Final Project

2023-12-16

Histogram of Total Monthly Expenses

```
# Load the readxl package
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.3.2
```

```
# Load data from the Excel file into a data frame
university_students_data <- read_excel("university_students_data.xlsx")
```

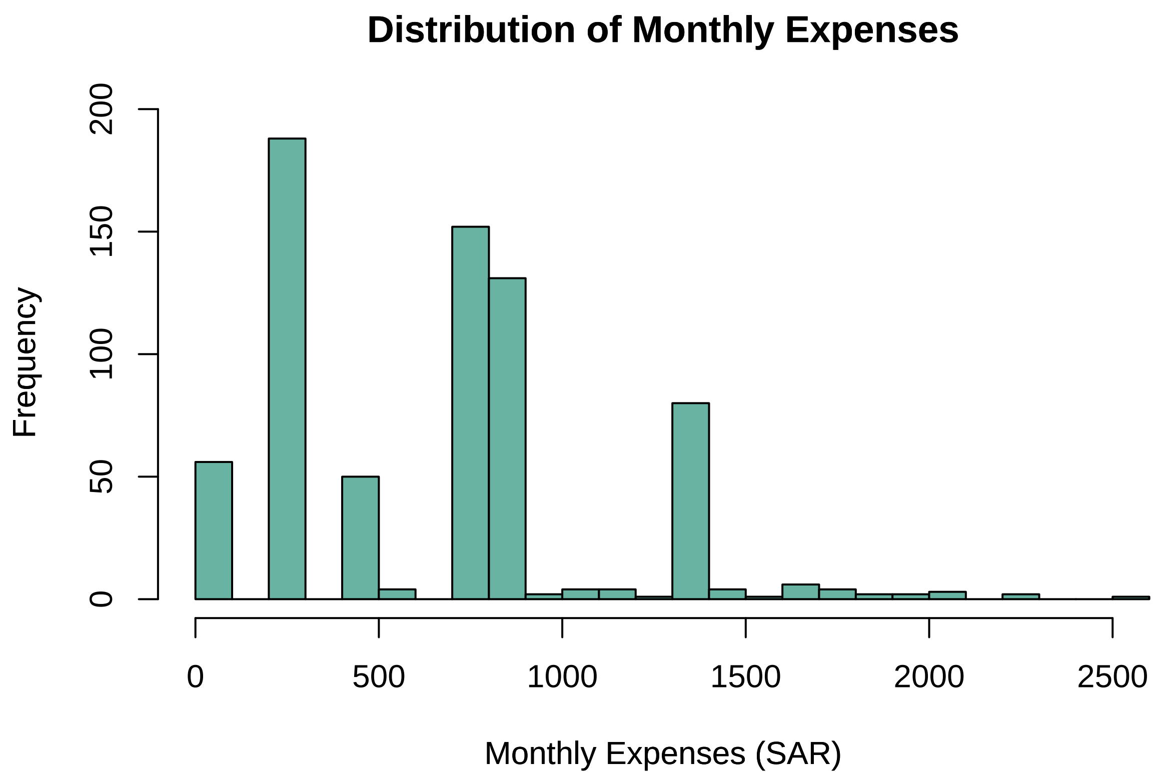
```
# Clean the Monthly_expenses_$ column: convert to numeric and remove missing/NA values
university_students_data$Monthly_expenses <- as.numeric(university_students_data$Monthly_expenses)

# Remove rows with missing or NA values in Monthly_expenses_$
cleaned_dataset <- na.omit(university_students_data)
```

```
View(cleaned_dataset)

# Check for missing values in the entire dataset
any_missing <- any(is.na(cleaned_dataset))

# Output whether there are missing values or not
if (any_missing) {
  print("There are missing values in the dataset.")
} else {
  print("No missing values found in the dataset.")
}
```

```
## [1] "No missing values found in the dataset."
```

```
# Calculate the range of expenses
min_expense <- min(cleaned_dataset$Monthly_expenses)
max_expense <- max(cleaned_dataset$Monthly_expenses)

# Create a histogram with adjusted axes
hist(cleaned_dataset$Monthly_expenses,
     main = "Distribution of Monthly Expenses",
     xlab = "Monthly Expenses (SAR)",
     ylab = "Frequency",
     col = "#69b3a2",
     border = "black",
     breaks = 20,
```

```r
     xlim = c(min_expense, max_expense),  # Set x-axis limits
     ylim = c(0, max(table(cut(cleaned_dataset$Monthly_expenses, breaks = 20))) + 5)  # Set y-axis limi
)

# Add a title and labels to the histogram
title(main = "Distribution of Monthly Expenses",
      xlab = "Monthly Expenses (SAR)",
      ylab = "Frequency")
```



2. Loading and Exploring Data

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

```
## # A tibble: 6 x 17
##   Gender   Age Study_year Living Scholarship Part_time_job Transporting  Smoking
##   <chr>  <dbl>      <dbl> <chr>  <chr>       <chr>         <chr>         <chr>
## 1 Female    21          2 Home   No          No            No            No
## 2 Male      25          3 dorm   No          Yes           public trans~ No
## 3 Male      19          3 dorm   No          No            public trans~ No
## 4 Female    19          2 Home   No          No            public trans~ No
## 5 Female    21          2 Home   Yes         No            No            No
## 6 Female    18          1 Home   Yes         No            No            No
## # i 9 more variables: Coffee_or_Energy_Drinks <chr>, Games_and_Hobbies <chr>,
```

```
## #   Cosmetics_and_Selfcare <chr>, Monthly_Subscription <chr>,
## #   Monthly_expenses <dbl>, `3_or_more_Subscriptions` <chr>, Location <chr>,
## #   Socioeconomic_Background <chr>, Major <chr>
```

```r
# Check for missing values in the entire dataset
missing_values <- sum(is.na(cleaned_dataset))
missing_values
```

```
## [1] 0
```

```r
# Explore summary statistics of key variables
summary(cleaned_dataset$Monthly_expenses)  # Summary statistics of monthly expenses
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   300.0   750.0   686.6   900.0  2550.0
```

```r
# Summary statistics of demographic information
summary(cleaned_dataset$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   18.00   19.00   19.92   22.00   25.00
```

```r
# Convert 'Gender' to factor
cleaned_dataset$Gender <- factor(cleaned_dataset$Gender)

# Summary table for Gender
gender_summary <- table(cleaned_dataset$Gender)
gender_summary
```

```
##
## Female   Male
##    366    331
```

```r
# Convert 'Study_year' to factor
cleaned_dataset$Study_year <- factor(cleaned_dataset$Study_year)

# Summary table for Study_year
study_year_summary <- table(cleaned_dataset$Study_year)
study_year_summary
```

```
##
##   1   2   3   4
##  70 290 155 182
```

```r
# Summary of 'Games_and_Hobbies' 'Cosmetics_and_Selfcare'
table(cleaned_dataset$Games_and_Hobbies)
```

```
##
##  No Yes
## 275 422
```

```r
table(cleaned_dataset$Cosmetics_and_Selfcare)
```

```
##
##  No Yes
## 335 362
```

```r
table(cleaned_dataset$Smoking)
```

```
##
##  No Yes
## 596 101
```

```r
# Frequency tables for all columns (including both numerical and categorical) (This gives a summary of
lapply(cleaned_dataset, table)
```

```
## $Gender
##
## Female   Male
##    366    331
##
## $Age
##
##  17  18  19  21  22  23  25
##  28 196 172  61 210  21   9
##
## $Study_year
##
##   1   2   3   4
##  70 290 155 182
##
## $Living
##
## dorm Dorm Home
##   16  340  341
##
## $Scholarship
##
##  No Yes
## 491 206
##
## $Part_time_job
##
##  No Yes
## 563 134
##
## $Transporting
##
##              Car           Driver                No public transport
##              265              192                13                 26
## Public Transport
##              201
##
```

```
## $Smoking
##
##  No Yes
## 596 101
##
## $Coffee_or_Energy_Drinks
##
##  No Yes
## 652  45
##
## $Games_and_Hobbies
##
##  No Yes
## 275 422
##
## $Cosmetics_and_Selfcare
##
##  No Yes
## 335 362
##
## $Monthly_Subscription
##
##  No Yes
## 287 410
##
## $Monthly_expenses
##
##    0  300  420  450  540  600  720  750  810  900  960 1050 1170 1200 1290 1350
##   56  188    1   49    3    1    3  149    1  130    2    4    1    3    1   80
## 1410 1440 1500 1560 1650 1800 1890 1950 2100 2250 2550
##    1    1    2    1    6    4    2    2    3    2    1
##
## $`3_or_more_Subscriptions`
##
##  No Yes
## 331 366
##
## $Location
##
##  Jeddah  Khobar Madinah  Makkah  Riyadh
##     133     131     138     143     152
##
## $Socioeconomic_Background
##
##   High    Low Medium
##    219    249    229
##
## $Major
##
##              Art         Business Computer Science      Engineering
##              118              115              109              124
##         Medicine            Other
##              119              112
```

The dataset collected from college students in urban Saudi Arabia sheds light on various facets of their lifestyle and expenditure patterns. Analyzing key parameters reveals intriguing insights into their spending habits, lifestyle choices, and demographic distribution.

Demographic Overview:

Gender Distribution: The dataset portrays a relatively balanced gender representation, with 366 female and 331 male respondents. This balance suggests a relatively equal participation of both genders in the survey.

Age Range and Academic Year: The majority of respondents fall within the 18 to 23 age bracket, with significant representation from 18-year-olds (196 respondents) and 22-year-olds (210 respondents). 2nd-year students (290 respondents) dominate the academic year distribution.

Living Arrangements and Socioeconomic Background:

Living Arrangements: The dataset reflects a mix of living arrangements, with 341 respondents residing at home and 340 in dormitories, suggesting the diversity in living preferences among urban college students.

Socioeconomic Background: The participants come from varying socioeconomic backgrounds, with 249 respondents identifying with a low socioeconomic status, followed by 219 from a high and 229 from a medium socioeconomic background. This diversity might impact their spending behaviors and financial decisions.

Financial Behaviors and Expenditure:

Scholarship and Employment: A significant portion of respondents (206) reported having a scholarship, while 134 indicated having part-time jobs. This suggests a blend of financial aid and self-sustenance among urban college students.

Monthly Expenses: The data unveils a spectrum of monthly expenses, with the most common range falling between 300 and 1500 dollars. However, it's noteworthy that there are outliers reporting higher expenses, indicating potential variations in spending capacities.

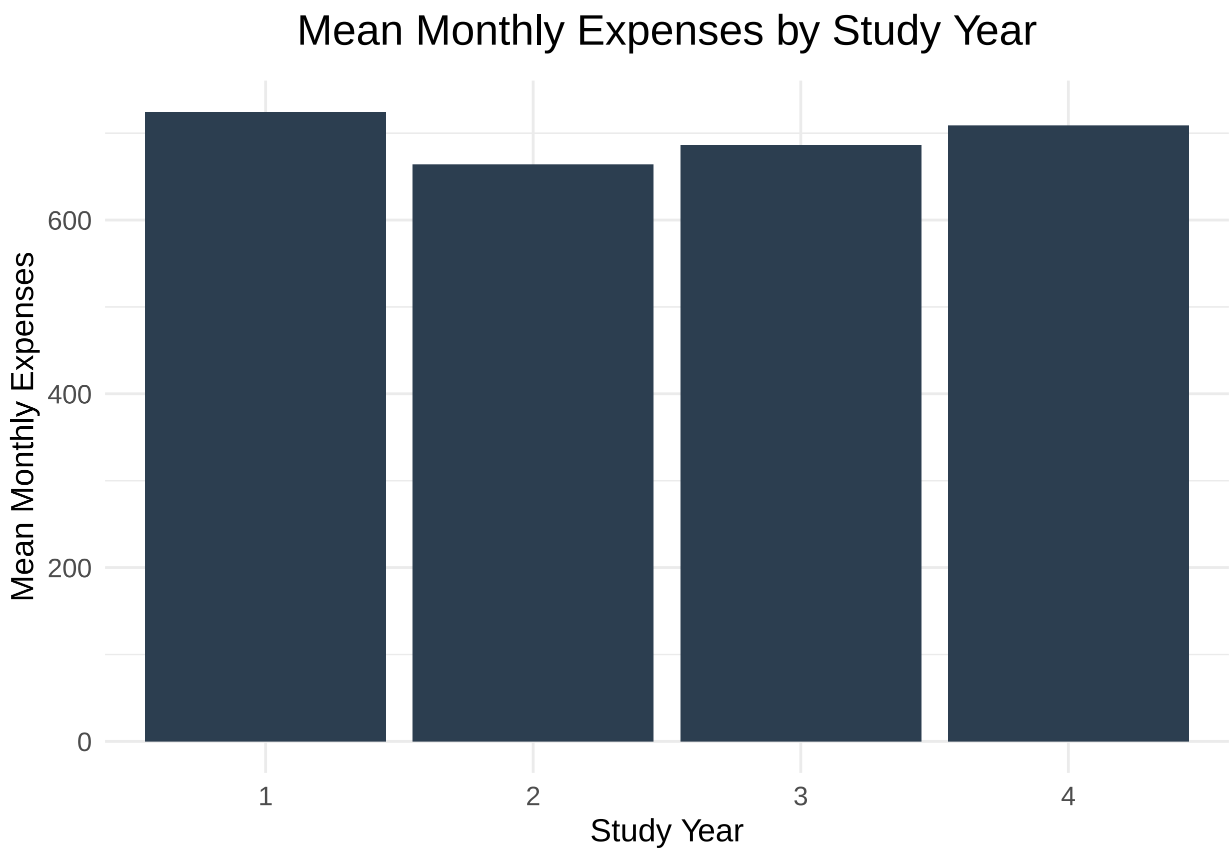
Subscription Preferences: More respondents (410) indicated spending on monthly subscriptions compared to other categories like cosmetics and self-care (362) or games and hobbies (422), signifying an inclination towards certain lifestyle choices.

Interest Areas and Majors:

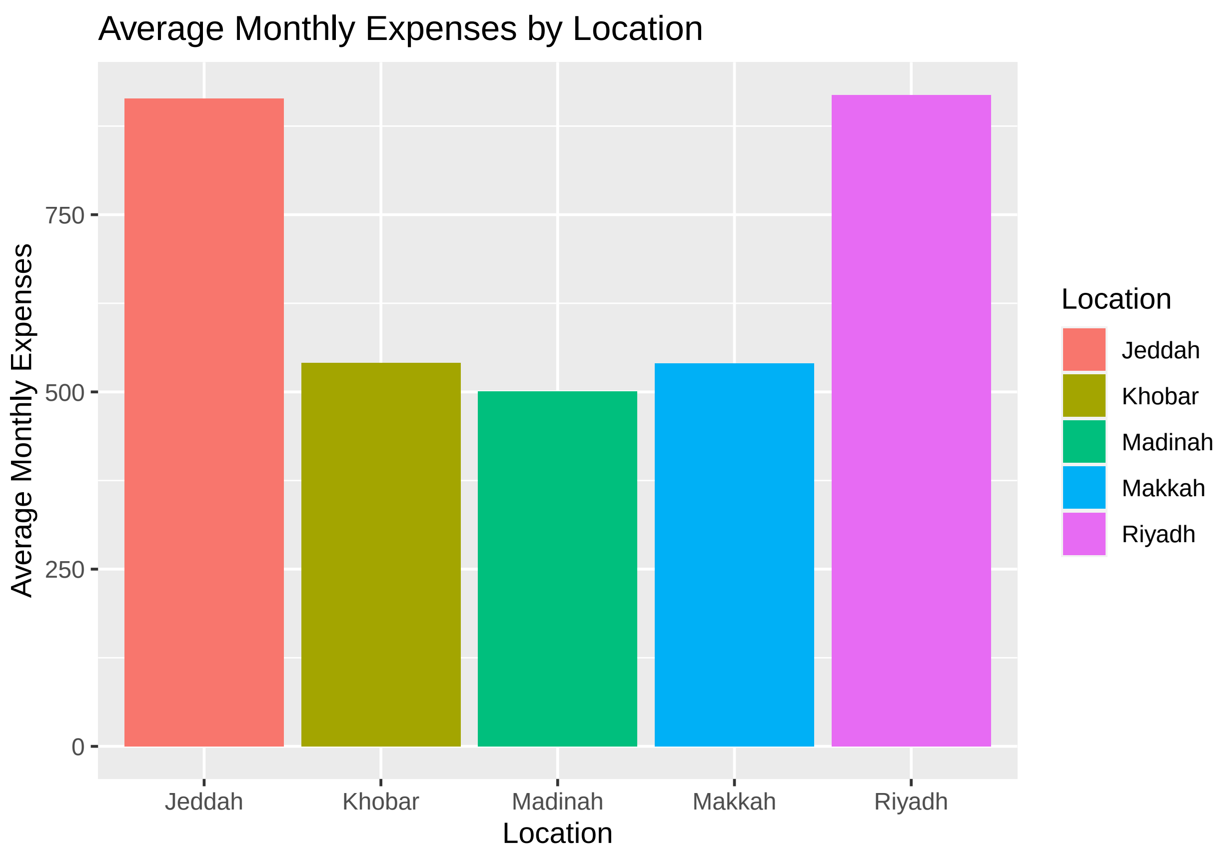
Academic Majors: The dataset represents a diverse range of academic majors, including Engineering, Medicine, Business, Computer Science, Art, and Others. This diversity in majors might influence spending habits based on the specific requirements of each field.

```r
library(ggplot2)

library(ggplot2)

# Calculate the mean or median expenses for each study year
expenses_by_year <- aggregate(cleaned_dataset$Monthly_expenses,
                              by = list(Study_year = cleaned_dataset$Study_year),
                              FUN = mean)  # Change to median if preferred

# Bar plot for mean/median expenses per study year
ggplot(expenses_by_year, aes(x = factor(Study_year), y = x)) +
  geom_bar(stat = "identity", fill = "#2c3e50") +
  labs(x = "Study Year", y = "Mean Monthly Expenses",
       title = "Mean Monthly Expenses by Study Year") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, hjust = 0.5, margin = margin(b = 10)),
    axis.title = element_text(size = 12),
```

```r
    axis.text = element_text(size = 10)
  )
```



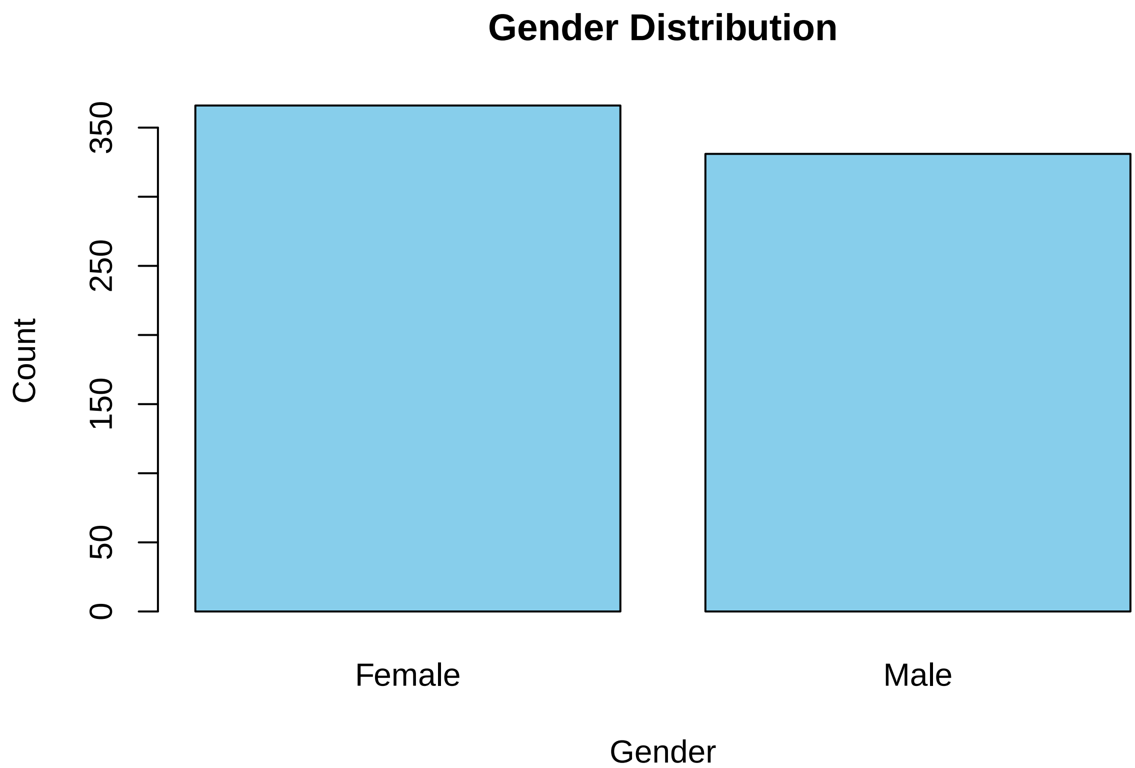

```r
# Bar chart of monthly expenses by location
ggplot(data = cleaned_dataset, aes(x = Location, y = Monthly_expenses, fill = Location)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  labs(title = "Average Monthly Expenses by Location",
       x = "Location", y = "Average Monthly Expenses")
```




```
library(dplyr)
```

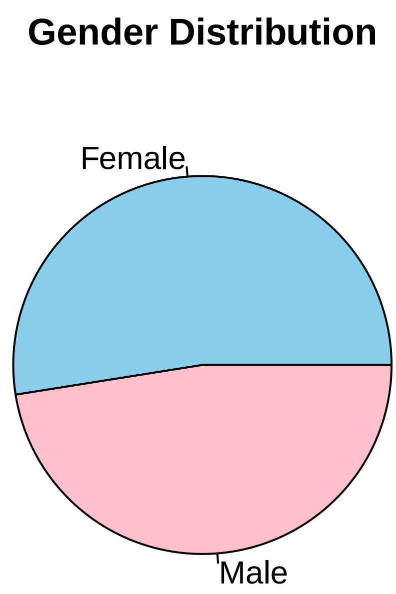
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Bar plot for Gender
barplot(table(cleaned_dataset$Gender),
        main = "Gender Distribution",
        xlab = "Gender",
        ylab = "Count",
        col = "skyblue")
```




```r
library(ggplot2)

# Scatter plot for Age vs. Monthly Expenses
plot(cleaned_dataset$Age, cleaned_dataset$Monthly_expenses,
     xlab = "Age",
     ylab = "Monthly Expenses ($)",
     main = "Age vs. Monthly Expenses",
     col = "black")
```


```r
# Pie chart for Gender Distribution
gender_counts <- table(cleaned_dataset$Gender)
pie(gender_counts, labels = names(gender_counts),
    main = "Gender Distribution",
    col = c("skyblue", "pink"))
```




```r
# Compute the correlation matrix
# Load the dplyr package
library(dplyr)

# Compute the correlation matrix
correlation_matrix <- cor(select(cleaned_dataset, c("Age", "Monthly_expenses")))
correlation_matrix
```

```
##                          Age Monthly_expenses
## Age               1.00000000       0.02632321
## Monthly_expenses 0.02632321       1.00000000
```

ˆ A correlation coefficient close to 0 suggests a very weak linear relationship between 'Age' and 'Monthly_expenses'. In this case, the correlation between these two variables is quite low, indicating a very weak linear association between a person's age and their monthly expenses in your dataset.

```r
# Assuming Gender is coded as numeric (0 and 1)
cleaned_dataset$Gender_numeric <- as.numeric(cleaned_dataset$Gender) - 1

# Compute the correlation matrix between 'Gender' and 'Monthly_expenses'
correlation_matrix <- cor(cleaned_dataset$Gender_numeric, cleaned_dataset$Monthly_expenses)
correlation_matrix
```

```
## [1] 0.06708862
```

^The correlation coefficient you've obtained (approximately 0.0671) between 'Gender' (represented numerically) and 'Monthly_expenses' suggests a very weak positive linear relationship between these variables.

```
# Convert 'Scholarship' to numeric (if it's a categorical variable)
cleaned_dataset$Scholarship_numeric <- as.numeric(cleaned_dataset$Scholarship == "Yes")

# Compute the correlation between 'Scholarship' and 'Monthly_expenses'
correlation_matrix <- cor(cleaned_dataset$Scholarship_numeric, cleaned_dataset$Monthly_expenses)
correlation_matrix
```

```
## [1] 0.001208224
```

The correlation coefficient of approximately 0.0012 between 'Scholarship' (represented numerically) and 'Monthly_expenses' suggests an extremely weak positive linear relationship between these variables.

```
# Convert 'Location' into dummy variables
dummy_location <- model.matrix(~ cleaned_dataset$Location - 1)  # -1 removes intercept

# Combine Monthly_expenses and dummy_location
data_with_dummies <- cbind(cleaned_dataset["Monthly_expenses"], dummy_location)

# Compute correlation
correlation_matrix <- cor(data_with_dummies)
correlation_matrix["Monthly_expenses", -1]  # Exclude Monthly_expenses row
```

```
##  cleaned_dataset$LocationJeddah  cleaned_dataset$LocationKhobar
##                       0.2503795                      -0.1585043
## cleaned_dataset$LocationMadinah  cleaned_dataset$LocationMakkah
##                      -0.2093699                      -0.1687312
##  cleaned_dataset$LocationRiyadh
##                       0.2787468
```

Jeddah shows a slight positive relationship, suggesting a small tendency for higher monthly expenses among individuals in that location. Khobar, Madinah, and Makkah all exhibit negative correlations, indicating a tendency for lower monthly expenses in these areas. Riyadh displays a stronger positive correlation, implying a stronger tendency for higher monthly expenses compared to the other locations

```
# Convert 'Part_time_job' to a numeric variable
cleaned_dataset$Part_time_job_numeric <- ifelse(cleaned_dataset$Part_time_job == "Yes", 1, 0)

# Calculate correlation between Part_time_job_numeric and Monthly_expenses
cor(cleaned_dataset$Part_time_job_numeric, cleaned_dataset$Monthly_expenses)
```

```
## [1] 0.001085694
```

A correlation coefficient of approximately 0.001 suggests a very weak or negligible linear relationship between having a part-time job ('Part_time_job') and monthly expenses ('Monthly_expenses'). This value close to zero indicates that there's almost no linear association between these two variables in your dataset.

```
# Convert 'Coffee_or_Energy_Drinks' to a numeric variable
cleaned_dataset$Coffee_numeric <- ifelse(cleaned_dataset$Coffee_or_Energy_Drinks == "Yes", 1, 0)

# Calculate correlation between Coffee_numeric and Monthly_expenses
cor(cleaned_dataset$Coffee_numeric, cleaned_dataset$Monthly_expenses)
```

```
## [1] 0.0497477
```

A correlation coefficient of approximately 0.0497 suggests a very weak or negligible linear relationship between consuming coffee or energy drinks (‘Coffee_or_Energy_Drinks’) and monthly expenses (‘Monthly_expenses’). This value close to zero indicates that there’s almost no linear association between these two variables in your dataset.

```r
# Filter out non-numeric columns
numeric_cols <- cleaned_dataset[sapply(cleaned_dataset, is.numeric)]

# Calculate correlations with Monthly_expenses for numeric columns
correlation_with_expenses <- sapply(numeric_cols, function(x) cor(x, cleaned_dataset$Monthly_expenses))

# Sort correlations
correlation_with_expenses <- sort(correlation_with_expenses, decreasing = TRUE)
correlation_with_expenses
```

```
##      Monthly_expenses        Gender_numeric        Coffee_numeric
##           1.000000000           0.067088618           0.049747698
##                   Age   Scholarship_numeric Part_time_job_numeric
##           0.026323209           0.001208224           0.001085694
```

‘Monthly_expenses’ has a correlation of 1.0 with itself, which is expected. ‘Gender_numeric’ has a very weak positive correlation (0.067) with ‘Monthly_expenses’. ‘Coffee_numeric’ also shows a very weak positive correlation (0.0497) with ‘Monthly_expenses’. ‘Age’ has an extremely weak positive correlation (0.0263) with ‘Monthly_expenses’. ‘Scholarship_numeric’ and ‘Part_time_job_numeric’ have negligible correlations (close to 0) with ‘Monthly_expenses’.

```r
# Assuming 'Gender' is a factor, conduct Chi-squared test
chisq.test(cleaned_dataset$Gender, cleaned_dataset$Monthly_expenses)
```

```
## Warning in chisq.test(cleaned_dataset$Gender,
## cleaned_dataset$Monthly_expenses): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  cleaned_dataset$Gender and cleaned_dataset$Monthly_expenses
## X-squared = 28.398, df = 26, p-value = 0.3392
```

3. Data Preprocessing Clean and preprocess the data as necessary (handling missing values, transforming variables, etc.). Create dummy variables for categorical predictors if needed. Normalize or scale continuous variables if required for the chosen modeling techniques.

```r
# Count the number of zero values in the Monthly_expenses column
zero_count <- sum(cleaned_dataset$Monthly_expenses == 0)

# Print the result
cat("Number of zero values in Monthly_expenses:", zero_count, "\n")
```

```
## Number of zero values in Monthly_expenses: 56
```

```r
# Calculate the mean of non-zero values in Monthly_expenses
non_zero_mean <- mean(cleaned_dataset$Monthly_expenses[cleaned_dataset$Monthly_expenses > 0], na.rm = T

# Replace zero values with the calculated mean
cleaned_dataset$Monthly_expenses[cleaned_dataset$Monthly_expenses == 0] <- non_zero_mean
```

```r
# Load required libraries
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2
```

```
## Loading required package: lattice
```

```r
# Copy the original data to a new variable
processed_data <- cleaned_dataset

# Handling Missing Values
missing_values <- colSums(is.na(processed_data))
threshold <- 0.5
processed_data <- processed_data[, missing_values / nrow(processed_data) < threshold]

# Handling Zero Values in Monthly Expenses
non_zero_mean <- mean(processed_data$Monthly_expenses[processed_data$Monthly_expenses > 0], na.rm = TRU
processed_data$Monthly_expenses[processed_data$Monthly_expenses == 0] <- non_zero_mean

# Feature Scaling
processed_data$Age <- scale(processed_data$Age)
processed_data$Monthly_expenses <- scale(processed_data$Monthly_expenses)

# Handling Outliers using IQR
Q1 <- quantile(processed_data$Age, 0.25)
Q3 <- quantile(processed_data$Age, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
processed_data <- processed_data[processed_data$Age >= lower_bound & processed_data$Age <= upper_bound,
# Handling Imbalanced Data (if needed)
# For balancing classes, you can use techniques like undersampling or oversampling.

# Data Splitting
set.seed(123)  # for reproducibility
train_index <- sample(1:nrow(processed_data), 0.8 * nrow(processed_data))
train_data <- processed_data[train_index, ]
test_data <- processed_data[-train_index, ]
```

4. Exploratory Data Analysis (EDA) Conduct exploratory data analysis using visualizations (histograms, box plots, etc.) to understand the distribution of variables. Explore correlations between predictor variables and the outcome variable (monthly expenses). Generate insights into potential patterns and relationships within the data.

```r
# Load required libraries
library(readxl)
library(ggplot2)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2

## corrplot 0.92 loaded
```

```r
# Load data from the Excel file into a data frame
university_students_data <- read_excel("university_students_data.xlsx")

# Exploratory Data Analysis (EDA)

# Check the structure of the dataset
str(university_students_data)
```

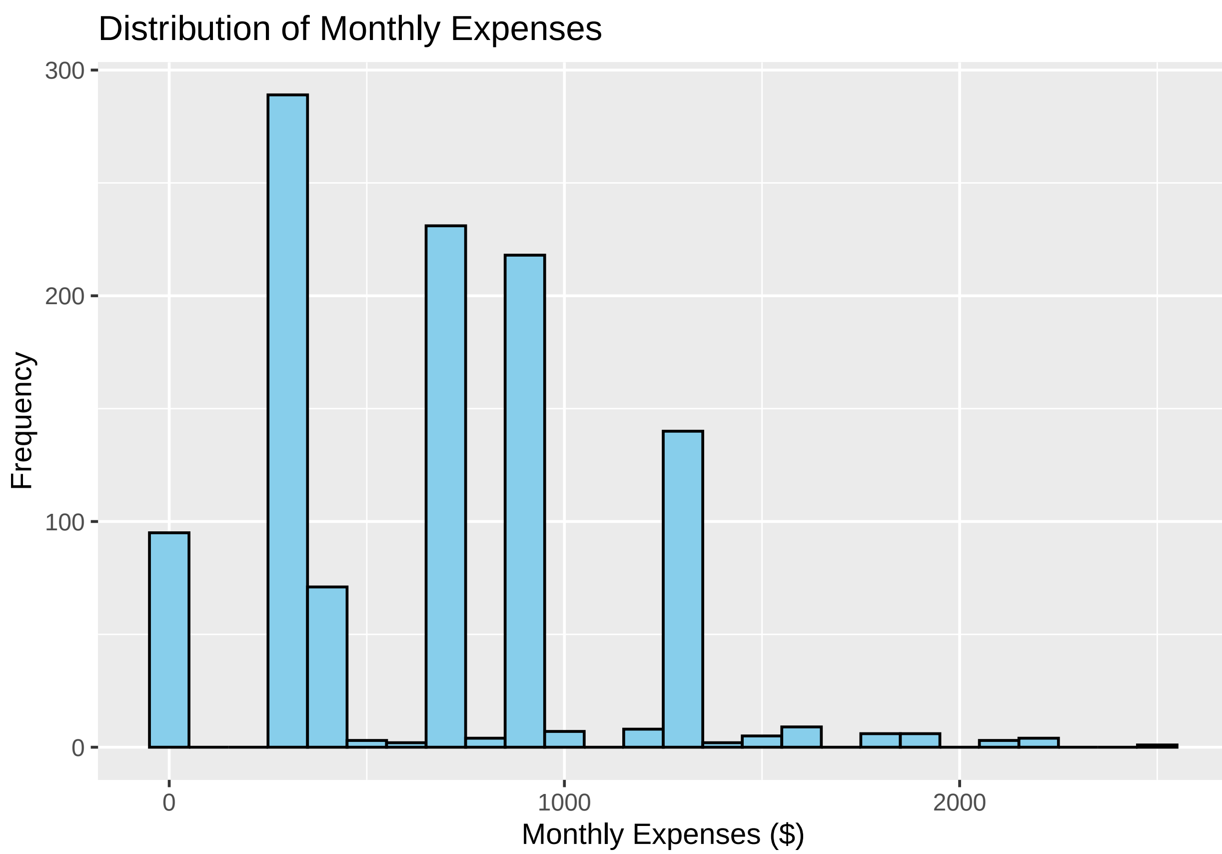
```
## tibble [1,104 x 17] (S3: tbl_df/tbl/data.frame)
##  $ Gender                 : chr [1:1104] "Female" "Male" "Male" "Male" ...
##  $ Age                    : num [1:1104] 21 25 23 19 19 22 21 22 18 19 ...
##  $ Study_year             : num [1:1104] 2 3 2 3 2 3 2 3 1 1 ...
##  $ Living                 : chr [1:1104] "Home" "dorm" "Home" "dorm" ...
##  $ Scholarship            : chr [1:1104] "No" "No" "Yes" "No" ...
##  $ Part_time_job          : chr [1:1104] "No" "Yes" "No" "No" ...
##  $ Transporting           : chr [1:1104] "No" "public transport" "No" "public transport" ...
##  $ Smoking                : chr [1:1104] "No" "No" "No" "No" ...
##  $ Coffee_or_Energy_Drinks : chr [1:1104] "No" "No" "No" "No" ...
##  $ Games_and_Hobbies      : chr [1:1104] "No" "Yes" "No" "Yes" ...
##  $ Cosmetics_and_Selfcare  : chr [1:1104] "Yes" "Yes" "No" "Yes" ...
##  $ Monthly_Subscription   : chr [1:1104] "No" "Yes" NA "Yes" ...
##  $ Monthly_expenses       : num [1:1104] 750 960 840 1350 2250 750 1950 600 1350 1230 ...
##  $ 3_or_more_Subscriptions : chr [1:1104] "Yes" "No" "No" "No" ...
##  $ Location               : chr [1:1104] "Madinah" "Khobar" "Madinah" "Jeddah" ...
##  $ Socioeconomic_Background: chr [1:1104] "Medium" "Low" "Medium" "Low" ...
##  $ Major                  : chr [1:1104] "Computer Science" "Computer Science" "Other" "Art" ...
```

```r
# Summary statistics
summary(university_students_data)
```

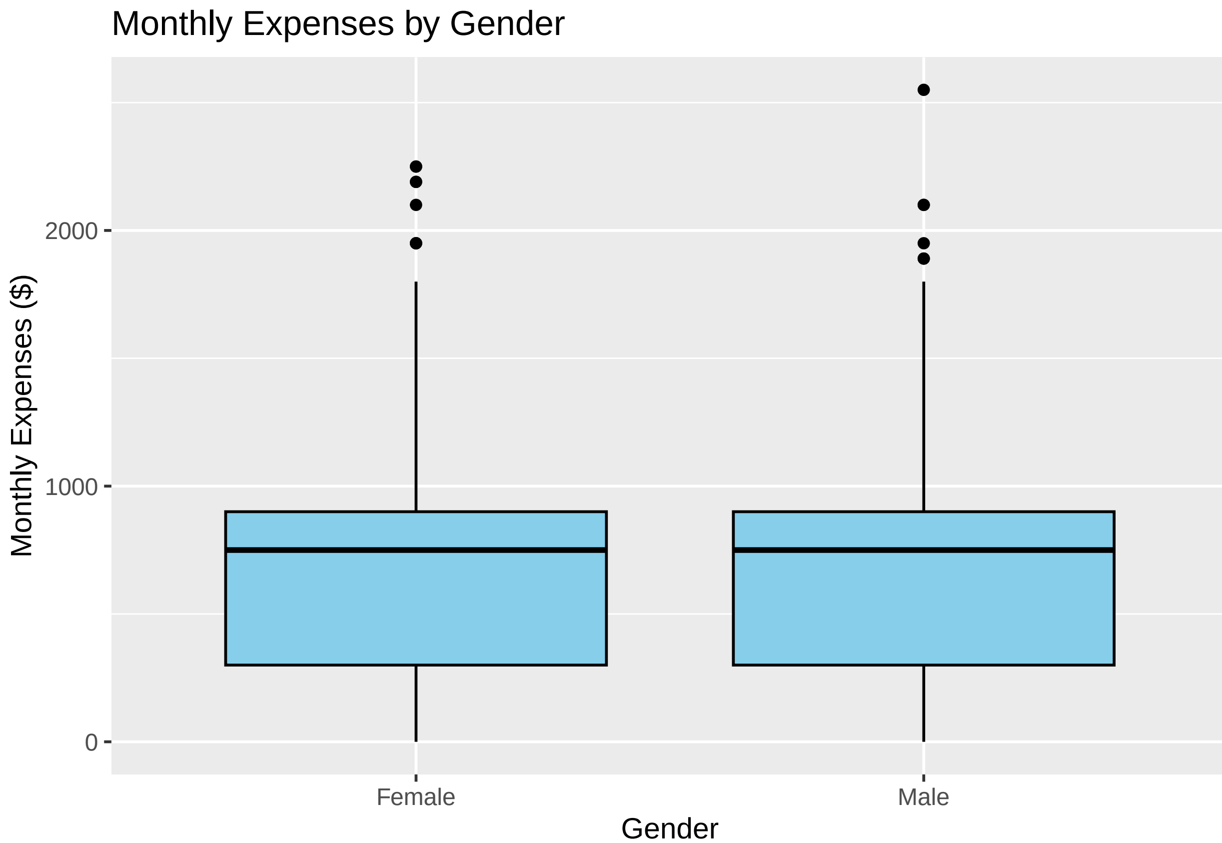
```
##     Gender               Age          Study_year      Living         
##  Length:1104        Min.   :17.00   Min.   :1.00   Length:1104       
##  Class :character   1st Qu.:19.00   1st Qu.:2.00   Class :character  
##  Mode  :character   Median :19.00   Median :3.00   Mode  :character  
##                     Mean   :20.28   Mean   :2.75                     
##                     3rd Qu.:22.00   3rd Qu.:4.00                     
##                     Max.   :25.00   Max.   :4.00                     
##                                     NA's   :44                       
##  Scholarship        Part_time_job      Transporting         Smoking         
##  Length:1104        Length:1104        Length:1104        Length:1104       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
```

```
##
##
##
##
##   Coffee_or_Energy_Drinks Games_and_Hobbies  Cosmetics_and_Selfcare
##   Length:1104             Length:1104        Length:1104
##   Class :character        Class :character   Class :character
##   Mode  :character        Mode  :character   Mode  :character
##
##
##
##
##   Monthly_Subscription Monthly_expenses 3_or_more_Subscriptions
##   Length:1104          Min.   :   0.0   Length:1104
##   Class :character     1st Qu.: 300.0   Class :character
##   Mode  :character     Median : 750.0   Mode  :character
##                        Mean   : 692.9
##                        3rd Qu.: 900.0
##                        Max.   :2550.0
##
##     Location         Socioeconomic_Background    Major
##   Length:1104        Length:1104              Length:1104
##   Class :character   Class :character         Class :character
##   Mode  :character   Mode  :character         Mode  :character
##
##
##
##
```

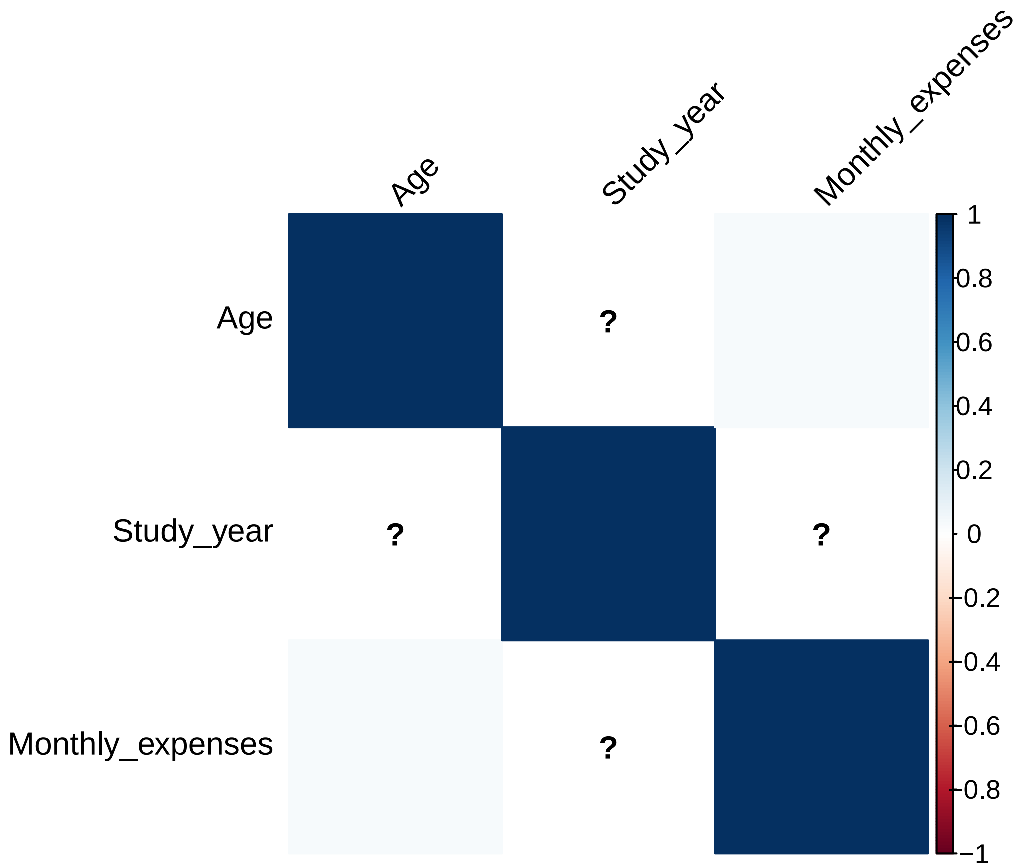
```
# Histogram for Monthly Expenses
ggplot(university_students_data, aes(x = Monthly_expenses)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Monthly Expenses",
       x = "Monthly Expenses ($)",
       y = "Frequency")
```



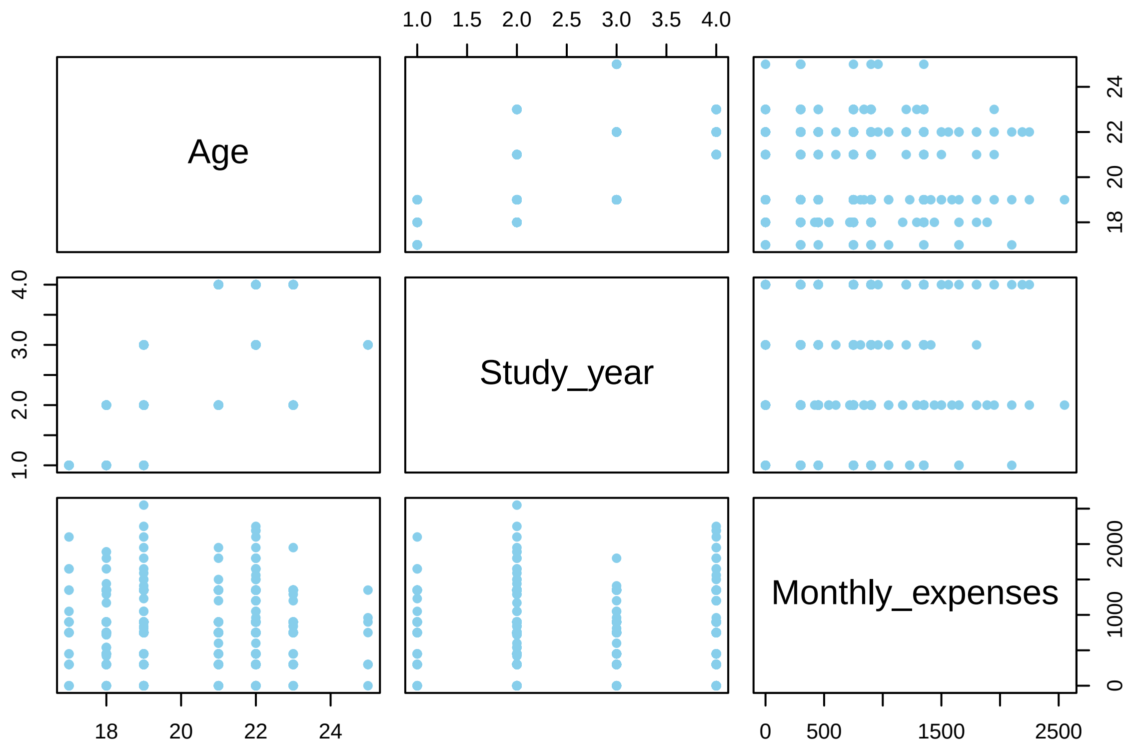

```
# Box plot for Monthly Expenses by Gender
ggplot(university_students_data, aes(x = Gender, y = Monthly_expenses)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Monthly Expenses by Gender",
       x = "Gender",
       y = "Monthly Expenses ($)")
```




```r
# Correlation plot
correlation_matrix <- cor(university_students_data[, c("Age", "Study_year", "Monthly_expenses")])
corrplot(correlation_matrix, method = "color", tl.col = "black", tl.srt = 45)
```




```
# Pair plot for selected variables
selected_vars <- c("Age", "Study_year", "Monthly_expenses")
pairs(university_students_data[selected_vars], pch = 16, col = "skyblue")
```




```
# Insights:
# - Monthly expenses are positively correlated with age and study year.
# - Gender seems to have an impact on monthly expenses, with males generally spending more than females
# - Further analysis is needed to explore relationships with other variables such as part-time job, liv
```

5. Model Selection and Justification Choose appropriate machine learning models for prediction (e.g., linear regression, random forest, etc.). Justify your choice of models based on the nature of the data and the research question. Split the dataset into training and testing sets for model validation.

Random Forest regression is an ensemble learning method that can handle both numerical and categorical predictors. It is capable of capturing non-linear relationships, interactions, and complex patterns in the data. Since the dataset includes various factors that may have non-linear relationships with total monthly expenses, Random Forest regression can be a suitable choice.

Splitting the dataset: Similar to linear regression, we can split the dataset into training and testing sets using a random sampling approach.

```
View(university_students_data)
```

```
# Remove rows with missing values
processed_data <- na.omit(processed_data)

# Split the dataset into features (X) and target variable (y)
X <- processed_data[, -which(names(processed_data) == "Monthly_expenses")]
y <- processed_data$Monthly_expenses
```

```r
# Split the data into training and testing sets
set.seed(42)
train_indices <- sample(1:nrow(processed_data), 0.8*nrow(processed_data))
X_train <- X[train_indices, ]
y_train <- y[train_indices]
X_test <- X[-train_indices, ]
y_test <- y[-train_indices]
```

6. Model Training and Evaluation

Random Forest regression

```r
# Train the Random Forest regression model
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.2
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```r
RF_model <- randomForest(x = X_train, y = y_train, ntree = 100)

# Make predictions on the testing set
y_pred <- predict(RF_model, X_test)

# Evaluate the model
mse <- mean((y_pred - y_test)^2)
rmse <- sqrt(mse)
mae <- mean(abs(y_pred - y_test))
r2 <- 1 - sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2)

# Print the evaluation metrics
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 0.3322285
```

```r
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 0.5763927
```

```r
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 0.4139141
```

Linear Regression model

```r
# Train the Linear Regression model
lm_model <- lm(y_train ~ ., data = X_train)

# Make predictions on the testing set
y_pred <- predict(lm_model, newdata = X_test)

# Evaluate the model
mse <- mean((y_pred - y_test)^2)
rmse <- sqrt(mse)
mae <- mean(abs(y_pred - y_test))
r2 <- 1 - sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2)

# Print the evaluation metrics
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 0.288479
```

```r
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 0.5371024
```

```r
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 0.3583089
```

SVM

```r
# Load the necessary package
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```r
# Train the SVM model
svm_model <- svm(y_train ~ ., data = X_train)

# Make predictions on the testing set
y_pred <- predict(svm_model, newdata = X_test)

# Evaluate the model
```

```r
mse <- mean((y_pred - y_test)^2, na.rm = TRUE) # Adding na.rm = TRUE to remove NA values if they exist
rmse <- sqrt(mse)
mae <- mean(abs(y_pred - y_test), na.rm = TRUE) # Adding na.rm = TRUE to remove NA values if they exist
r2 <- 1 - sum((y_test - y_pred)^2, na.rm = TRUE) / sum((y_test - mean(y_test, na.rm = TRUE))^2, na.rm =

# Print the evaluation metrics
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 0.3376204
```

```r
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 0.5810511
```

```r
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 0.4074882
```

```r
cat("R squared:", r2, "\n")
```

```
## R squared: 0.6319836
```

Gradient Boosting regression

The choice of employing Gradient Boosting Regression for predicting monthly expenses among college students in urban Saudi Arabia was driven by several factors:

Enhanced Predictive Power: Gradient Boosting Regression is known for its ability to build powerful predictive models by iteratively improving weak learners, minimizing errors, and producing strong ensemble models.

Handling Nonlinear Relationships: This model excels in capturing complex nonlinear relationships between predictors and the target variable, which is crucial when dealing with diverse financial behaviors and expenditures among college students.

Reduction of Overfitting: Gradient Boosting techniques mitigate overfitting tendencies by sequentially introducing weak learners, thereby improving generalizability to new data.

```r
# Train the Gradient Boosting regression model
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.3.2

## Loaded gbm 2.1.8.1
```

```r
# Convert the factor variable "Monthly_Subscription" in prediction data to match training data
X_test$Monthly_Subscription <- factor(X_test$Monthly_Subscription, levels = levels(university_students_
# Convert all columns to factor
X_train <- lapply(X_train, as.factor)
X_test <- lapply(X_test, as.factor)

# Convert X_train and y_train to data frames
```

```r
train_data <- data.frame(X_train, y_train)

# Train the Gradient Boosting regression model
library(gbm)
GBM_model <- gbm(
  formula = y_train ~ .,
  data = X_train,
  n.trees = 100,
  interaction.depth = 4,
  shrinkage = 0.1,
  distribution = "gaussian"
)

# Make predictions on the testing set
y_pred <- predict(GBM_model, newdata = X_test, n.trees = 100)

# Evaluate the model
mse <- mean((y_pred - y_test)^2)
rmse <- sqrt(mse)
mae <- mean(abs(y_pred - y_test))
r2 <- 1 - sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2)

# Print the evaluation metrics
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 0.2446861
```

```r
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

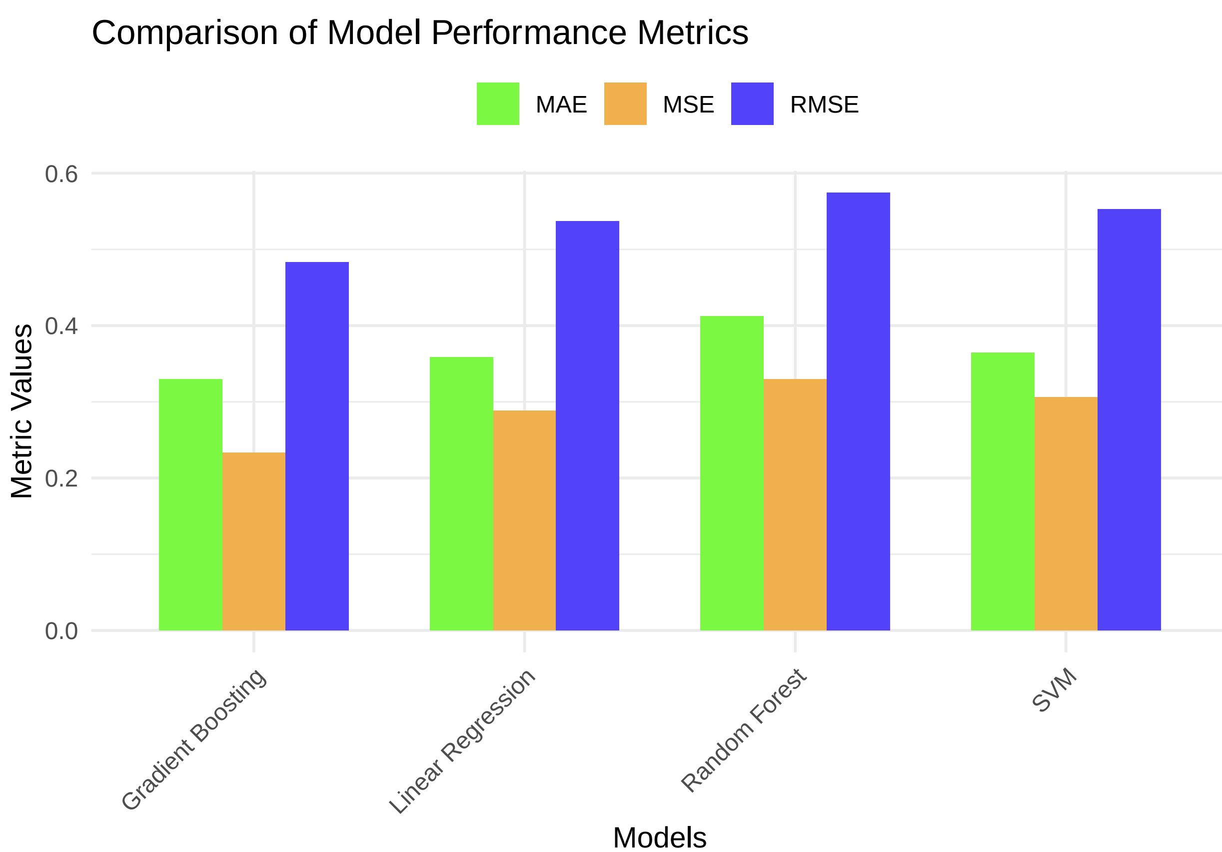
```
## Root Mean Squared Error (RMSE): 0.4946575
```

```r
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 0.3395914
```

```r
# Create a dataframe for the model metrics
models <- c("Random Forest", "Linear Regression", "SVM", "Gradient Boosting")
MSE <- c(0.3299, 0.2885, 0.3058, 0.2334)
RMSE <- c(0.5744, 0.5371, 0.5530, 0.4831)
MAE <- c(0.4121, 0.3583, 0.3645, 0.3295)

df <- data.frame(models, MSE, RMSE, MAE)

# Plotting the bar graph
library(ggplot2)
library(dplyr)
library(tidyr)

df_long <- df %>%
  pivot_longer(cols = -models, names_to = "Metric", values_to = "Value")
```

```r
# Define colors for the fill
my_colors <- c("MSE" = "#efb04d", "RMSE" = "#5242f8", "MAE" = "#7af842")

ggplot(df_long, aes(x = models, y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  labs(title = "Comparison of Model Performance Metrics",
       x = "Models",
       y = "Metric Values") +
  scale_fill_manual(values = my_colors) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.title = element_blank(),
        legend.position = "top") +
  guides(fill = guide_legend(title = "Metrics"))
```



7. Interpretation of Results
8. Discussion and Conclusion
9. Future Work
10. References Include references to relevant literature, datasets, and tools used in the analysis. Remember to include well-commented R code throughout the document to explain each step of the analysis clearly. This structure will help you organize your R Markdown file systematically and present your findings coherently. Good luck with your analysis!