Assignment1-DAV

EMMANUEL GODWIN BASSEY

2025-03-03

1. Load the Data

```r
# Parse CSV into bike_data
bike_data <- read.csv("Bike Buyers Assignment 1.csv", stringsAsFactors = FALSE)
str(bike_data)
```

```
## '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" "" ...
##  $ 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" ...
```

```r
summary(bike_data)
```

```
##        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   : 56268
##  3rd Qu.:24471                                         3rd Qu.: 70000
##  Max.   :29447                                         Max.   :170000
##                                                        NA's   :6
##     Children     Education          Occupation          Home.Owner
##  Min.   :0.00   Length:1000        Length:1000        Length:1000
##  1st Qu.:0.00   Class :character   Class :character   Class :character
##  Median :2.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1.91
##  3rd Qu.:3.00
##  Max.   :5.00
##  NA's   :8
```

```
##       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   :1.455                                         Mean   :44.18
##  3rd Qu.:2.000                                         3rd Qu.:52.00
##  Max.   :4.000                                         Max.   :89.00
##  NA's   :9                                             NA's   :8
##  Purchased.Bike
##  Length:1000
##  Class :character
##  Mode  :character
##
##
##
##
```

```r
head(bike_data)
```

```
##      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         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
```

2. Data Cleaning

```r
# Checking for duplicate IDs
duplicate_count <- sum(duplicated(bike_data$ID))
cat("Duplicate IDs:", duplicate_count, "\n\n")
```

```
## Duplicate IDs: 0
```

```r
# Expected values for categorical variables
expected_marital <- c("Married", "Single")
expected_gender <- c("Male", "Female")
expected_education <- c("Bachelors", "Partial College", "High School", "Graduate Degree", "Partial High
expected_home_owner <- c("Yes", "No")
expected_commute <- c("0-1 Miles", "1-2 Miles", "2-5 Miles", "5-10 Miles", "10+ Miles")
expected_region <- c("Europe", "North America", "Pacific")
expected_purchase <- c("Yes", "No")
```

```r
# Function to check incorrect values
check_invalid_values <- function(column, expected) {
  invalid_values <- setdiff(unique(bike_data[[column]]), expected)
  cat("Incorrect values in", column, ":", if(length(invalid_values) == 0) "None" else invalid_values, "
}

# Run checks
check_invalid_values("Marital.Status", expected_marital)
```

```
## Incorrect values in Marital.Status :
```

```r
check_invalid_values("Gender", expected_gender)
```

```
## Incorrect values in Gender :
```

```r
check_invalid_values("Education", expected_education)
```

```
## Incorrect values in Education : None
```

```r
check_invalid_values("Home.Owner", expected_home_owner)
```

```
## Incorrect values in Home.Owner :
```

```r
check_invalid_values("Commute.Distance", expected_commute)
```

```
## Incorrect values in Commute.Distance : None
```

```r
check_invalid_values("Region", expected_region)
```

```
## Incorrect values in Region : None
```

```r
check_invalid_values("Purchased.Bike", expected_purchase)
```

```
## Incorrect values in Purchased.Bike : None
```

```r
# Check numeric columns for negative values
for (col in c("Income", "Children", "Cars", "Age")) {
  invalid_values <- bike_data[[col]][bike_data[[col]] < 0]
  cat("Invalid values in", col, ":", if(length(invalid_values) == 0) "None" else invalid_values, "\n")
}
```

```
## Invalid values in Income : NA NA NA NA NA NA
## Invalid values in Children : NA NA NA NA NA NA NA NA
## Invalid values in Cars : NA NA NA NA NA NA NA NA NA
## Invalid values in Age : NA NA NA NA NA NA NA NA
```

3. Identify Missing Values

```r
# Check for missing values in each column, treating empty strings as NA
missing_bike_data <- sapply(bike_data, function(x) sum(is.na(x) | x == ""))
print(missing_bike_data)
```

```
##              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
# Display total missing values across all columns
total_missing <- sum(missing_bike_data)
cat("Total missing values in bike_data:", total_missing, "\n\n")
```

```
## Total missing values in bike_data: 53
```

4. Impute Missing Values

```r
# Convert categorical variables to factors
categorical_vars <- c("Marital.Status", "Gender", "Education", "Occupation",
                      "Home.Owner", "Commute.Distance", "Region", "Purchased.Bike")
bike_data[categorical_vars] <- lapply(bike_data[categorical_vars], as.factor)

# Save indices of missing values
missing_indices <- lapply(bike_data, function(x) which(is.na(x) | x == ""))

# Impute missing values using MICE
imputed_data <- mice(bike_data, m = 5, method = "pmm", seed = 123)
```

```
##
##  iter imp variable
##   1   1  Income  Children  Cars  Age
##   1   2  Income  Children  Cars  Age
##   1   3  Income  Children  Cars  Age
##   1   4  Income  Children  Cars  Age
##   1   5  Income  Children  Cars  Age
##   2   1  Income  Children  Cars  Age
##   2   2  Income  Children  Cars  Age
##   2   3  Income  Children  Cars  Age
##   2   4  Income  Children  Cars  Age
##   2   5  Income  Children  Cars  Age
##   3   1  Income  Children  Cars  Age
##   3   2  Income  Children  Cars  Age
##   3   3  Income  Children  Cars  Age
##   3   4  Income  Children  Cars  Age
##   3   5  Income  Children  Cars  Age
##   4   1  Income  Children  Cars  Age
```

```
##   4   2  Income  Children  Cars  Age
##   4   3  Income  Children  Cars  Age
##   4   4  Income  Children  Cars  Age
##   4   5  Income  Children  Cars  Age
##   5   1  Income  Children  Cars  Age
##   5   2  Income  Children  Cars  Age
##   5   3  Income  Children  Cars  Age
##   5   4  Income  Children  Cars  Age
##   5   5  Income  Children  Cars  Age
```

```r
bike_data <- complete(imputed_data, 1)

# Check missing values after imputation
missing_bike_data_after <- sapply(bike_data, function(x) sum(is.na(x) | x == ""))
print(missing_bike_data_after)
```

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

```r
# Display updated values
cat("Updated imputed values per variable:\n\n")
```

```
## Updated imputed values per variable:
```

```r
for (var in names(missing_indices)) {
  indices <- missing_indices[[var]]
  if (length(indices) > 0) {
    cat("Variable:", var, " | Count:", length(indices), "\n")
    print(data.frame(Row = indices, ImputedValue = bike_data[[var]][indices]))
  }
}
```

```
## Variable: Marital.Status  | Count: 7
##   Row ImputedValue
## 1   9
## 2  28
## 3  50
## 4  99
## 5 151
## 6 235
## 7 302
## Variable: Gender  | Count: 11
##    Row ImputedValue
## 1    4
## 2  155
## 3  336
```

```
## 4  602
## 5  689
## 6  696
## 7  868
## 8  909
## 9  952
## 10 974
## 11 998
## Variable: Income  | Count: 6
##   Row ImputedValue
## 1  10        20000
## 2 111        10000
## 3 192        20000
## 4 302        20000
## 5 442        90000
## 6 510        70000
## Variable: Children  | Count: 8
##   Row ImputedValue
## 1 118            2
## 2 218            0
## 3 387            2
## 4 550            4
## 5 639            2
## 6 689            2
## 7 806            3
## 8 961            1
## Variable: Home.Owner  | Count: 4
##   Row ImputedValue
## 1   7
## 2 366
## 3 647
## 4 944
## Variable: Cars  | Count: 9
##   Row ImputedValue
## 1  13            4
## 2 197            0
## 3 203            0
## 4 352            0
## 5 449            0
## 6 512            0
## 7 562            2
## 8 616            0
## 9 934            2
## Variable: Age  | Count: 8
##   Row ImputedValue
## 1  10           46
## 2  99           47
## 3 226           41
## 4 372           67
## 5 555           68
## 6 689           43
## 7 771           48
## 8 987           47
```

5. Checking for Outliers

```r
# Define a function to detect outliers using the IQR method
detect_outliers <- function(x) {
  x_clean <- na.omit(x)
  Q1 <- quantile(x_clean, 0.25)
  Q3 <- quantile(x_clean, 0.75)
  IQR_val <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR_val
  upper_bound <- Q3 + 1.5 * IQR_val
  outliers <- x_clean[x_clean < lower_bound | x_clean > upper_bound]
  return(outliers)
}

# Detect and print outliers for Income, Age, Children, and Cars
outlier_vars <- c("Income", "Age", "Children", "Cars")
for (var in outlier_vars) {
  outliers <- detect_outliers(bike_data[[var]])
  cat(var, "outliers:", if (length(outliers) == 0) "None" else outliers, "\n")
  cat("Count of", var, "outliers:", length(outliers), "\n\n")
}
```

```
## Income outliers: 160000 170000 170000 150000 160000 150000 160000 150000 170000 150000
## Count of Income outliers: 10
##
## Age outliers: 78 89 80 78
## Count of Age outliers: 4
##
## Children outliers: None
## Count of Children outliers: 0
##
## Cars outliers: 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## Count of Cars outliers: 60
```

6. Data Visualization

```r
# Summary of Variables
str(bike_data)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ ID              : int  12496 24107 14177 24381 25597 13507 27974 19364 22155 19280 ...
##  $ Marital.Status  : Factor w/ 3 levels "","Married","Single": 2 2 2 3 3 2 3 2 1 2 ...
##  $ Gender          : Factor w/ 3 levels "","Female","Male": 2 3 3 1 3 2 3 3 3 3 ...
##  $ Income          : int  40000 30000 80000 70000 30000 10000 160000 40000 20000 20000 ...
##  $ Children        : int  1 3 5 0 0 2 2 1 2 2 ...
##  $ Education       : Factor w/ 5 levels "Bachelors","Graduate Degree",..: 1 4 4 1 1 4 3 1 5 4 ...
##  $ Occupation      : Factor w/ 5 levels "Clerical","Management",..: 5 1 4 4 1 3 2 5 1 3 ...
##  $ Home.Owner      : Factor w/ 3 levels "","No","Yes": 3 3 2 3 2 3 1 3 3 3 ...
##  $ Cars            : int  0 1 2 1 0 0 4 0 2 1 ...
##  $ Commute.Distance: Factor w/ 5 levels "0-1 Miles","1-2 Miles",..: 1 1 4 5 1 2 1 1 5 1 ...
```

```
##  $ Region          : Factor w/ 3 levels "Europe","North America",..: 1 1 1 3 1 1 3 1 3 1 ...
##  $ Age             : int  42 43 60 41 36 50 33 43 58 46 ...
##  $ Purchased.Bike  : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 2 2 1 2 ...
```

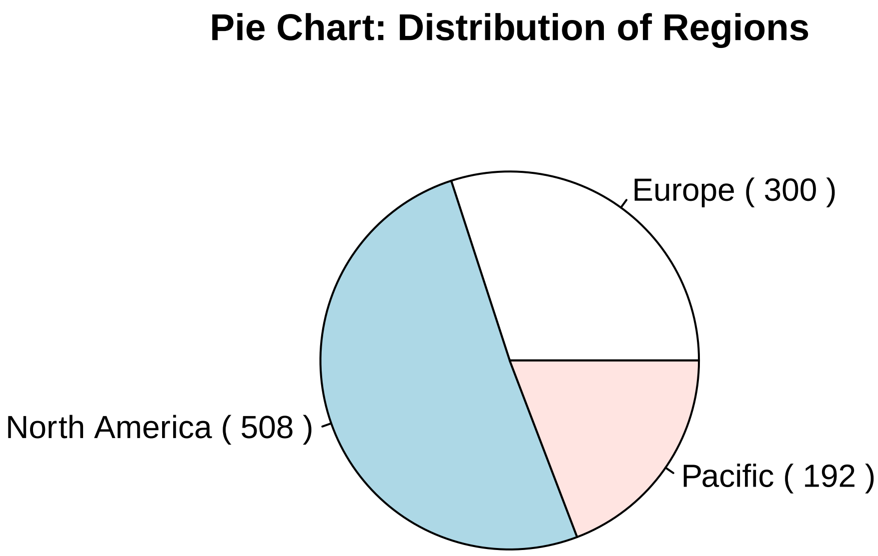
```
summary(bike_data)
```

```
##        ID        Marital.Status    Gender        Income          Children    
##  Min.   :11000          :  7          : 11   Min.   : 10000   Min.   :0.000  
##  1st Qu.:15291   Married:535    Female:489   1st Qu.: 30000   1st Qu.:0.000  
##  Median :19744   Single :458    Male  :500   Median : 60000   Median :2.000  
##  Mean   :19966                               Mean   : 56160   Mean   :1.911  
##  3rd Qu.:24471                               3rd Qu.: 70000   3rd Qu.:3.000  
##  Max.   :29447                               Max.   :170000   Max.   :5.000  
##               Education            Occupation  Home.Owner      Cars     
##  Bachelors          :306   Clerical      :177      :  4    Min.   :0.00  
##  Graduate Degree    :174   Management    :173   No :314    1st Qu.:1.00  
##  High School        :179   Manual        :119   Yes:682    Median :1.00  
##  Partial College    :265   Professional  :276              Mean   :1.45  
##  Partial High School: 76   Skilled Manual:255              3rd Qu.:2.00  
##                                                            Max.   :4.00  
##    Commute.Distance           Region         Age        Purchased.Bike
##  0-1 Miles :366     Europe       :300   Min.   :25.00   No :519       
##  1-2 Miles :169     North America:508   1st Qu.:35.00   Yes:481       
##  10+ Miles :111     Pacific      :192   Median :43.00                 
##  2-5 Miles :162                         Mean   :44.23                 
##  5-10 Miles:192                         3rd Qu.:52.00                 
##                                         Max.   :89.00                 
```

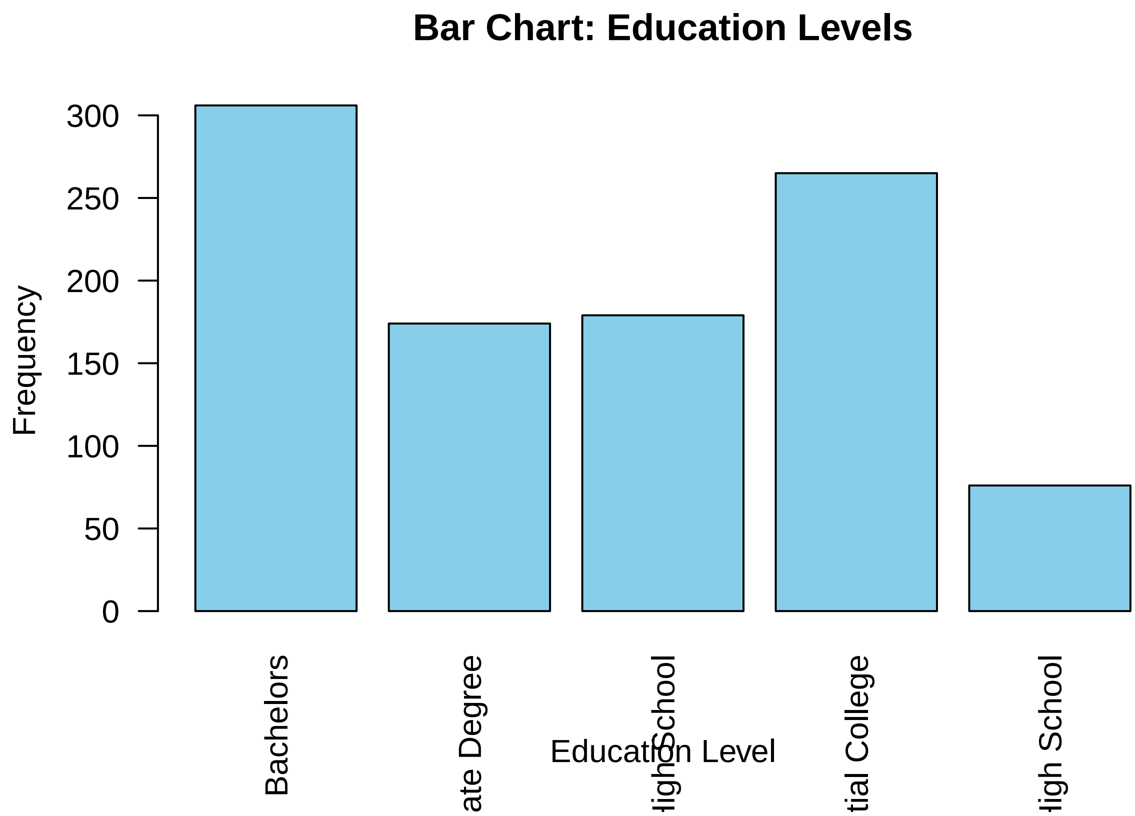
```
numeric_vars <- sapply(bike_data, is.numeric)
describe(bike_data[, numeric_vars])
```

```
##          vars    n     mean       sd median  trimmed      mad   min    max
## ID          1 1000 19965.99  5347.33  19744 19925.80  6848.13 11000  29447
## Income      2 1000 56160.00 31093.75  60000 53562.50 29652.00 10000 170000
## Children    3 1000     1.91     1.62      2     1.79     1.48     0      5
## Cars        4 1000     1.45     1.13      1     1.36     1.48     0      4
## Age         5 1000    44.24    11.37     43    43.54    11.86    25     89
##           range skew kurtosis     se
## ID        18447 0.05    -1.19 169.10
## Income   160000 0.75     0.49 983.27
## Children      5 0.39    -1.02   0.05
## Cars          4 0.42    -0.41   0.04
## Age          64 0.52    -0.27   0.36
```

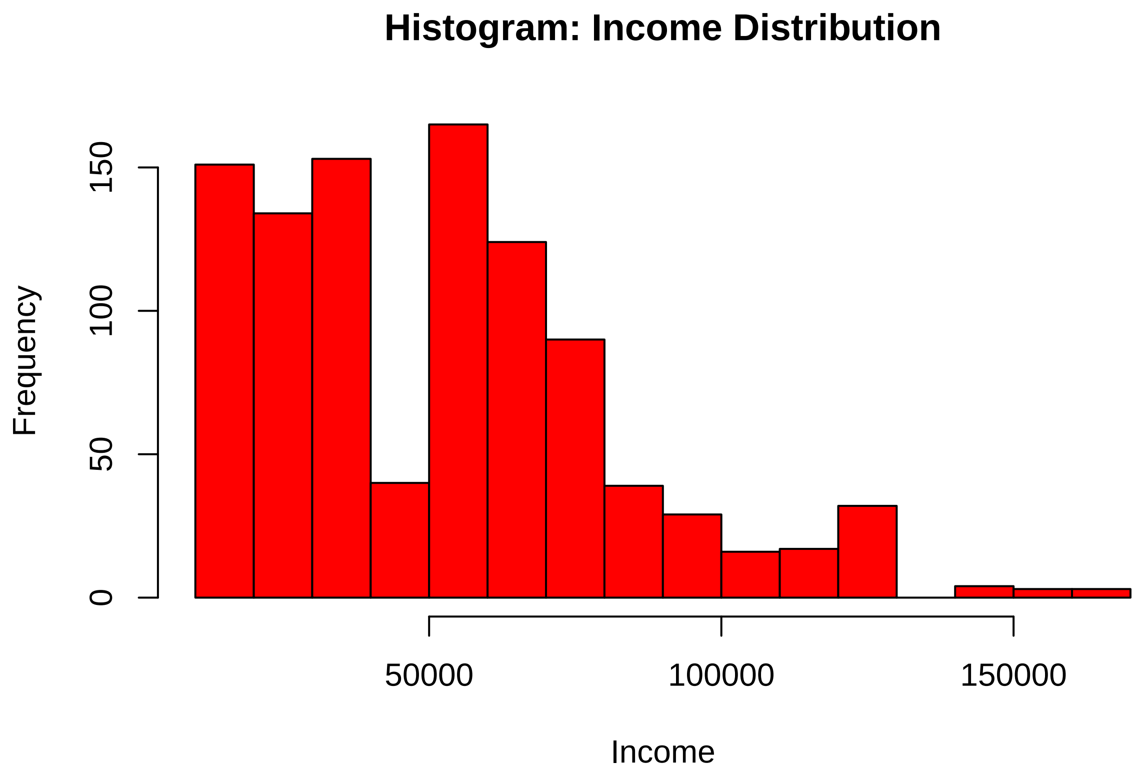
```
# Pie Chart for Region
region_counts <- table(bike_data$Region)
pie(region_counts, main = "Pie Chart: Distribution of Regions", labels = paste(names(region_counts), "(
```



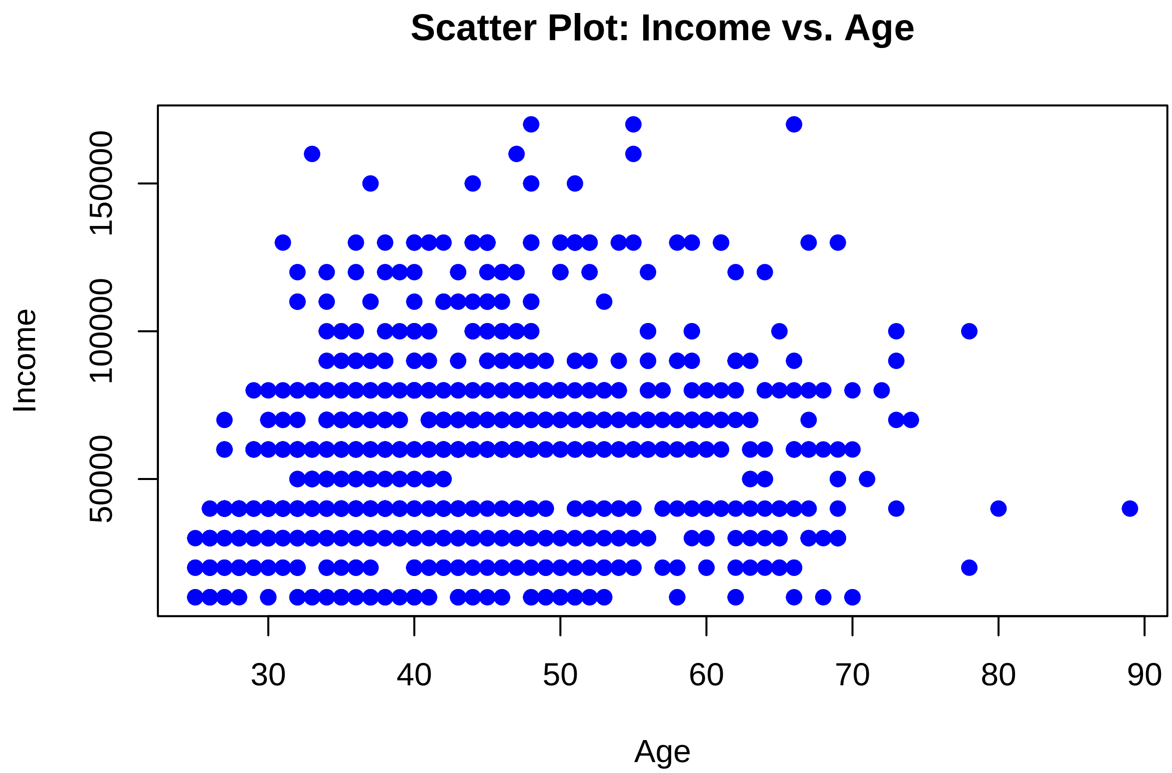

```
# Bar Chart for Education Levels
barplot(table(bike_data$Education), main = "Bar Chart: Education Levels", xlab = "Education Level", ylab
```




```
# Histogram for Income Distribution
hist(bike_data$Income, breaks = 20, main = "Histogram: Income Distribution", xlab = "Income", ylab = "F
```

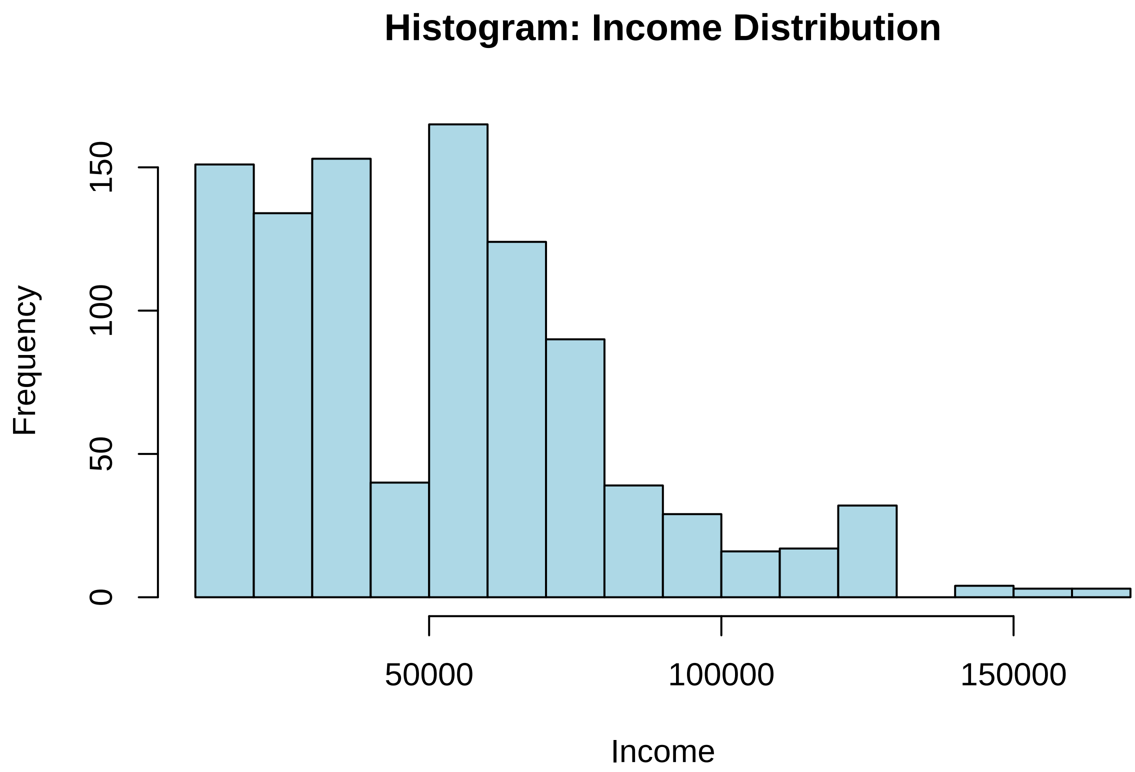



```
# Scatter Plot for Income vs. Age
plot(bike_data$Age, bike_data$Income, main = "Scatter Plot: Income vs. Age", xlab = "Age", ylab = "Incom
```



7. Purchased Bike Analysis

```
# Histogram of Income Variable with Summary Statistics
hist(bike_data$Income, breaks = 20, main = "Histogram: Income Distribution", xlab = "Income", ylab = "F
```



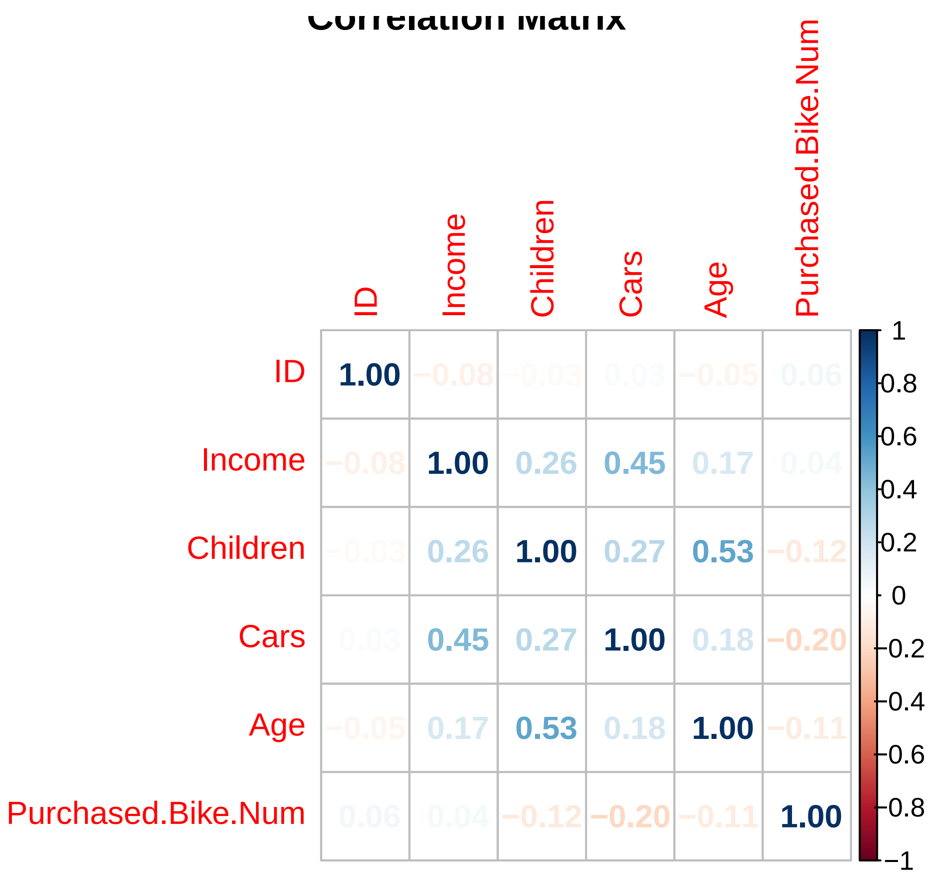

```
# Summary Statistics
income_stats <- c(Mean = mean(bike_data$Income), Median = median(bike_data$Income), Variance = var(bike_
print(income_stats)
```

```
##      Mean    Median  Variance
##     56160     60000 966821221
```

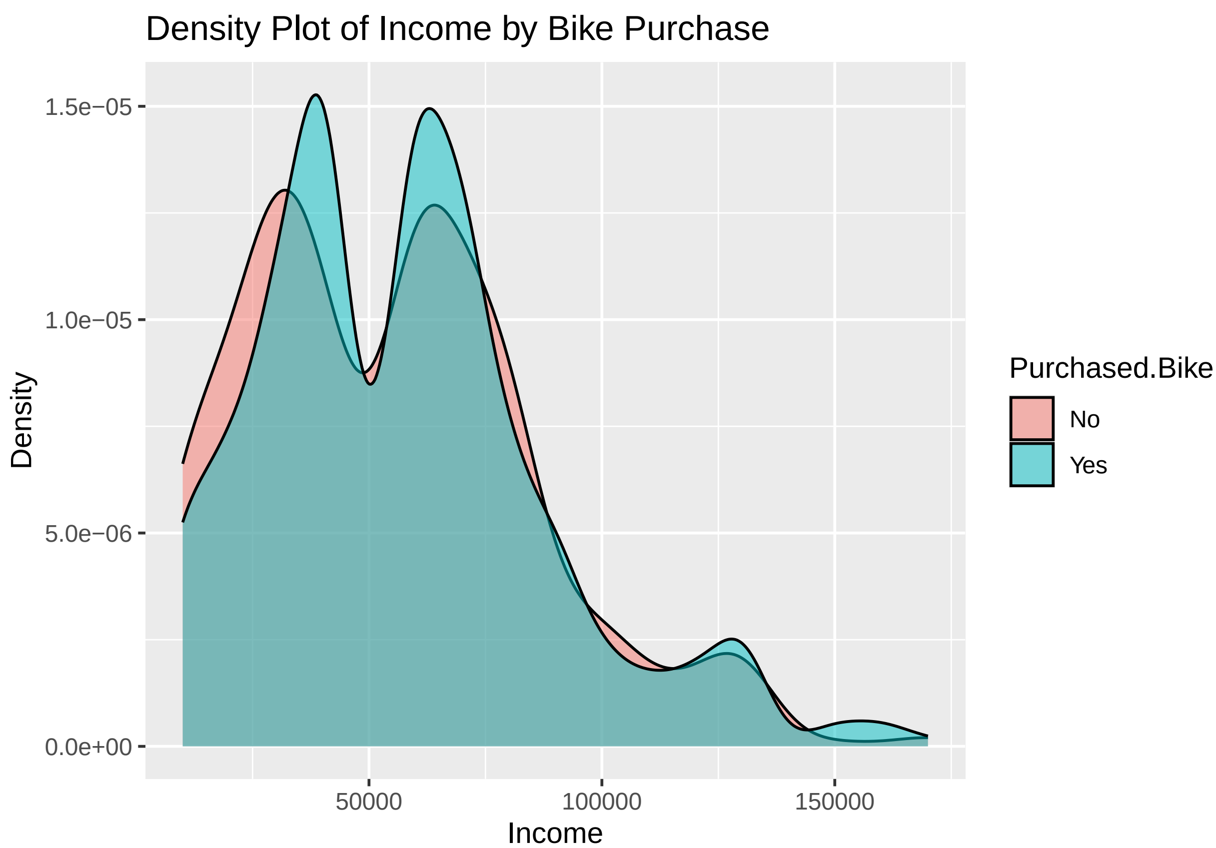
```
# Grouping Bikers by Income Ranges
bike_data$Income_Group <- cut(bike_data$Income, breaks = quantile(bike_data$Income, probs = c(0, 0.33, 
                              include.lowest = TRUE, labels = c("Low", "Medium", "High"))
income_group_summary <- bike_data %>%
  group_by(Income_Group) %>%
  summarise(Total_Count = n(),
            Purchased_Count = sum(Purchased.Bike == "Yes", na.rm = TRUE),
            Purchased_Percent = round(100 * mean(Purchased.Bike == "Yes", na.rm = TRUE), 2))
print(income_group_summary)
```

```
## # A tibble: 3 x 4
##   Income_Group Total_Count Purchased_Count Purchased_Percent
##   <fct>              <int>           <int>             <dbl>
## 1 Low                  438             205              46.8
## 2 Medium               329             167              50.8
## 3 High                 233             109              46.8
```

```r
# Correlation of Attributes with Purchased Bike
bike_data$Purchased.Bike.Num <- ifelse(bike_data$Purchased.Bike == "Yes", 1, 0)
correlations <- cor(bike_data[, sapply(bike_data, is.numeric)], use = "complete.obs")
corrplot(correlations, method = "number", title = "Correlation Matrix")
```

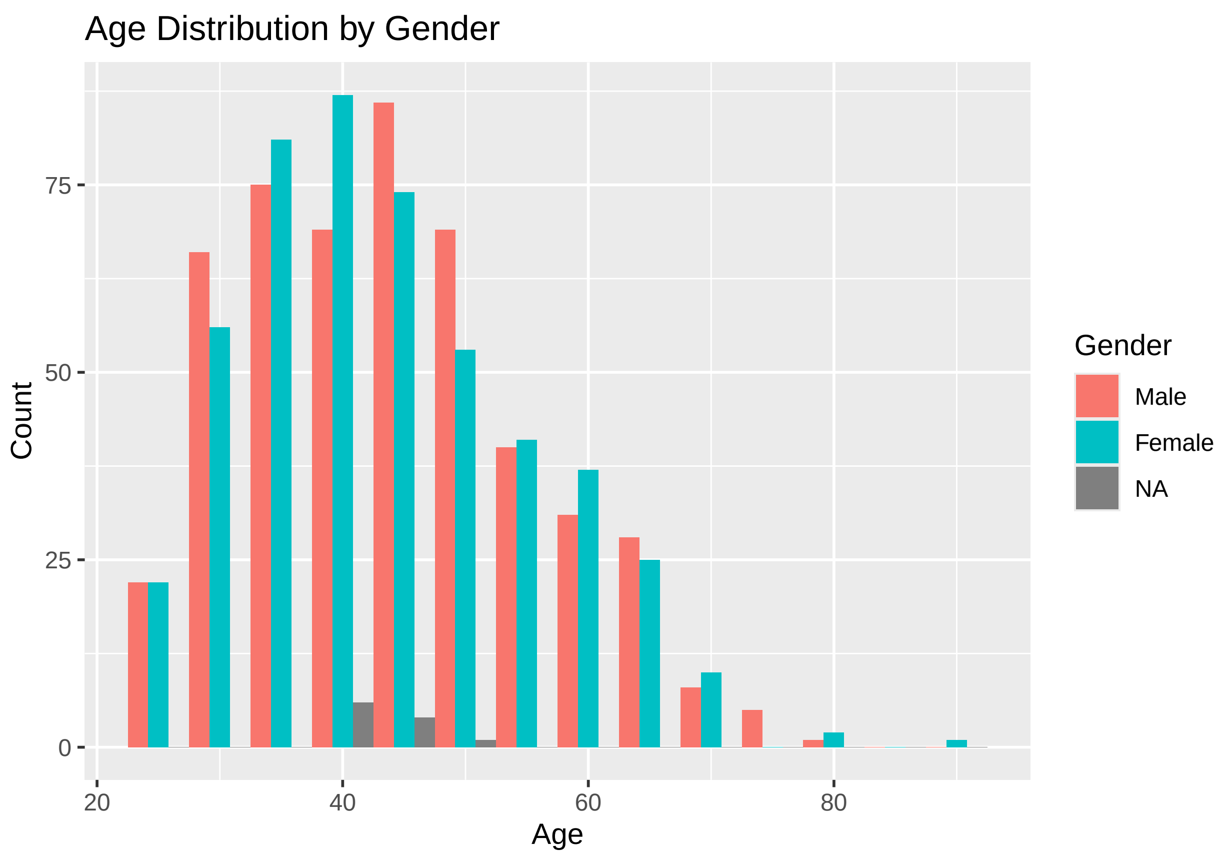



```r
# Density Plot: Income by Bike Purchase
ggplot(bike_data, aes(x = Income, fill = Purchased.Bike)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot of Income by Bike Purchase", x = "Income", y = "Density")
```




```r
# Clean Gender Before Plot
bike_data$Gender[bike_data$Gender == ""] <- "Missing"
bike_data$Gender <- factor(bike_data$Gender, levels = c("Male", "Female", "Missing"))

# Age vs Gender Histogram
ggplot(bike_data, aes(x = Age, fill = Gender)) +
  geom_histogram(binwidth = 5, position = "dodge") +
  labs(title = "Age Distribution by Gender", x = "Age", y = "Count")
```



8. Conclusions & Next Steps

- **Data Cleaning:** Missing values imputed using mice, outliers detected using IQR.
- **EDA Findings:** Distributions of key variables analyzed, correlations identified.
- **Next Steps:**
  - Further statistical tests (e.g., Chi-square for categorical variables).
  - Consider transformations for skewed variables like Income.
  - Scale the process for larger datasets, ensuring robust missing-value handling.

```
##      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 Income_Group
## 1        Yes    0        0-1 Miles  Europe  42             No          Low
## 2        Yes    1        0-1 Miles  Europe  43             No          Low
## 3         No    2        2-5 Miles  Europe  60             No         High
## 4        Yes    1       5-10 Miles Pacific  41            Yes       Medium
## 5         No    0        0-1 Miles  Europe  36            Yes          Low
## 6        Yes    0        1-2 Miles  Europe  50             No          Low
##   Purchased.Bike.Num
```

```
## 1                     0
## 2                     0
## 3                     0
## 4                     1
## 5                     1
## 6                     0
```