Bike Buyers EDA

Pankaj Bhusal 3081563

2025-03-20

1.1 Load the Dataset

```r
# Read the CSV file  and na. strings helps to find empty strings as NA during import
bike_buyers = read.csv("bike_buyers.csv", header=TRUE, na.strings='')

#Display first few columns of datasets
head(bike_buyers)
```

```
##      ID Marital.Status Gender Income Children       Education     Occupation
## 1 12496        Married Female  40000        1       Bachelors Skilled Manual
## 2 24107        Married   Male  30000        3 Partial College       Clerical
## 3 14177        Married   Male  80000        5 Partial College   Professional
## 4 24381         Single   <NA>  70000        0       Bachelors   Professional
## 5 25597         Single   Male  30000        0       Bachelors       Clerical
## 6 13507        Married Female  10000        2 Partial College         Manual
##   Home.Owner Cars Commute.Distance  Region Age Purchased.Bike
## 1        Yes    0        0-1 Miles  Europe  42             No
## 2        Yes    1        0-1 Miles  Europe  43             No
## 3         No    2        2-5 Miles  Europe  60             No
## 4        Yes    1       5-10 Miles Pacific  41            Yes
## 5         No    0        0-1 Miles  Europe  36            Yes
## 6        Yes    0        1-2 Miles  Europe  50             No
```

```r
#see the structure of the datasets
str(bike_buyers)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ ID              : int  12496 24107 14177 24381 25597 13507 27974 19364 22155 19280 ...
##  $ Marital.Status  : chr  "Married" "Married" "Married" "Single" ...
##  $ Gender          : chr  "Female" "Male" "Male" NA ...
##  $ Income          : int  40000 30000 80000 70000 30000 10000 160000 40000 20000 NA ...
##  $ Children        : int  1 3 5 0 0 2 2 1 2 2 ...
##  $ Education       : chr  "Bachelors" "Partial College" "Partial College" "Bachelors" ...
##  $ Occupation      : chr  "Skilled Manual" "Clerical" "Professional" "Professional" ...
##  $ Home.Owner      : chr  "Yes" "Yes" "No" "Yes" ...
##  $ Cars            : int  0 1 2 1 0 0 4 0 2 1 ...
##  $ Commute.Distance: chr  "0-1 Miles" "0-1 Miles" "2-5 Miles" "5-10 Miles" ...
##  $ Region          : chr  "Europe" "Europe" "Europe" "Pacific" ...
##  $ Age             : int  42 43 60 41 36 50 33 43 58 NA ...
##  $ Purchased.Bike  : chr  "No" "No" "No" "Yes" ...
```

1.2 Data Cleaning

```r
# Check the class of the dataset
class(bike_buyers)
```

```
## [1] "data.frame"
```

```r
# Check for missing values
any(is.na(bike_buyers))
```

```
## [1] TRUE
```

```r
# Count the number of missing (NA) values in each column of the data frame
sapply(bike_buyers, function(x) sum(is.na(x)))
```

```
##               ID   Marital.Status           Gender           Income
##                0                7               11                6
##         Children        Education       Occupation       Home.Owner
##                8                0                0                4
##             Cars Commute.Distance           Region              Age
##                9                0                0                8
##    Purchased.Bike
##                0
```

```r
### Converting these categorical columns to factors.
bike_buyers$Marital.Status <- as.factor(bike_buyers$Marital.Status)
bike_buyers$Gender <- as.factor(bike_buyers$Gender)
bike_buyers$Home.Owner <- as.factor(bike_buyers$Home.Owner)
bike_buyers$Purchased.Bike <- as.factor(bike_buyers$Purchased.Bike)

### Check the structure after conversion
str(bike_buyers)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ ID              : int  12496 24107 14177 24381 25597 13507 27974 19364 22155 19280 ...
##  $ Marital.Status  : Factor w/ 2 levels "Married","Single": 1 1 1 2 2 1 2 1 NA 1 ...
##  $ Gender          : Factor w/ 2 levels "Female","Male": 1 2 2 NA 2 1 2 2 2 2 ...
##  $ Income          : int  40000 30000 80000 70000 30000 10000 160000 40000 20000 NA ...
##  $ Children        : int  1 3 5 0 0 2 2 1 2 2 ...
##  $ Education       : chr  "Bachelors" "Partial College" "Partial College" "Bachelors" ...
##  $ Occupation      : chr  "Skilled Manual" "Clerical" "Professional" "Professional" ...
##  $ Home.Owner      : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 NA 2 2 2 ...
##  $ Cars            : int  0 1 2 1 0 0 4 0 2 1 ...
##  $ Commute.Distance: chr  "0-1 Miles" "0-1 Miles" "2-5 Miles" "5-10 Miles" ...
##  $ Region          : chr  "Europe" "Europe" "Europe" "Pacific" ...
##  $ Age             : int  42 43 60 41 36 50 33 43 58 NA ...
##  $ Purchased.Bike  : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 2 2 1 2 ...
```

Reexamine the data after cleaning

```r
summary(bike_buyers)
```

```
##        ID          Marital.Status     Gender          Income          Children
##  Min.   :11000   Married:535    Female:489   Min.   : 10000   Min.   :0.00
##  1st Qu.:15291   Single :458    Male  :500   1st Qu.: 30000   1st Qu.:0.00
##  Median :19744   NA's   :  7    NA's  : 11   Median : 60000   Median :2.00
##  Mean   :19966                               Mean   : 56268   Mean   :1.91
##  3rd Qu.:24471                               3rd Qu.: 70000   3rd Qu.:3.00
##  Max.   :29447                               Max.   :170000   Max.   :5.00
##                                              NA's   :6        NA's   :8
##   Education          Occupation         Home.Owner      Cars
##  Length:1000        Length:1000        No  :314   Min.   :0.000
##  Class :character   Class :character   Yes :682   1st Qu.:1.000
##  Mode  :character   Mode  :character   NA's:  4   Median :1.000
##                                                   Mean   :1.455
##                                                   3rd Qu.:2.000
##                                                   Max.   :4.000
##                                                   NA's   :9
##  Commute.Distance      Region              Age        Purchased.Bike
##  Length:1000        Length:1000        Min.   :25.00   No :519
##  Class :character   Class :character   1st Qu.:35.00   Yes:481
##  Mode  :character   Mode  :character   Median :43.00
##                                        Mean   :44.18
##                                        3rd Qu.:52.00
##                                        Max.   :89.00
##                                        NA's   :8
```

1.3 Fill missing categorical values with the most frequent value (mode)

```r
fill_mode <- function(x) {
  if (is.factor(x) || is.character(x)) {
    x[is.na(x)] <- as.character(names(sort(table(x), decreasing = TRUE))[1])
  }
  return(x)
}

# Convert factors variables to characters as it allows for easier manipulation when filling Missing val
bike_buyers$Marital.Status <- as.character(bike_buyers$Marital.Status)
bike_buyers$Gender <- as.character(bike_buyers$Gender)
bike_buyers$Home.Owner <- as.character(bike_buyers$Home.Owner)

# Fill missing values with mode( it replaces NA values with most Mode value in each column)
bike_buyers$Marital.Status <- fill_mode(bike_buyers$Marital.Status)
bike_buyers$Gender <- fill_mode(bike_buyers$Gender)
bike_buyers$Home.Owner <- fill_mode(bike_buyers$Home.Owner)
```

1.4 Filling missing numerical values with the median

```r
fill_median <- function(x) {
  x[is.na(x)] <- median(x, na.rm = TRUE)
  return(x)
}

bike_buyers$Income <- fill_median(bike_buyers$Income)
bike_buyers$Children <- fill_median(bike_buyers$Children)
bike_buyers$Cars <- fill_median(bike_buyers$Cars)
bike_buyers$Age <- fill_median(bike_buyers$Age)
```

1.5 Check and count missing values again per column

```r
any(is.na(bike_buyers))
```

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

```r
sapply(bike_buyers, function(x) sum(is.na(x)))
```

```
##               ID   Marital.Status           Gender           Income
##                0                0                0                0
##         Children        Education       Occupation       Home.Owner
##                0                0                0                0
##             Cars Commute.Distance           Region              Age
##                0                0                0                0
##    Purchased.Bike
##                0
```

2.1 Summary Statistics

The summary(bike_buyers) command provides a quick numerical and categorical summary of the dataset. For numerical variables (such as Age or Annual Income), it shows metrics like minimum, median, and maximum values. For categorical variables (such as Gender and Marital Status), it displays frequency counts

```r
summary(bike_buyers)
```

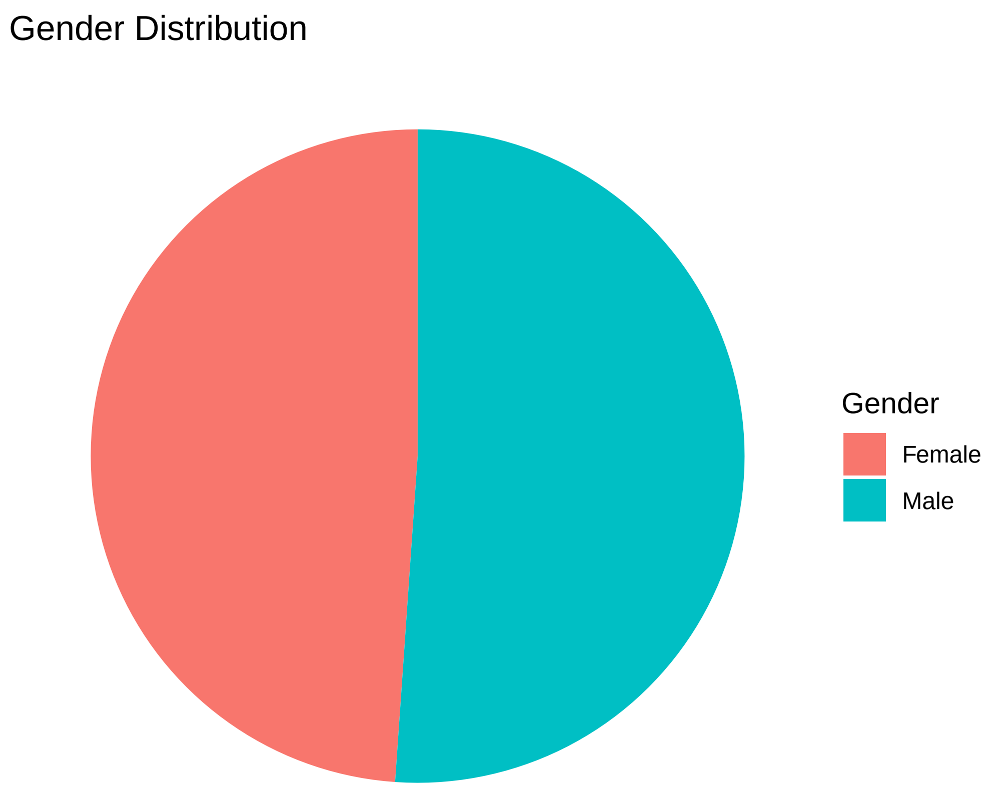
```
##        ID        Marital.Status        Gender              Income
##  Min.   :11000   Length:1000        Length:1000        Min.   : 10000
##  1st Qu.:15291   Class :character   Class :character   1st Qu.: 30000
##  Median :19744   Mode  :character   Mode  :character   Median : 60000
##  Mean   :19966                                         Mean   : 56290
##  3rd Qu.:24471                                         3rd Qu.: 70000
##  Max.   :29447                                         Max.   :170000
##     Children      Education          Occupation         Home.Owner
##  Min.   :0.000   Length:1000        Length:1000        Length:1000
##  1st Qu.:0.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
```

```
##  Mean   :1.911
##  3rd Qu.:3.000
##  Max.   :5.000
##       Cars        Commute.Distance      Region               Age
##  Min.   :0.000   Length:1000        Length:1000        Min.   :25.00
##  1st Qu.:1.000   Class :character   Class :character   1st Qu.:35.00
##  Median :1.000   Mode  :character   Mode  :character   Median :43.00
##  Mean   :1.451                                         Mean   :44.17
##  3rd Qu.:2.000                                         3rd Qu.:52.00
##  Max.   :4.000                                         Max.   :89.00
##  Purchased.Bike
##  No :519
##  Yes:481
##
##
##
##
```

2.2 Pie Chart: Gender Distribution

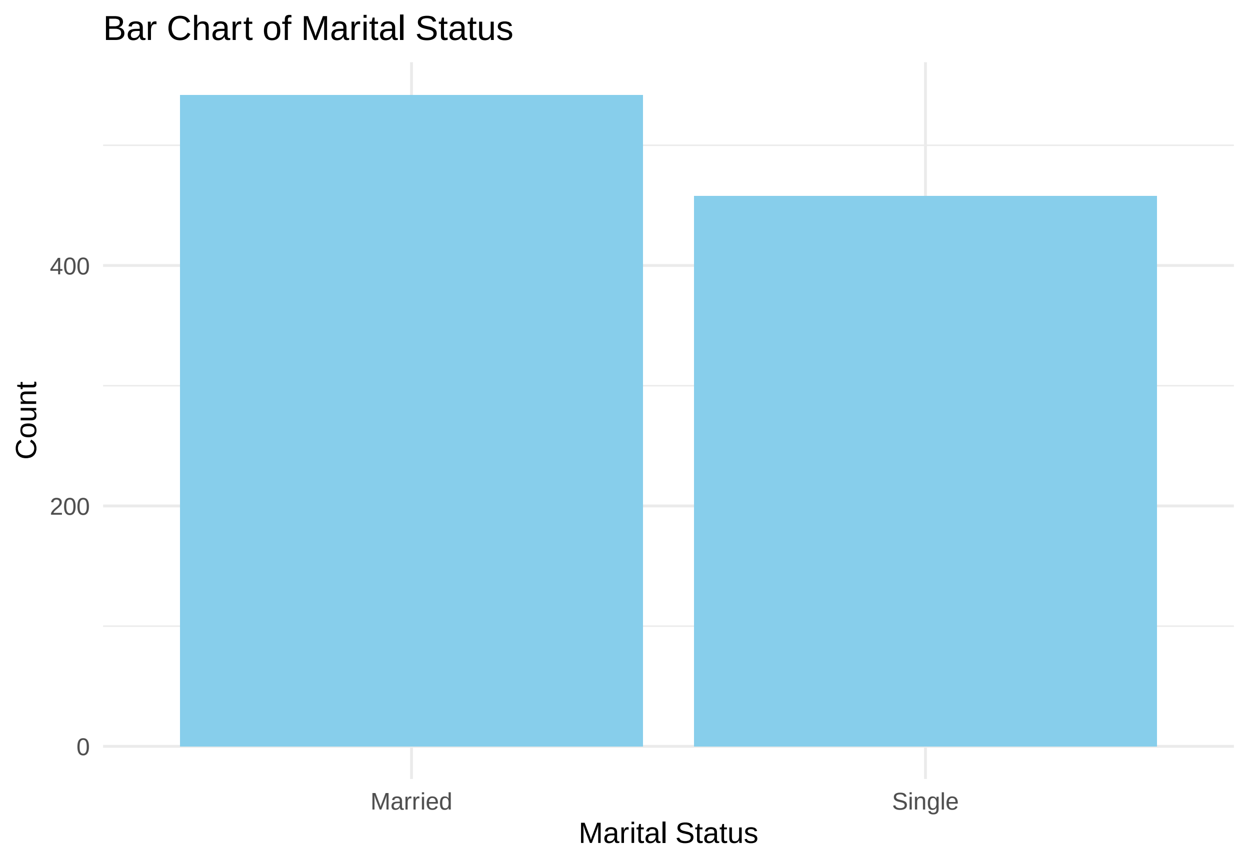
The pie chart (of gender) illustrates the relative proportions of genders, making it easy to see which group is more prevalent.

```
ggplot(bike_buyers, aes(x = "", fill = Gender)) +
  geom_bar( width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Gender Distribution", fill = "Gender") +
  theme_void()
```



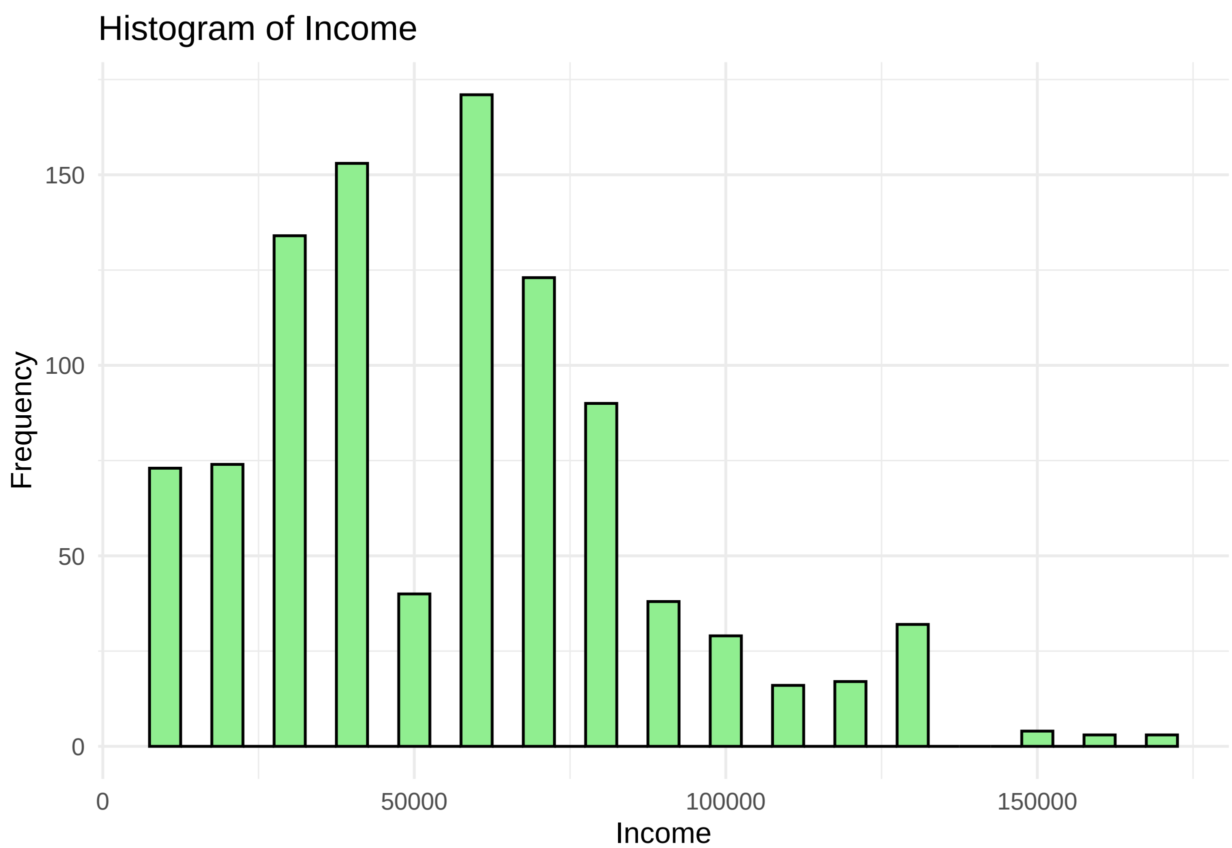
2.3 Bar Chart: Count of Marital Status

```
ggplot(bike_buyers, aes(x = Marital.Status)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Bar Chart of Marital Status", x = "Marital Status", y = "Count") +
  theme_minimal()
```



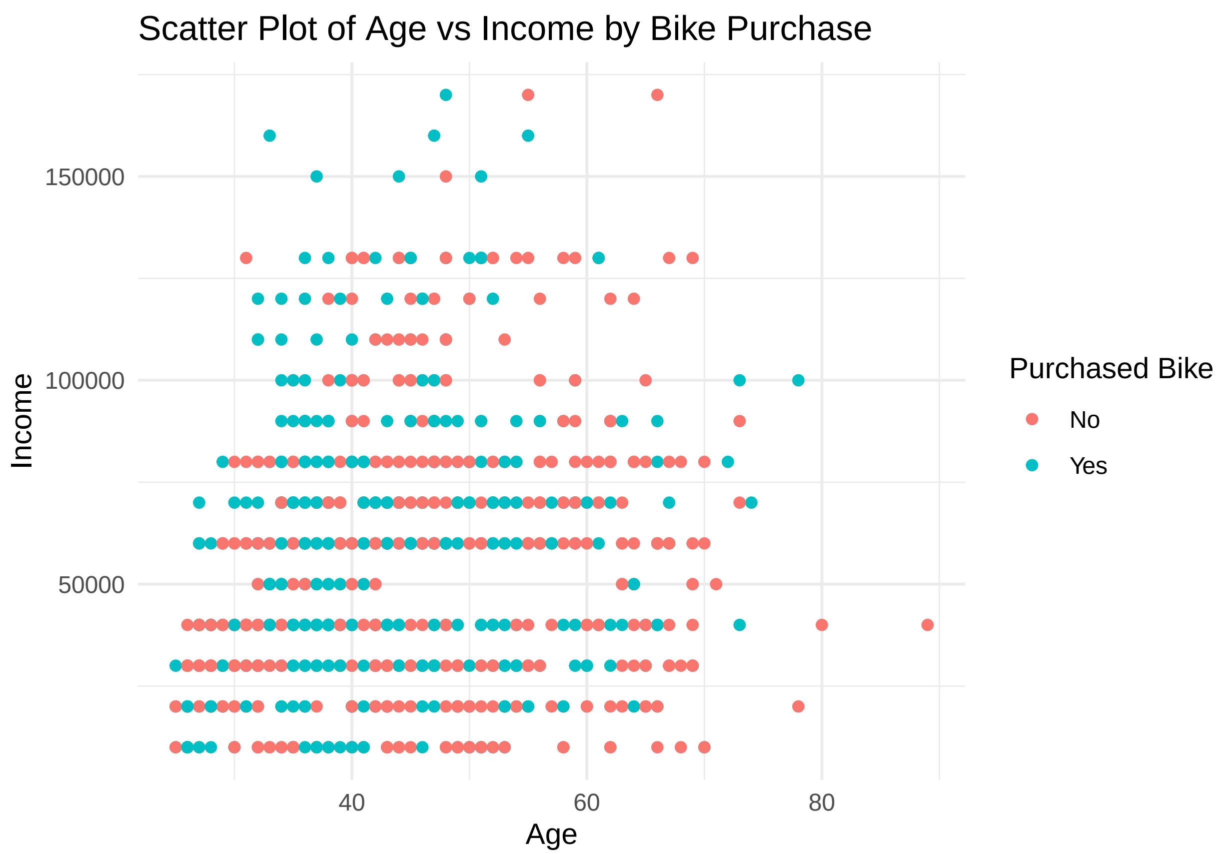
2.4 Histogram: Income Distribution

```
ggplot(bike_buyers, aes(x = Income)) +
  geom_histogram(binwidth = 5000, fill = "lightgreen", color = "black") +
  labs(title = "Histogram of Income", x = "Income", y = "Frequency") +
  theme_minimal()
```



2.5 Scatter Plot: Age vs. Income

```
ggplot(bike_buyers, aes(x = Age, y = Income, color = Purchased.Bike)) +
  geom_point() +
  labs(title = "Scatter Plot of Age vs Income by Bike Purchase",
       x = "Age",
       y = "Income",
       color = "Purchased Bike") +
  theme_minimal()
```

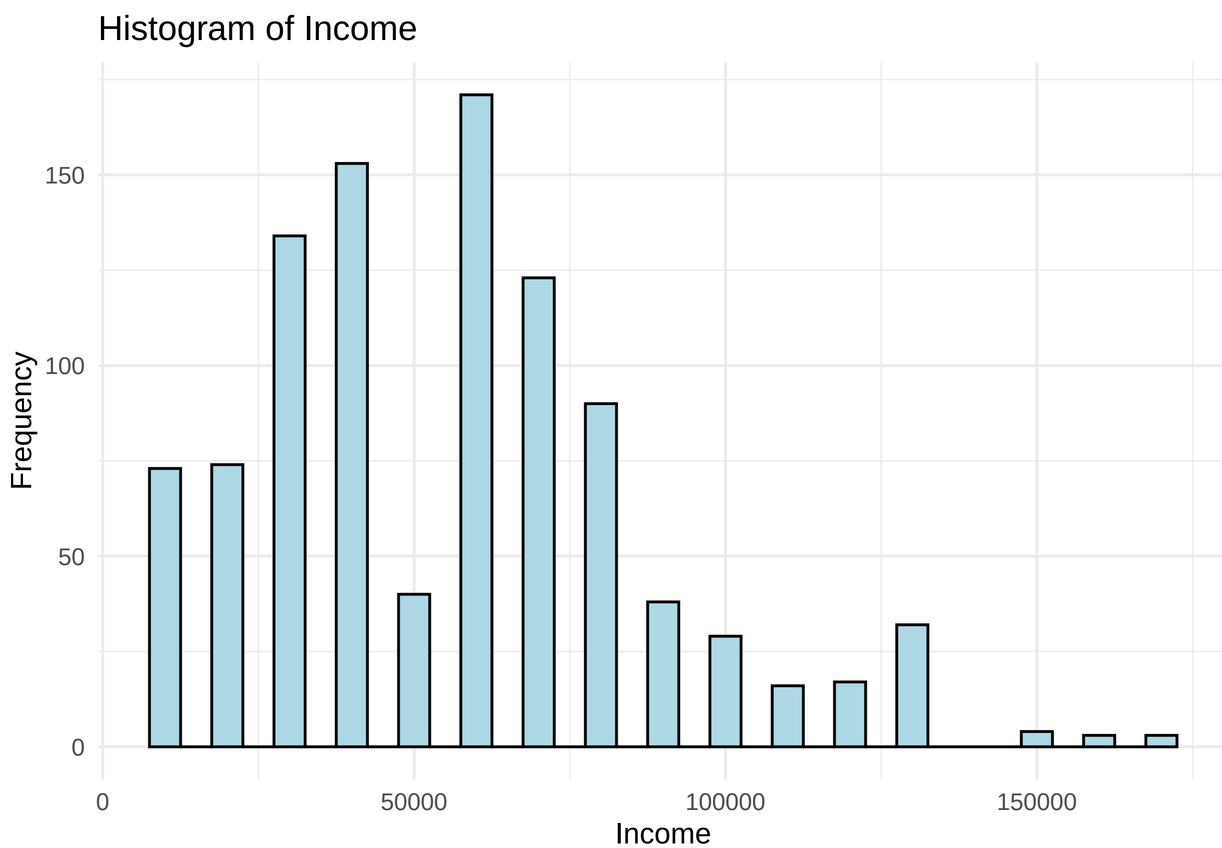


3. Focus on Purchased.Bike Analysis

3.1 Histogram of the Income variable

The histogram provides a visual idea of how income values are spread.

```
# Plot a histogram of the Income variable
ggplot(bike_buyers, aes(x = Income)) +
  geom_histogram(binwidth = 5000, fill = "lightblue", color = "black") +
  labs(title = "Histogram of Income", x = "Income", y = "Frequency") +
  theme_minimal()
```




```r
# Calculate summary statistics for Income
income_mean <- mean(bike_buyers$Income, na.rm = TRUE)
income_median <- median(bike_buyers$Income, na.rm = TRUE)
income_variance <- var(bike_buyers$Income, na.rm = TRUE)

#prints ( strings and number) using cat() function

cat("Summary Statistics for Income:\n")
```

```
## Summary Statistics for Income:
```

```r
cat("Mean: ", income_mean, "\n")
```

```
## Mean:  56290
```

```r
cat("Median: ", income_median, "\n")
```

```
## Median:  60000
```

```r
cat("Variance: ", income_variance, "\n")
```

```
## Variance:  959495395
```

3.2 Grouping Bikers by Income Ranges

```
# Create income groups using cut()
bike_buyers$Income.Range <- cut(bike_buyers$Income,
                                breaks = c(0, 30000, 60000, 90000, 120000, Inf),
                                labels = c("Low", "Medium", "High", "Very High", "Very Very High"),
                                right = FALSE)

# Create a contingency table of Income.Range by Purchased.Bike
income_group_summary <- table(bike_buyers$Income.Range, bike_buyers$Purchased.Bike)

# Add row and column totals to the table
income_group_summary_totals <- addmargins(income_group_summary)

# Display the table
print("Income Group Summary by Purchased.Bike (including totals):")
```

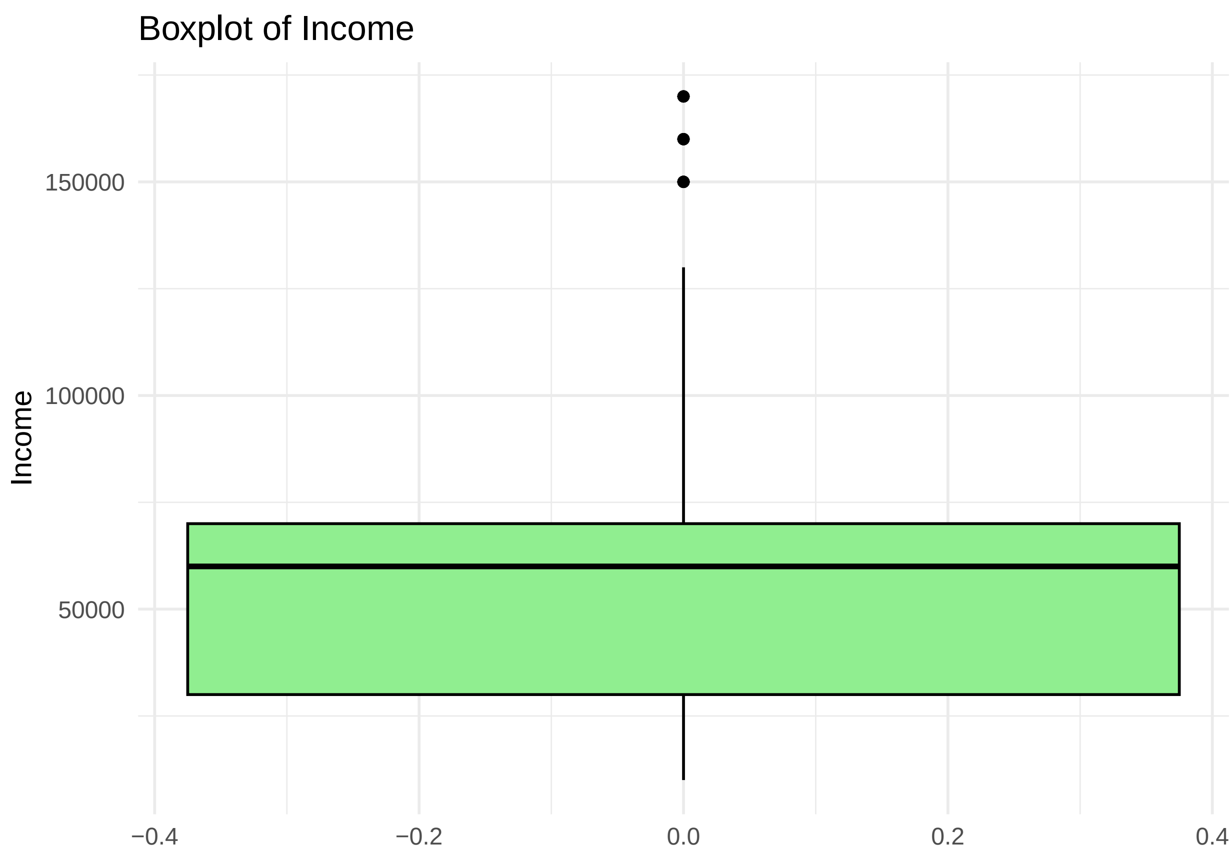
```
## [1] "Income Group Summary by Purchased.Bike (including totals):"
```

```
print(income_group_summary_totals)
```

```
##
##                   No  Yes  Sum
##   Low             88   59  147
##   Medium         165  162  327
##   High           198  186  384
##   Very High       40   43   83
##   Very Very High  28   31   59
##   Sum            519  481 1000
```

3.3 Outlier Exploration for Income with a Boxplot

```
# Boxplot to detect outliers in Income
ggplot(bike_buyers, aes(y = Income)) +
  geom_boxplot(fill = "lightgreen", color = "black") +
  labs(title = "Boxplot of Income", y = "Income") +
  theme_minimal()
```



3.4 Correlation Analysis with Purchased.Bike

From the table below, we can see the following observations:-

- Income has a positive correlation with Purchase.Bike that means higer income user has more chances to buy a bike (i.e 0.0474829)
- cars has a moderate negative correlation(i.e -0.19877383)
- Children has a negative correlation that means more children tends to less likely to buy a bike.(i.e -0.1213416)
- Age has a weaK Negative Correlation means older people slightly less likely to buy(i.e -0.1064722)

```r
# Convert Purchased.Bike to numeric: 1 for "Yes", 0 for "No" for correlation analysis.
bike_buyers$Purchased.Bike.Num <- ifelse(bike_buyers$Purchased.Bike == "Yes", 1, 0)

# Select numeric variables for correlation analysis
library(dplyr)
numeric_vars <- bike_buyers %>%
  select(Income, Age, Children, Cars, Purchased.Bike.Num)

# Compute the correlation matrix for the selected numeric variables
cor_matrix <- cor(numeric_vars, use = "complete.obs")
print("Correlation Matrix:")
```

```
## [1] "Correlation Matrix:"
```

```r
print(cor_matrix)
```

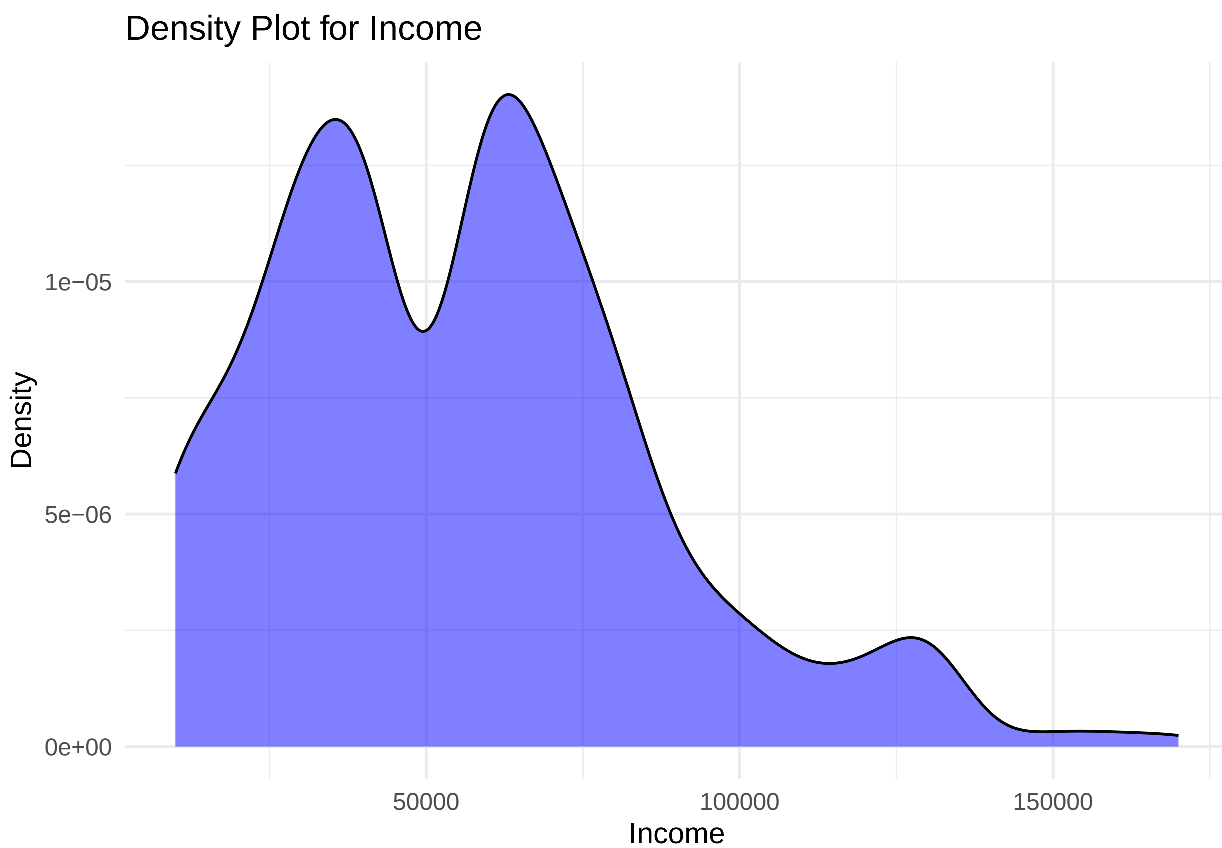
```
##                       Income        Age   Children       Cars
## Income            1.00000000  0.1703264  0.2588561  0.4335637
## Age               0.17032637  1.0000000  0.5256829  0.1842955
## Children          0.25885613  0.5256829  1.0000000  0.2753641
## Cars              0.43356371  0.1842955  0.2753641  1.0000000
## Purchased.Bike.Num 0.04748291 -0.1064722 -0.1213416 -0.1987738
##                   Purchased.Bike.Num
## Income                    0.04748291
## Age                      -0.10647220
## Children                 -0.12134162
## Cars                     -0.19877383
## Purchased.Bike.Num        1.00000000
```

4. Create density plots for Income and ggplot comparing Age and Gender.

4.1 Density Plot for Income

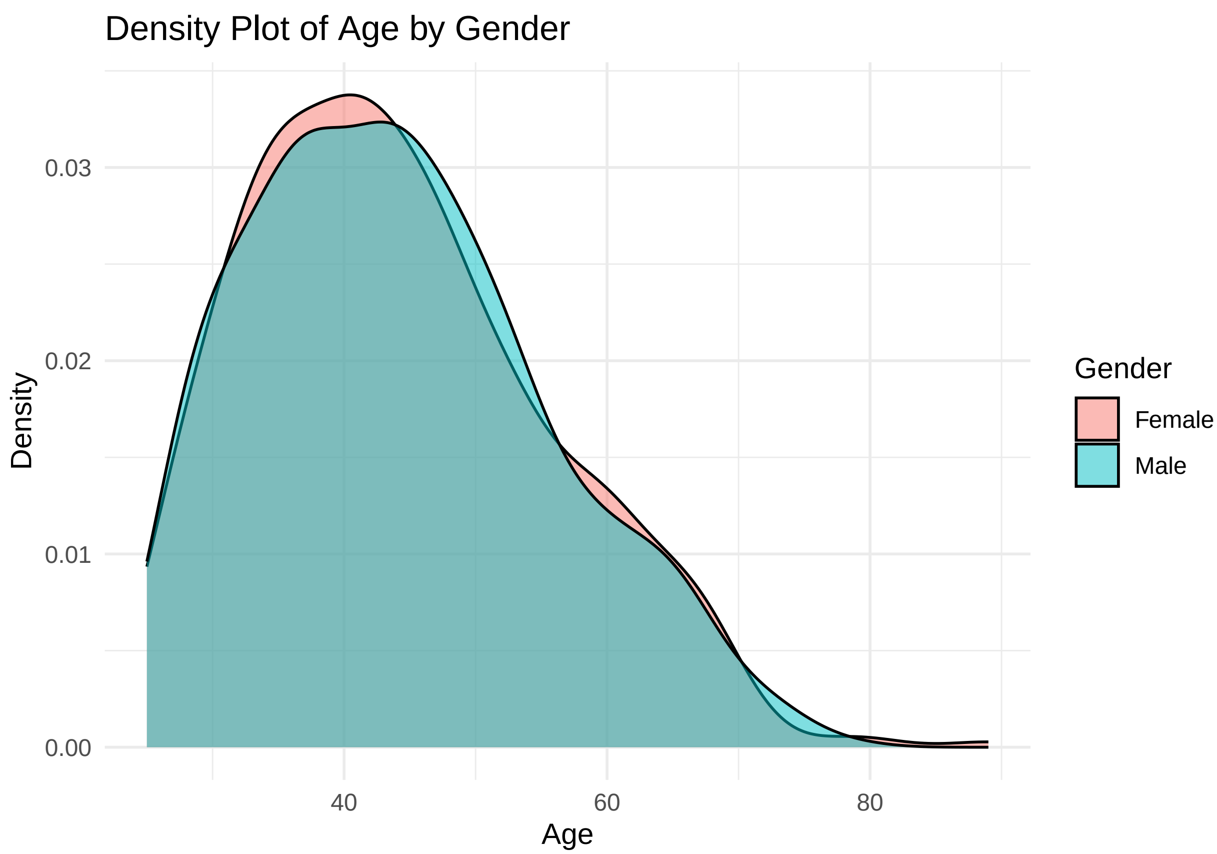
This plot visualizes the distribution of the Income variable using a density plot with a light blue fill.

```r
ggplot(bike_buyers, aes(x = Income)) +
  geom_density(fill = "blue", alpha = 0.5) +
  labs(title = "Density Plot for Income", x = "Income", y = "Density") +
  theme_minimal()
```



4.2 Density Plot Comparing Age by Gender This plot overlays the age density curves for each gender.

```
ggplot(bike_buyers, aes(x = Age, fill = Gender)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot of Age by Gender", x = "Age", y = "Density") + theme_minimal()
```


Density Plot of Age by Gender
Density
0.03
0.02
0.01
0.00
40
60
80
Age
Gender
Female
Male