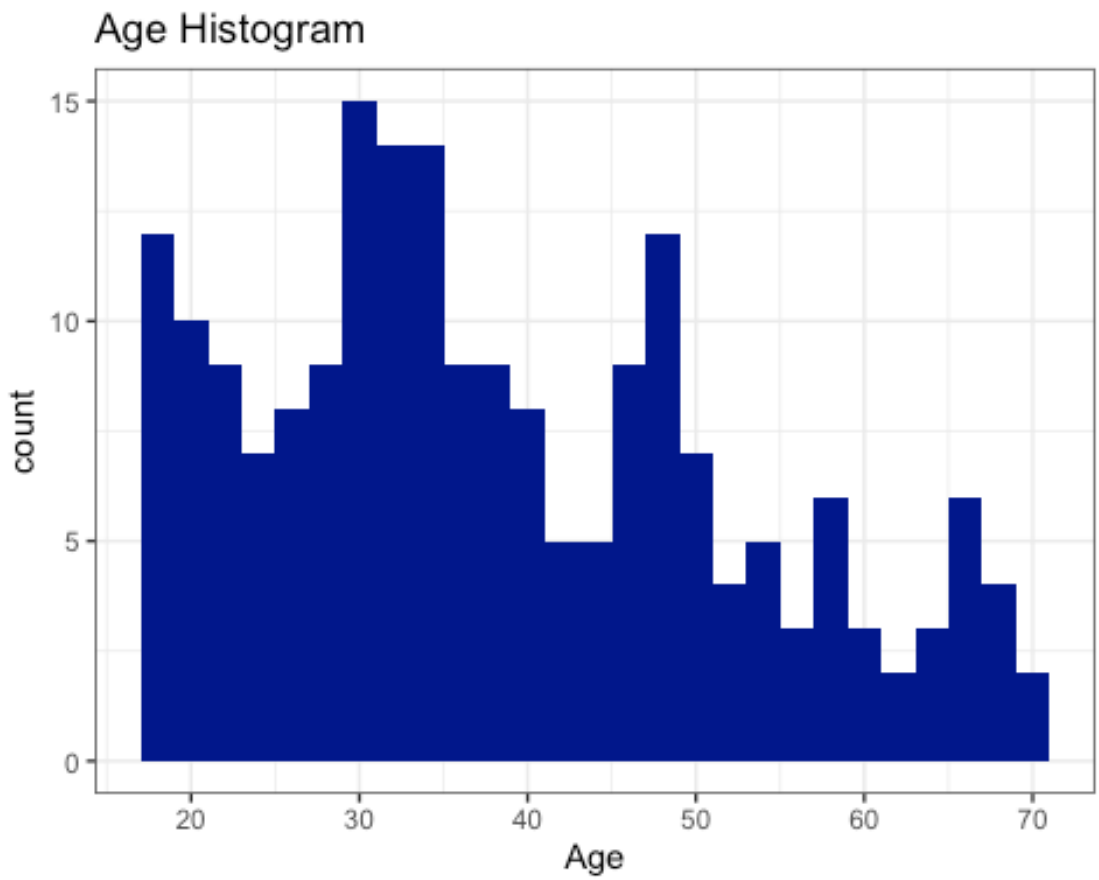
```r
  data <- transform_data(data)

  correlation_analysis(data)
  create_histograms(data)
  create_boxplots(data, 'Gender')
  create_boxplots(data, 'Rating')

  calculate_and_plot_averages(data, 'Gender', 'Income', 'darkorange',
'Average Income by Gender')
  calculate_and_plot_averages(data, 'Gender', 'Spending_Score', 'darkred',
'Average Spending Score by Gender')
  calculate_and_plot_averages(data, 'Rating', 'Income', 'darkgreen', 'Average
Income by Rating')
  calculate_and_plot_averages(data, 'Rating', 'Spending_Score', 'purple',
'Average Spending Score by Rating')

  create_scatter_plots(data, 'Age', 'Income', 'Gender', 'Age vs Income by
Gender')
  create_scatter_plots(data, 'Income', 'Spending_Score', 'Gender', 'Income vs
Spending Score by Gender')
  create_scatter_plots(data, 'Age', 'Income', 'Rating', 'Age vs Income by
Rating')
  create_scatter_plots(data, 'Income', 'Spending_Score', 'Rating', 'Income vs
Spending Score by Rating')

  perform_kmeans_clustering(data, 3)
}

# Run the main function
main()
```

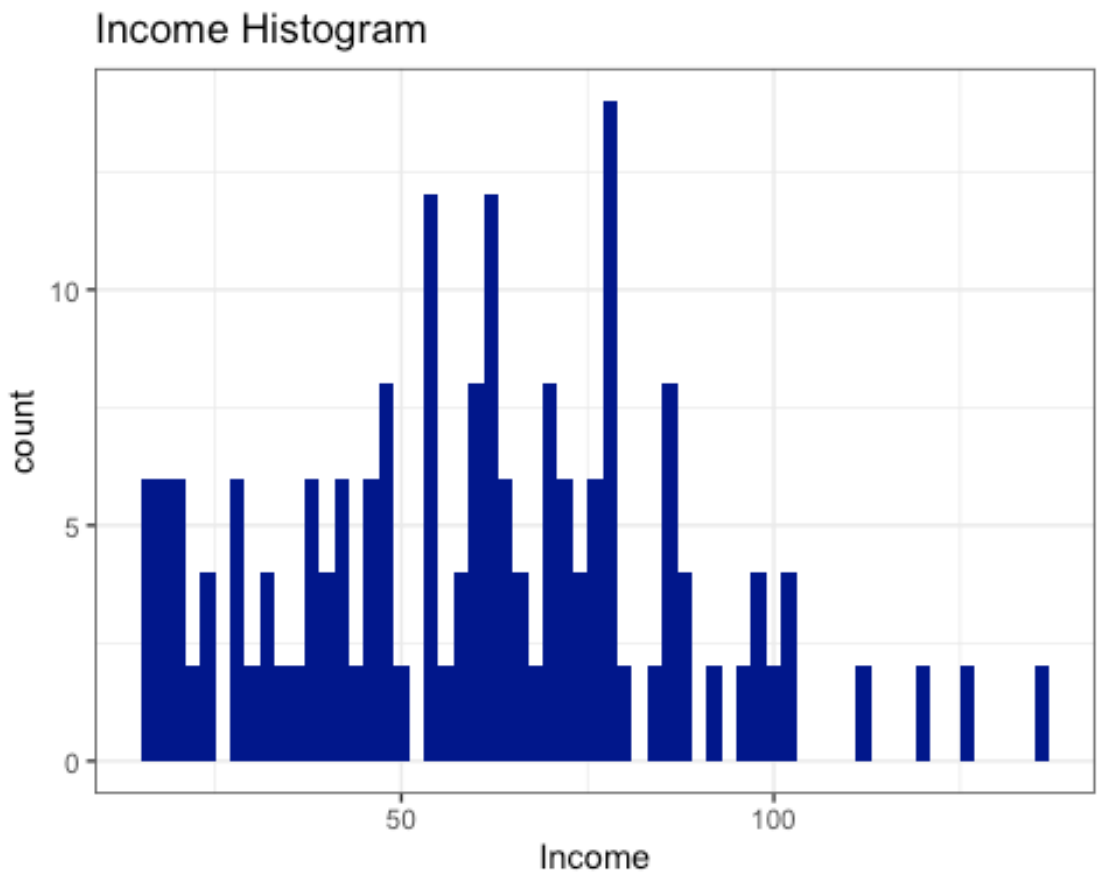
```
## [1] "Data Summary:"
##    CustomerID        Genre                Age        Annual.Income..k..
##  Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00
##  1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50
##  Median :100.50   Mode  :character   Median :36.00   Median : 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
## [1] "Data Structure:"
## 'data.frame':    200 obs. of  5 variables:
##  $ CustomerID            : int  1 2 3 4 5 6 7 8 9 10 ...
```

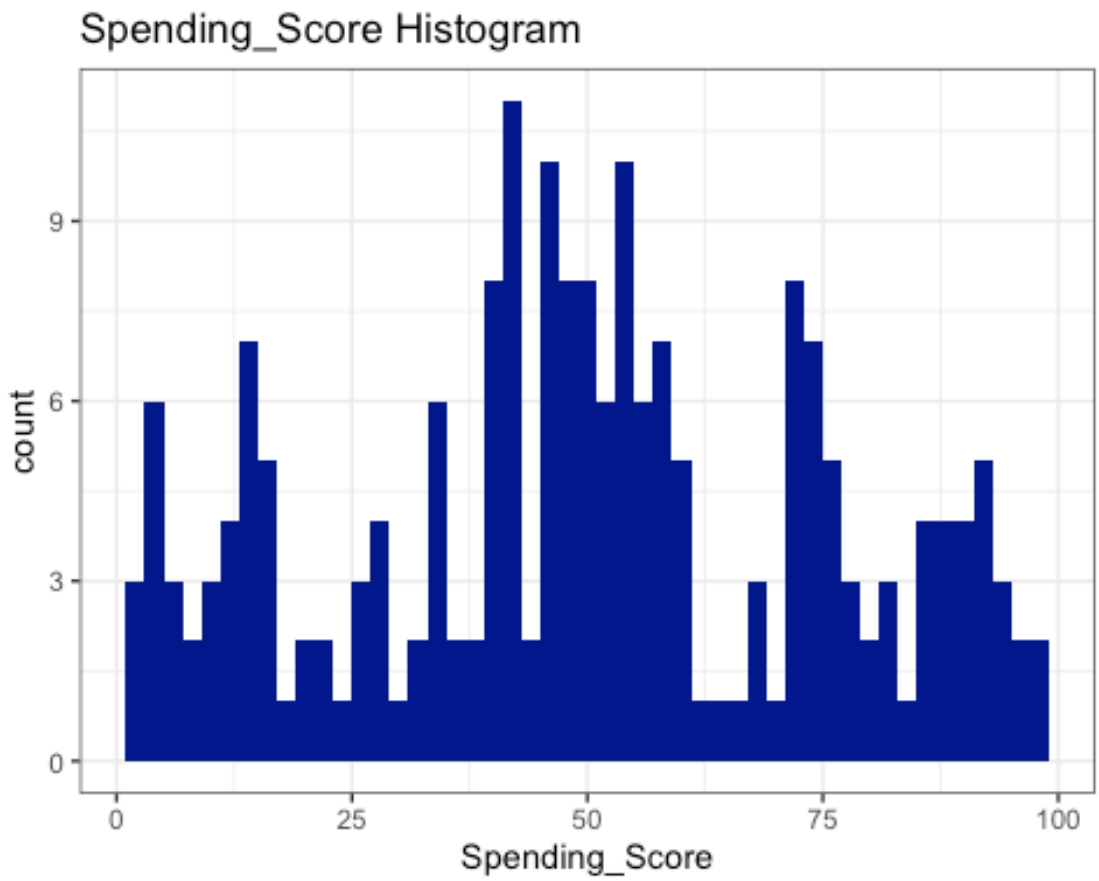
```
##  $ Genre                 : chr  "Male" "Male" "Female" "Female" ...
##  $ Age                   : int  19 21 20 23 31 22 35 23 64 30 ...
##  $ Annual.Income..k..    : int  15 15 16 16 17 17 18 18 19 19 ...
##  $ Spending.Score..1.100.: int  39 81 6 77 40 76 6 94 3 72 ...
## NULL
## [1] "Descriptive Statistics:"
##                        vars   n   mean    sd median trimmed   mad min max
range
## CustomerID                1 200 100.50 57.88  100.5  100.50 74.13   1 200
199
## Genre*                    2 200   1.44  0.50    1.0    1.42  0.00   1   2
1
## Age                       3 200  38.85 13.97   36.0   37.94 16.31  18  70
52
## Annual.Income..k..        4 200  60.56 26.26   61.5   59.64 24.46  15 137
122
## Spending.Score..1.100.    5 200  50.20 25.82   50.0   50.31 29.65   1  99
98
##                         skew kurtosis   se
## CustomerID              0.00    -1.22 4.09
## Genre*                  0.24    -1.95 0.04
## Age                     0.48    -0.71 0.99
## Annual.Income..k..      0.32    -0.15 1.86
## Spending.Score..1.100. -0.05    -0.86 1.83
## [1] "Data Head:"
##   CustomerID  Genre 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
## [1] "Missing Values:"
##             CustomerID                  Genre                    Age
##                      0                      0                      0
##     Annual.Income..k.. Spending.Score..1.100.
##                      0                      0
## [1] "Cleaned Data Summary:"
##    CustomerID        Genre                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
```

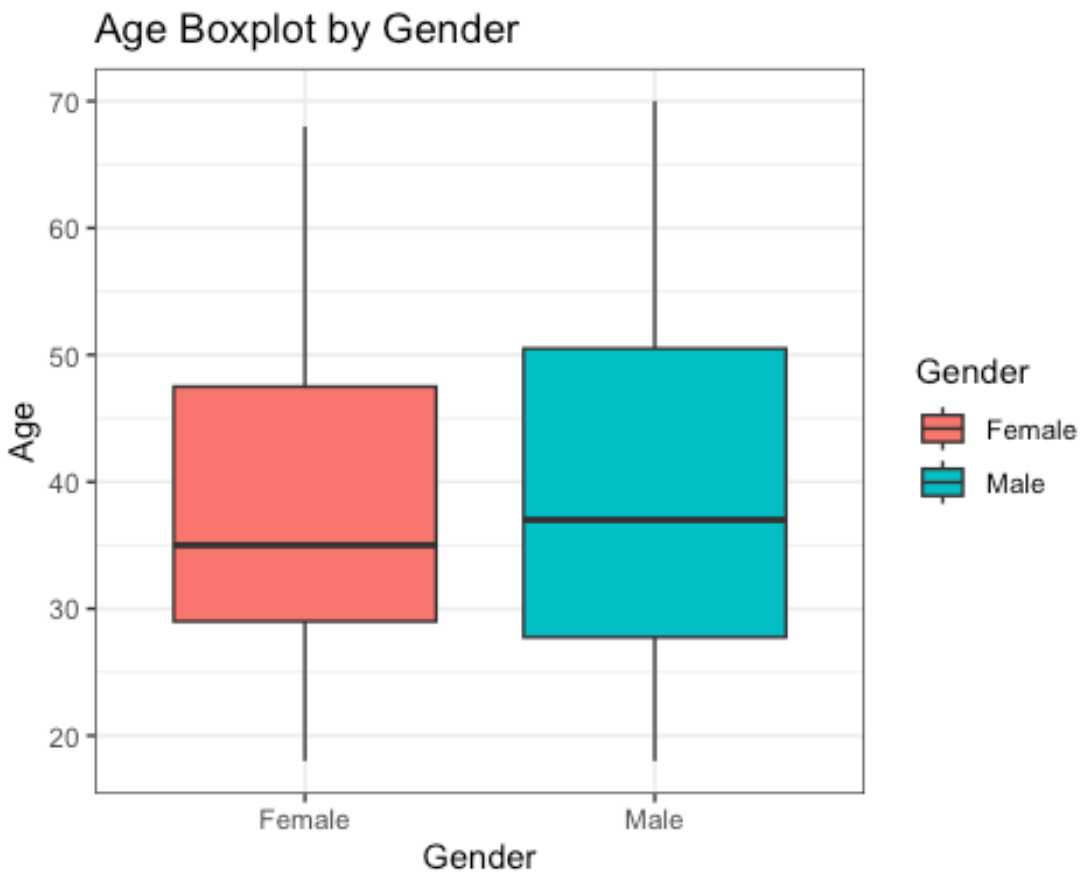
```
##  3rd Qu.:73.00
##  Max.   :99.00
## [1] "Transformed Data Head:"
##   ID Gender Age Income Spending_Score Rating
## 1  1   Male  19     15             39 Normal
## 2  2   Male  21     15             81   Good
## 3  3 Female  20     16              6    Bad
## 4  4 Female  23     16             77   Good
## 5  5 Female  31     17             40 Normal
## 6  6 Female  22     17             76   Good
## [1] "Correlation Matrix:"
##                        Age       Income Spending_Score
## Age             1.00000000 -0.012398043   -0.327226846
## Income         -0.01239804  1.000000000    0.009902848
## Spending_Score -0.32722685  0.009902848    1.000000000

## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Age
Income
Spending_Score
Age
Income
Spending_Score
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Age Histogram
count
15
10
5
0
20
30
40
50
60
70
Age

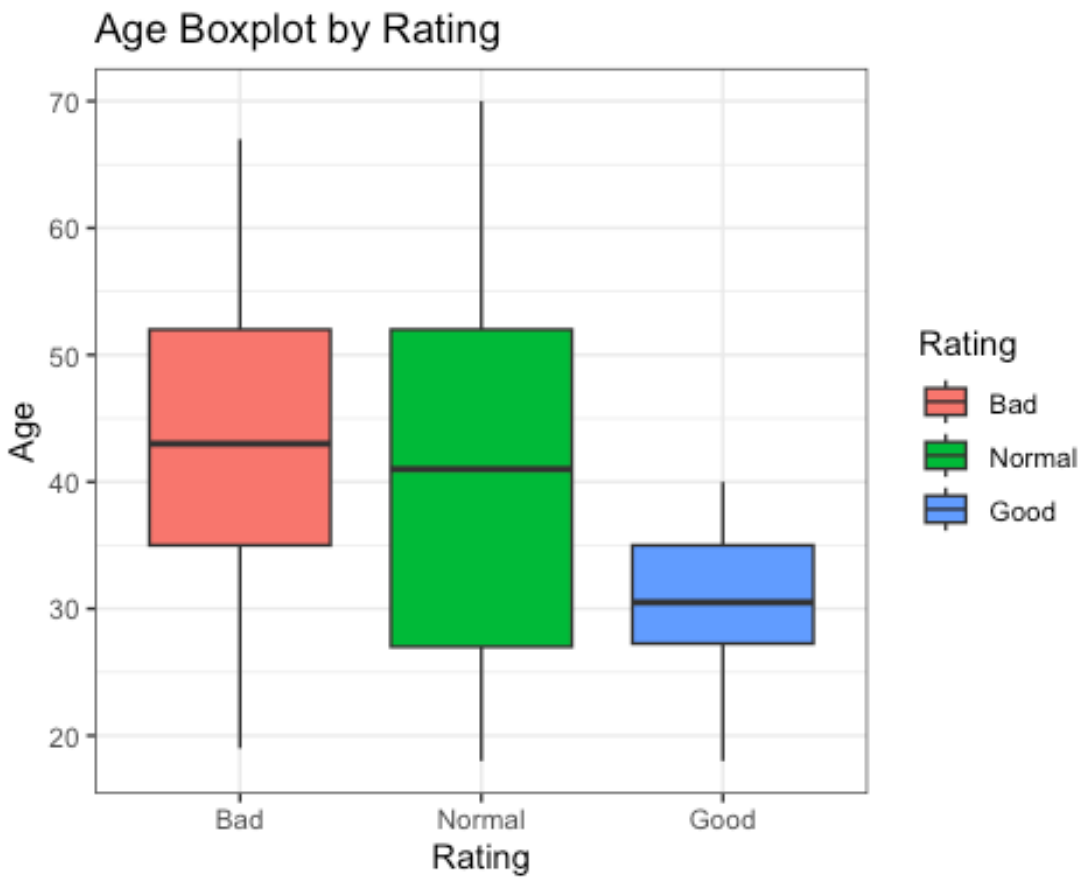

Income Histogram
count
10
5
0
50
100
Income

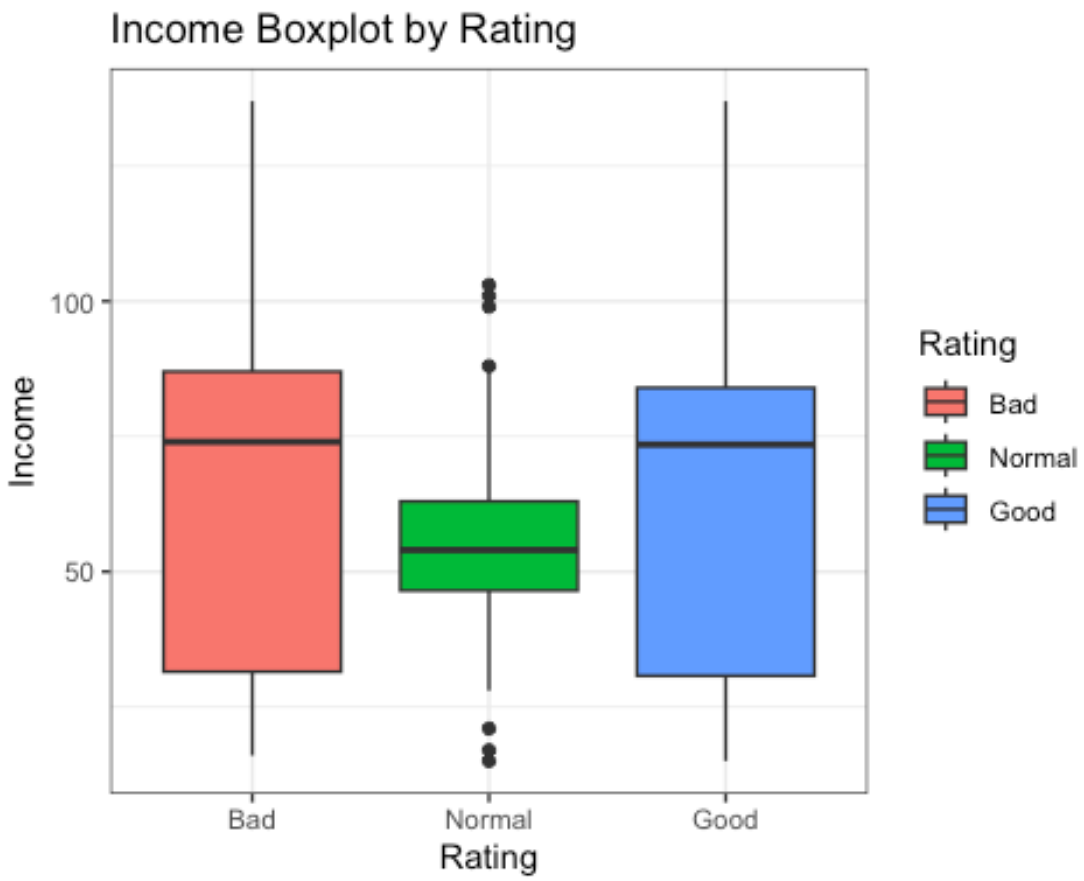

Spending_Score Histogram
count
9
6
3
0
0
25
50
75
100
Spending_Score

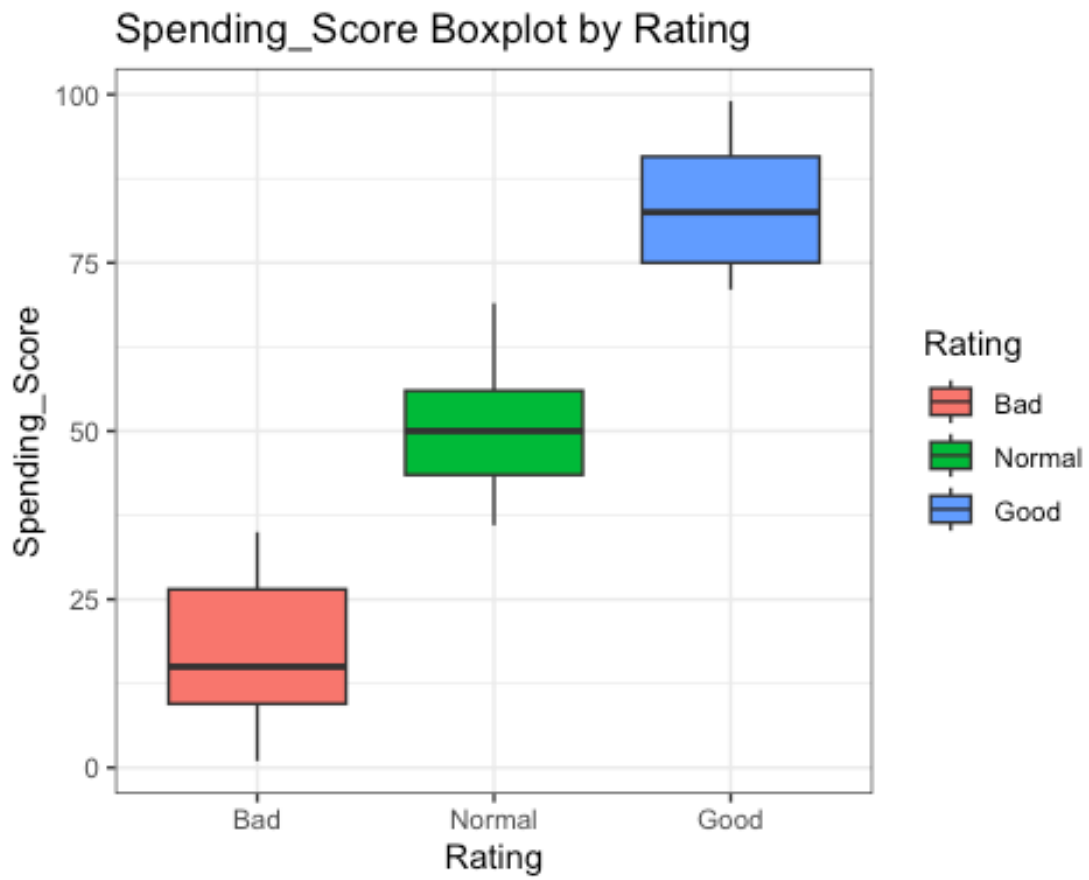

Age Boxplot by Gender
70
60
50
40
30
20
Age
Female
Male
Gender
Gender
Female
Male


Income Boxplot by Gender
100
50
Income
Female
Male
Gender
Gender
Female
Male


Spending_Score Boxplot by Gender
100
75
50
25
0
Spending_Score
Female
Male
Gender
Gender
Female
Male


Age Boxplot by Rating
70
60
50
40
30
20
Age
Bad
Normal
Good
Rating
Rating
Bad
Normal
Good


Income Boxplot by Rating
Income
100
50
Bad
Normal
Good
Rating
Rating
Bad
Normal
Good




```
## [1] "Average Income by Gender:"
## # A tibble: 2 × 3
##   Gender     N avg_value
##   <fct>  <int>     <dbl>
## 1 Female   112      59.2
## 2 Male      88      62.2
## [1] "Average Spending_Score by Gender:"
## # A tibble: 2 × 3
##   Gender     N avg_value
##   <fct>  <int>     <dbl>
## 1 Female   112      51.5
## 2 Male      88      48.5
## [1] "Average Income by Rating:"
## # A tibble: 3 × 3
##   Rating     N avg_value
##   <fct>  <int>     <dbl>
## 1 Bad       55      65.2
## 2 Normal    91      55.5
## 3 Good      54      64.3
## [1] "Average Spending_Score by Rating:"
## # A tibble: 3 × 3
##   Rating     N avg_value
##   <fct>  <int>     <dbl>
```

```
## 1 Bad       55      17.4
## 2 Normal    91      50.5
## 3 Good      54      83.1
## [1] "K-means Clustering Results:"
## K-means clustering with 3 clusters of sizes 38, 69, 93
##
## Cluster means:
##           Age       Income Spending_Score
## 1  0.03711223  0.987636603      -1.185781
## 2  0.98494441 -0.551596803      -0.420805
## 3 -0.74592934  0.005698801       0.796723
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
19  20
##   3   3   2   3   2   3   2   3   2   3   2   3   2   3   2   3   2   3
2   3
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
39  40
##   2   3   2   3   2   3   2   3   2   3   2   3   2   3   2   3   2   3
2   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58
59  60
##   2   3   2   3   2   3   2   3   3   3   2   3   3   2   2   2   2   2
3   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
79  80
##   2   3   2   2   2   3   2   2   3   3   2   2   2   2   2   3   2   2
3   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
99 100
##   2   3   2   2   3   2   2   3   3   2   2   3   2   2   3   3   2   3
2   3
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120
##   3   2   2   3   2   3   2   2   2   2   2   3   1   3   3   3   2   2
2   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
139 140
##   3   1   3   3   1   3   1   3   2   3   1   3   1   3   1   3   1   3
1   3
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158
159 160
##   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3
1   3
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178
179 180
##   2   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3
1   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
```

```
199 200
##   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3
1   3
##
## Within cluster sum of squares by cluster:
## [1]  44.01863  92.65184 158.84732
##  (between_SS / total_SS =  50.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```