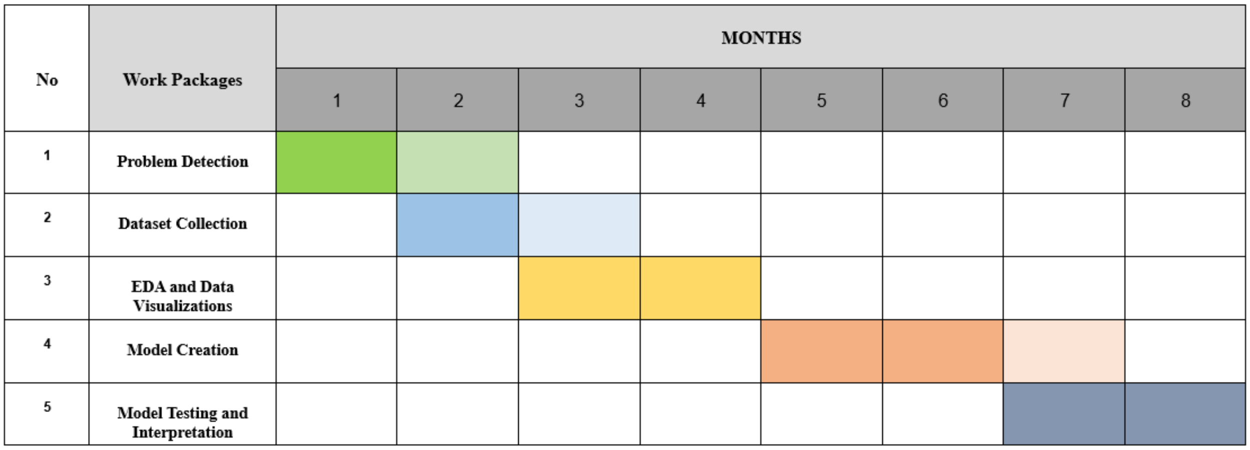
Analysis and Solution Proposals for the Obesity Issue Using Data Science Methods

2024-10-29

1-Problem Definition

Obesity has become an increasingly significant health issue today. It is directly linked to cardiovascular diseases, diabetes, and various chronic illnesses, reducing individuals' quality of life and placing a heavy financial burden on healthcare systems. Solving this issue requires evaluating various factors such as physical activity levels, eating habits, genetic factors, and lifestyle. Data science methods provide a powerful tool for analyzing these factors that contribute to obesity and developing effective solutions. Machine learning algorithms and statistical modeling techniques can be used to identify individuals at risk and create personalized health recommendations. These approaches enable the development of more effective and scientifically-based strategies in the fight against obesity.

The Gantt chart for the proposed study is as follows;



No	Work Packages	MONTHS							
		1	2	3	4	5	6	7	8
1	Problem Detection	🟩	🟩						
2	Dataset Collection		🟦	🟦					
3	EDA and Data Visualizations			🟨	🟨				
4	Model Creation					🟧	🟧	🟧	
5	Model Testing and Interpretation							🟦	🟦

2-Data Collection

A suitable dataset has been found on Kaggle for conducting a data science project related to the topic. You can access the dataset through the following link: kaggle.com/datasets/lesumitkumarroy/obesity-data-set

3-Exploratory Data Analysis and Data Visualization

```
# Load data from a CSV file
data <- read_csv("/cloud/project/ObesityDataSet_raw_and_data_sinthetic.csv")
```

```
## Rows: 2111 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): Gender, family_history_with_overweight, FAVC, CAEC, SMOKE, SCC, CAL...
## dbl (8): Age, Height, Weight, FCVC, NCP, CH2O, FAF, TUE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# View the first few rows of the dataset
head(data)
```

```
## # A tibble: 6 x 17
##   Gender   Age Height Weight family_history_with_overw~1 FAVC   FCVC   NCP CAEC 
##   <chr>  <dbl>  <dbl>  <dbl> <chr>                       <chr> <dbl> <dbl> <chr>
## 1 Female    21   1.62   64   yes                         no        2     3 Some~
## 2 Female    21   1.52   56   yes                         no        3     3 Some~
## 3 Male      23   1.8    77   yes                         no        2     3 Some~
## 4 Male      27   1.8    87   no                          no        3     3 Some~
## 5 Male      22   1.78   89.8 no                          no        2     1 Some~
## 6 Male      29   1.62   53   no                          yes       2     3 Some~
## # i abbreviated name: 1: family_history_with_overweight
## # i 8 more variables: SMOKE <chr>, CH2O <dbl>, SCC <chr>, FAF <dbl>, TUE <dbl>,
## #   CALC <chr>, MTRANS <chr>, NObeyesdad <chr>
```

```
# Check the number of rows and columns
str(data)
```

```
## spc_tbl_ [2,111 x 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Gender                        : chr [1:2111] "Female" "Female" "Male" "Male" ...
##  $ Age                           : num [1:2111] 21 21 23 27 22 29 23 22 24 22 ...
##  $ Height                        : num [1:2111] 1.62 1.52 1.8 1.8 1.78 1.62 1.5 1.64 1.78 1.72 ...
##  $ Weight                        : num [1:2111] 64 56 77 87 89.8 53 55 53 64 68 ...
##  $ family_history_with_overweight: chr [1:2111] "yes" "yes" "yes" "no" ...
##  $ FAVC                          : chr [1:2111] "no" "no" "no" "no" ...
##  $ FCVC                          : num [1:2111] 2 3 2 3 2 2 3 2 3 2 ...
##  $ NCP                           : num [1:2111] 3 3 3 3 1 3 3 3 3 3 ...
##  $ CAEC                          : chr [1:2111] "Sometimes" "Sometimes" "Sometimes" "Sometimes" ...
##  $ SMOKE                         : chr [1:2111] "no" "yes" "no" "no" ...
##  $ CH2O                          : num [1:2111] 2 3 2 2 2 2 2 2 2 2 ...
##  $ SCC                           : chr [1:2111] "no" "yes" "no" "no" ...
##  $ FAF                           : num [1:2111] 0 3 2 2 0 0 1 3 1 1 ...
##  $ TUE                           : num [1:2111] 1 0 1 0 0 0 0 0 1 1 ...
##  $ CALC                          : chr [1:2111] "no" "Sometimes" "Frequently" "Frequently" ...
##  $ MTRANS                        : chr [1:2111] "Public_Transportation" "Public_Transportation" "Publ
##  $ NObeyesdad                    : chr [1:2111] "Normal_Weight" "Normal_Weight" "Normal_Weight" "Over
##  - attr(*, "spec")=
##   .. cols(
##   ..   Gender = col_character(),
##   ..   Age = col_double(),
##   ..   Height = col_double(),
##   ..   Weight = col_double(),
##   ..   family_history_with_overweight = col_character(),
##   ..   FAVC = col_character(),
##   ..   FCVC = col_double(),
##   ..   NCP = col_double(),
##   ..   CAEC = col_character(),
##   ..   SMOKE = col_character(),
##   ..   CH2O = col_double(),
##   ..   SCC = col_character(),
##    ..   FAF = col_double(),
##    ..   TUE = col_double(),
##    ..   CALC = col_character(),
##    ..   MTRANS = col_character(),
##    ..   NObeyesdad = col_character()
##    .. )
##  - attr(*, "problems")=<externalptr>
```

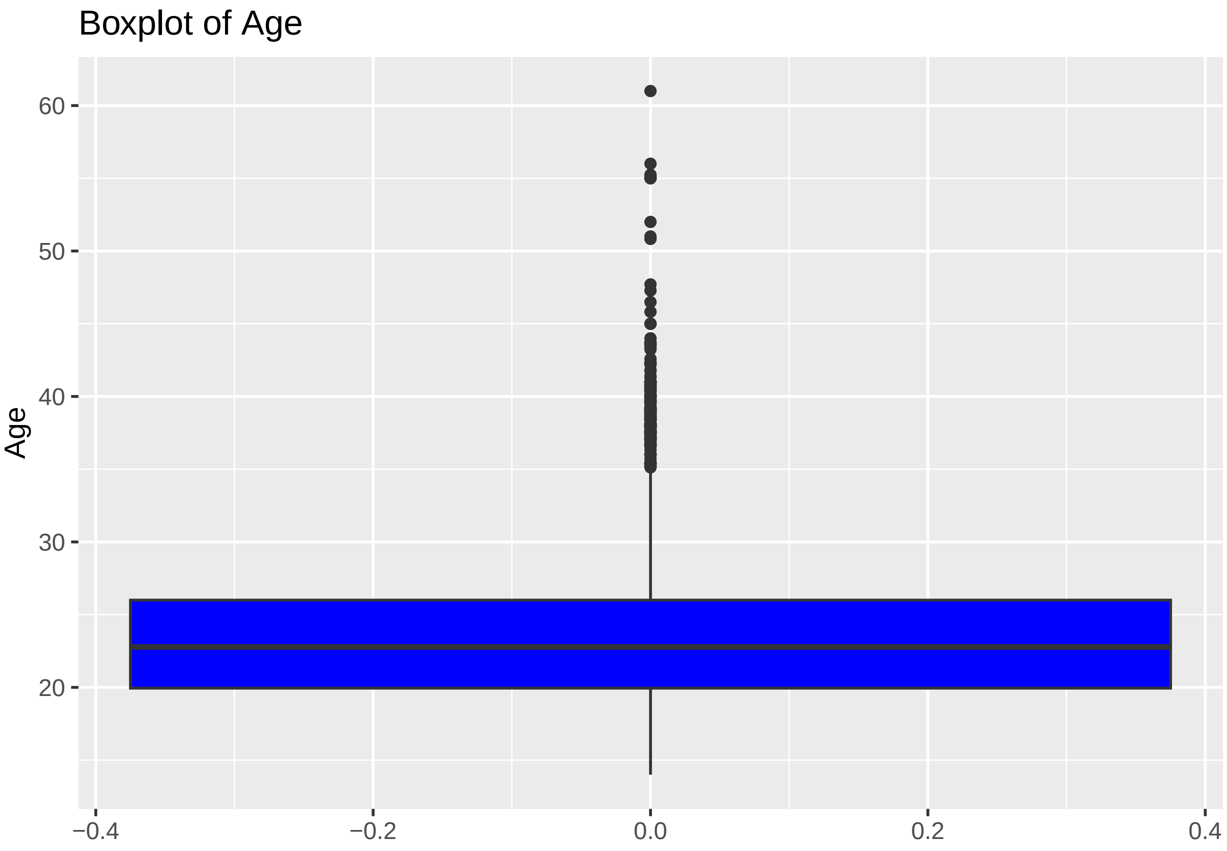
Filling missing values with the mean

```
# Veri setinde herhangi bir NA değeri olup olmadığını kontrol etme
any(is.na(data))
```

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

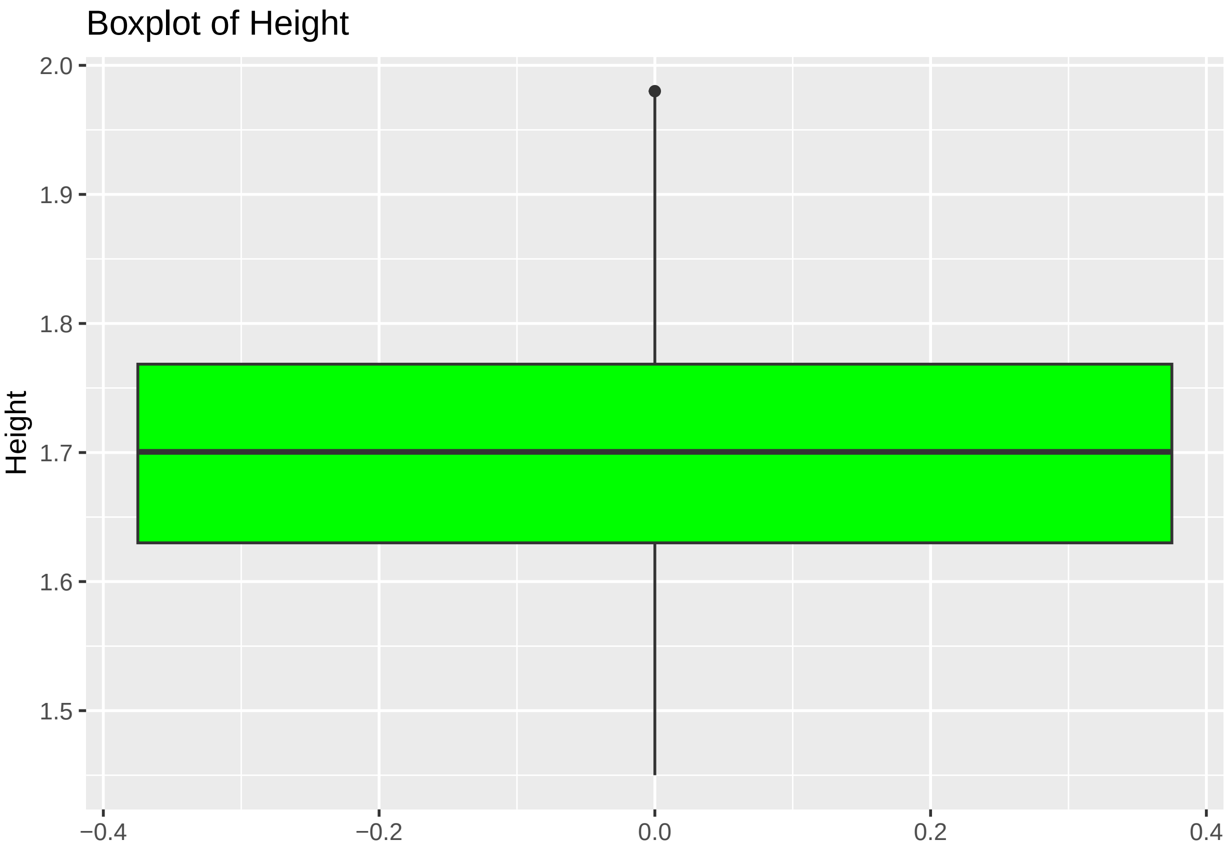
Boxplot for Age variable

```
ggplot(data, aes(y = Age)) +
  geom_boxplot(fill = "blue") +
  ggtitle("Boxplot of Age")
```



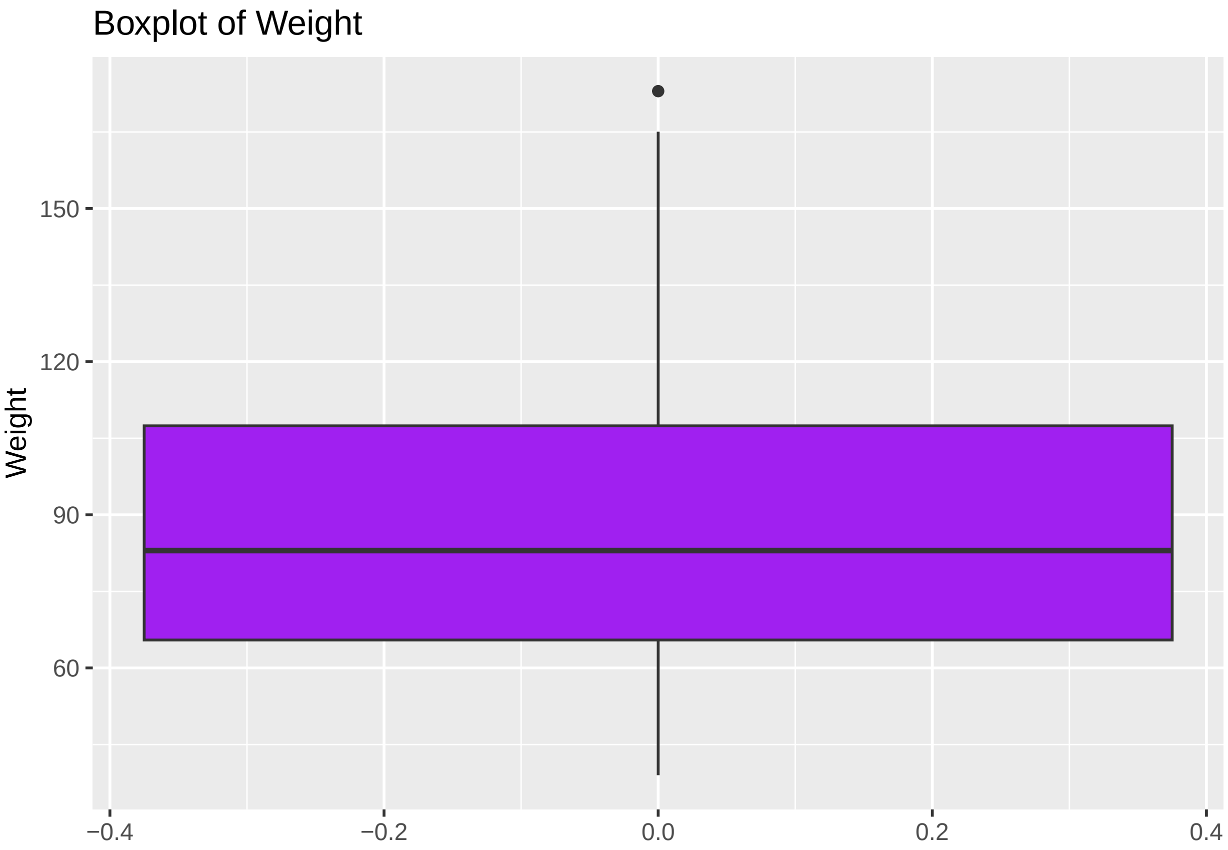
Boxplot for Height variable

```
ggplot(data, aes(y = Height)) +
  geom_boxplot(fill = "green") +
  ggtitle("Boxplot of Height")
```



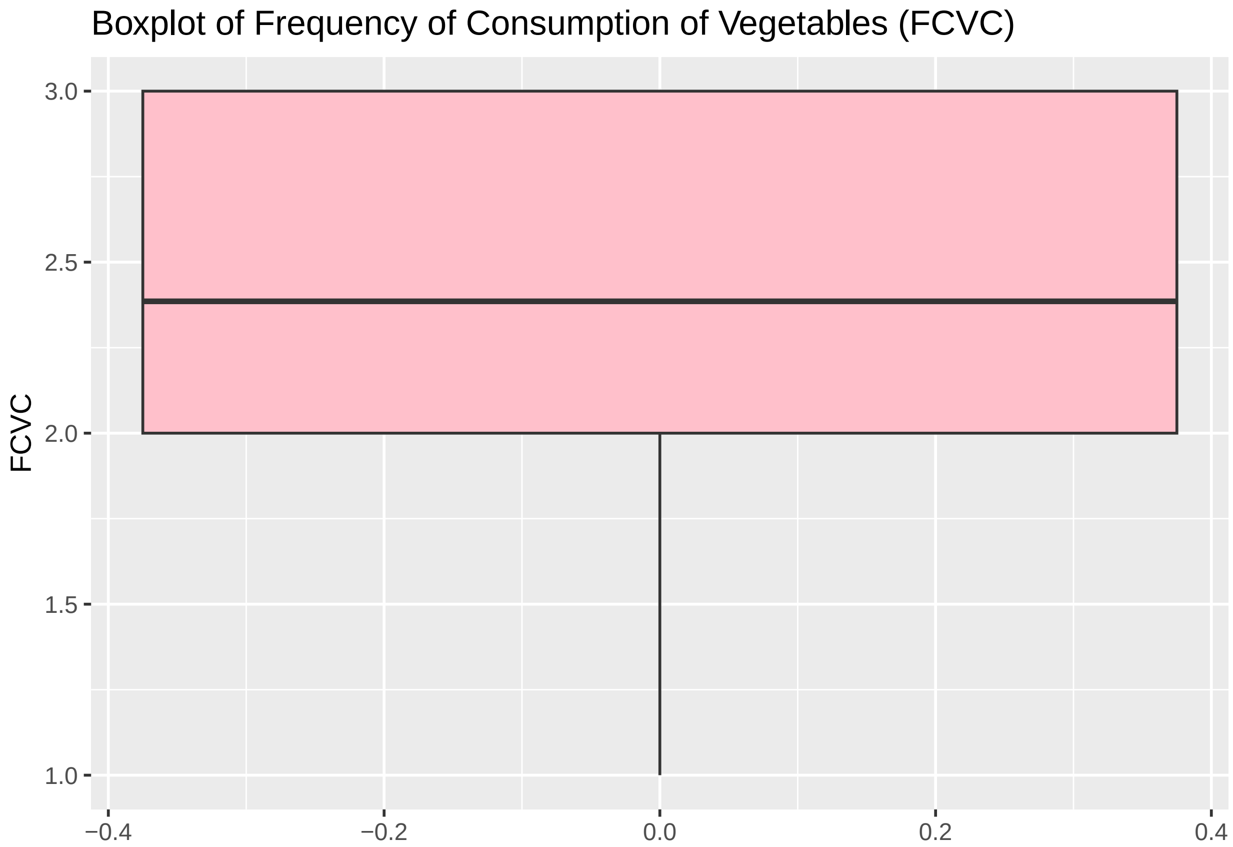
Boxplot for Weight variable

```
ggplot(data, aes(y = Weight)) +
  geom_boxplot(fill = "purple") +
  ggtitle("Boxplot of Weight")
```



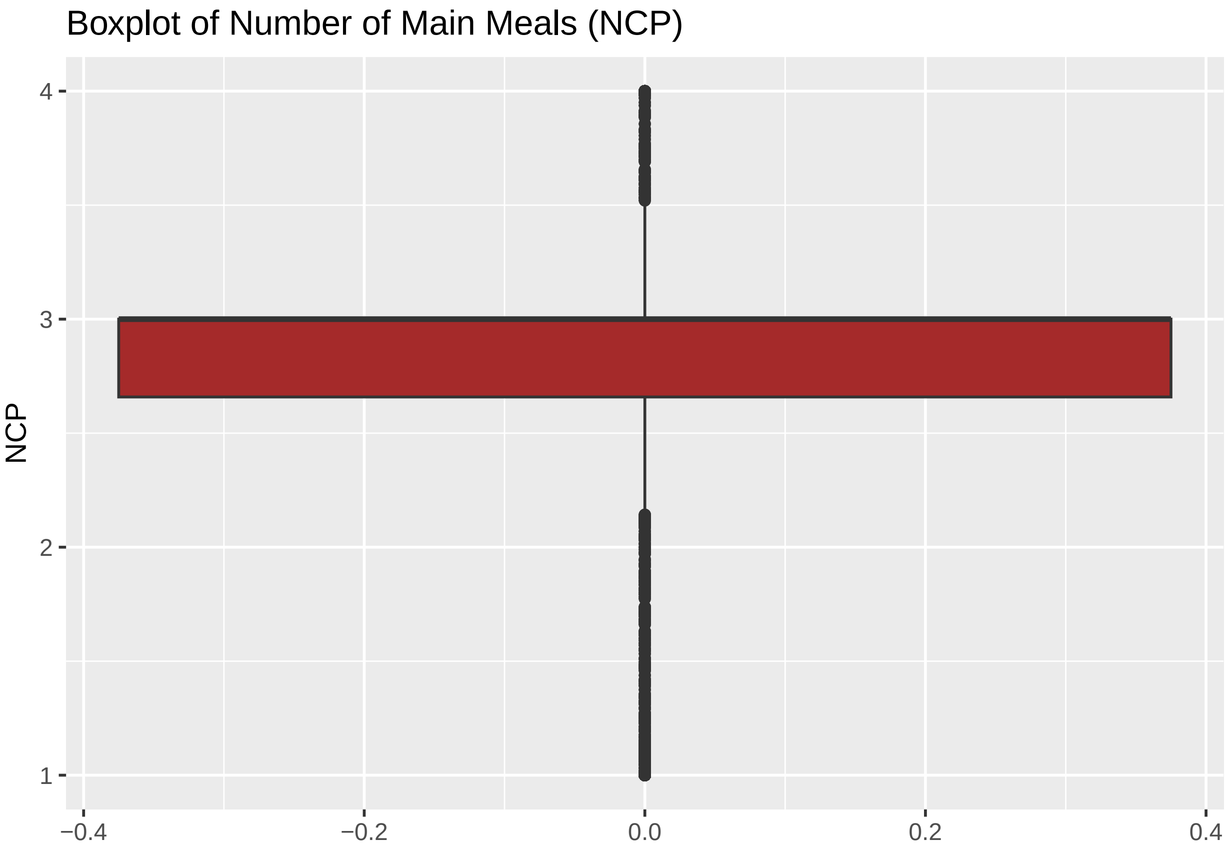
Boxplot for FCVC (Frequency of Consumption of Vegetables)

```
ggplot(data, aes(y = FCVC)) +
  geom_boxplot(fill = "pink") +
  ggtitle("Boxplot of Frequency of Consumption of Vegetables (FCVC)")
```



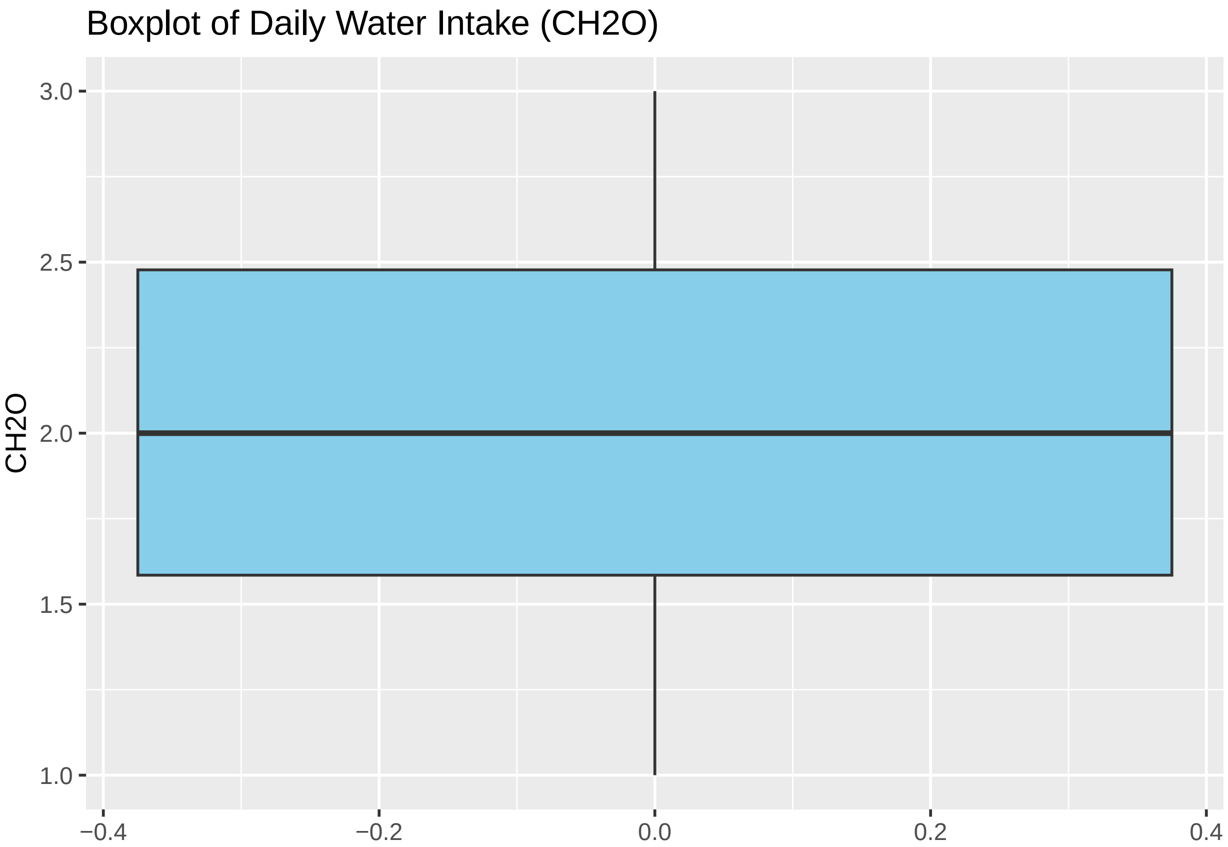
Boxplot for NCP (Number of Main Meals)

```r
ggplot(data, aes(y = NCP)) +
  geom_boxplot(fill = "brown") +
  ggtitle("Boxplot of Number of Main Meals (NCP)")
```



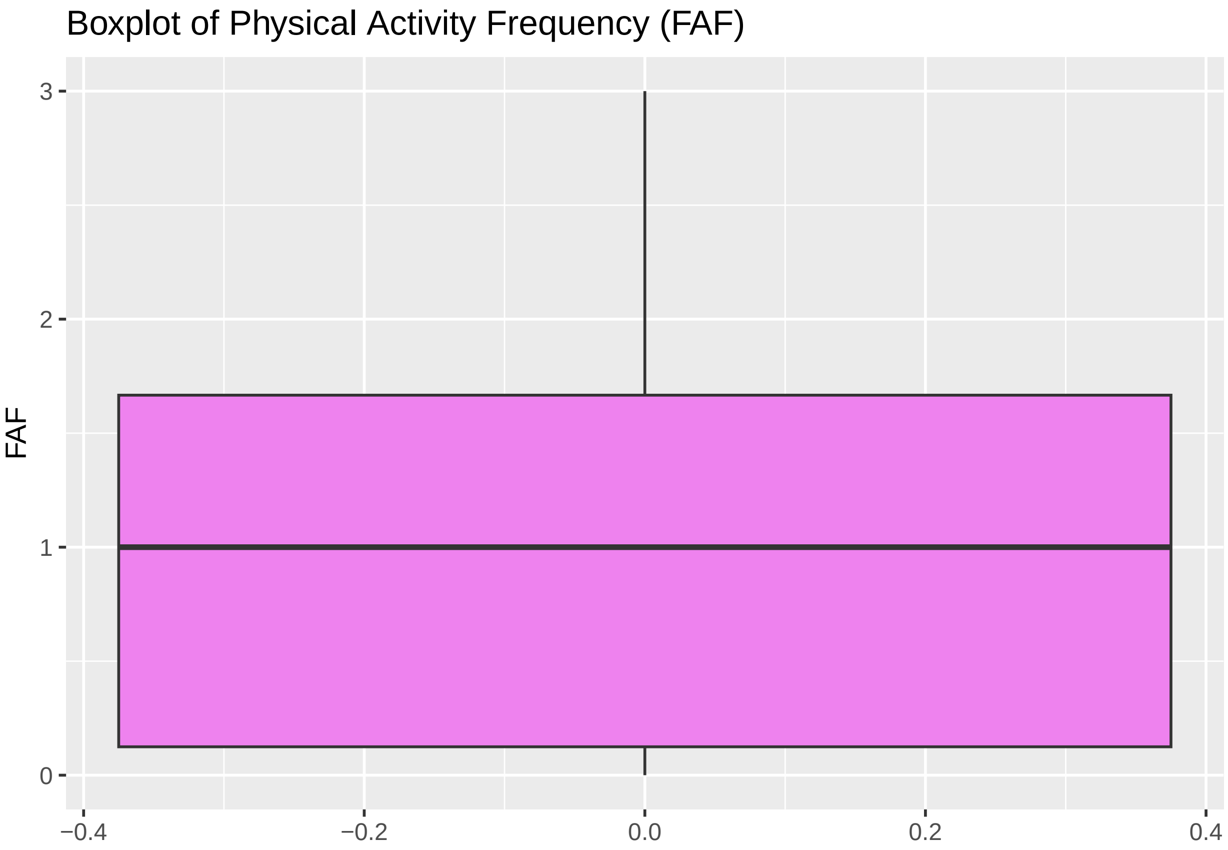
Boxplot for CH2O (Daily Water Intake)

```r
ggplot(data, aes(y = CH2O)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Daily Water Intake (CH2O)")
```



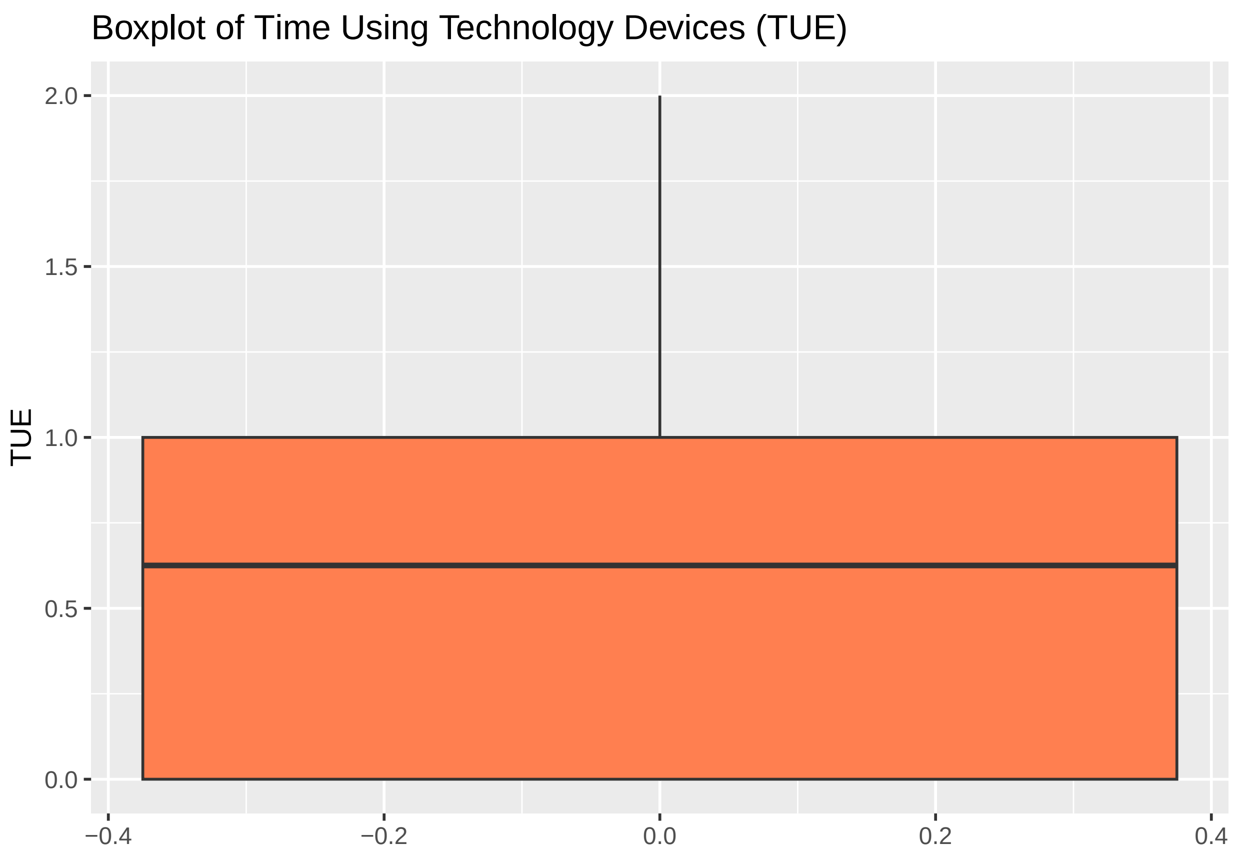
Boxplot for FAF (Physical Activity Frequency)

```r
ggplot(data, aes(y = FAF)) +
  geom_boxplot(fill = "violet") +
  ggtitle("Boxplot of Physical Activity Frequency (FAF)")
```



Boxplot for TUE (Time Using Technology Devices)

```
ggplot(data, aes(y = TUE)) +
  geom_boxplot(fill = "coral") +
  ggtitle("Boxplot of Time Using Technology Devices (TUE)")
```



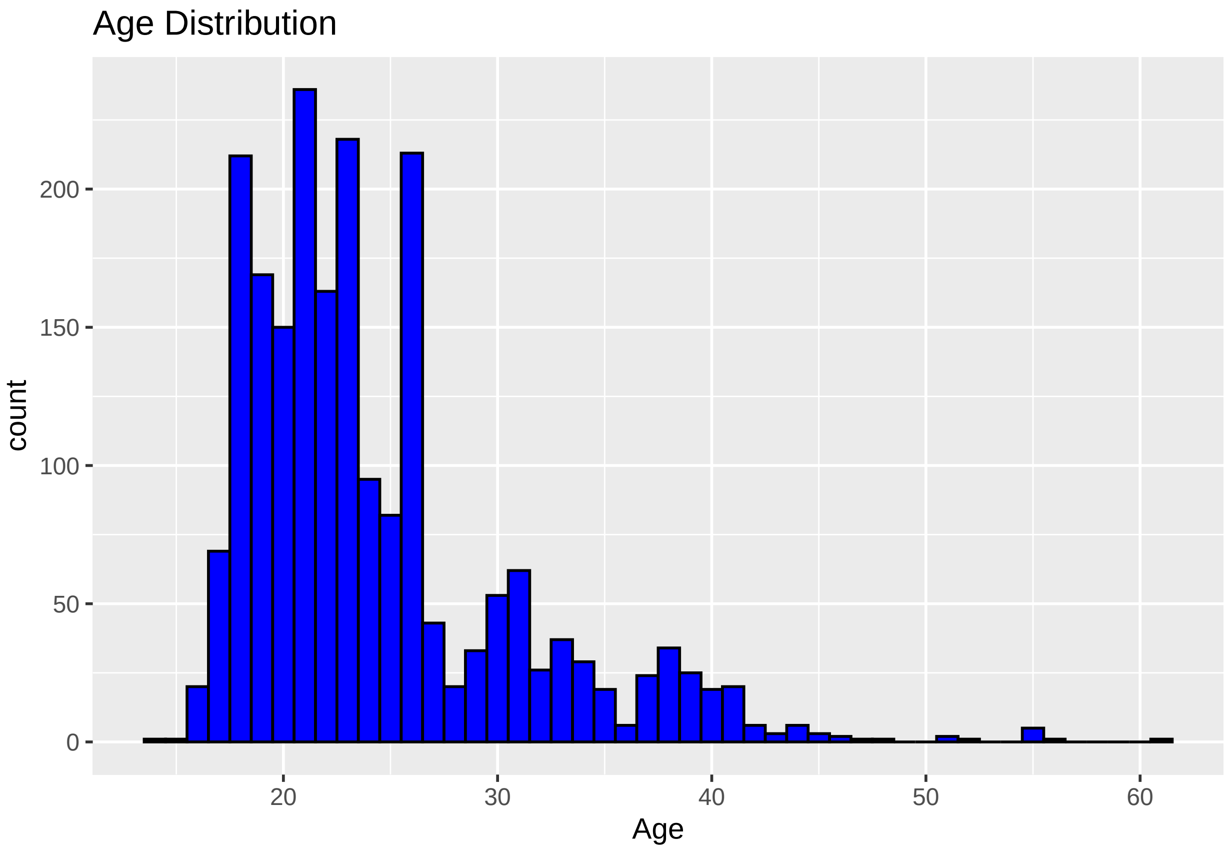
Plotting a histogram for a numerical column

Load the necessary library

```r
library(ggplot2)
```

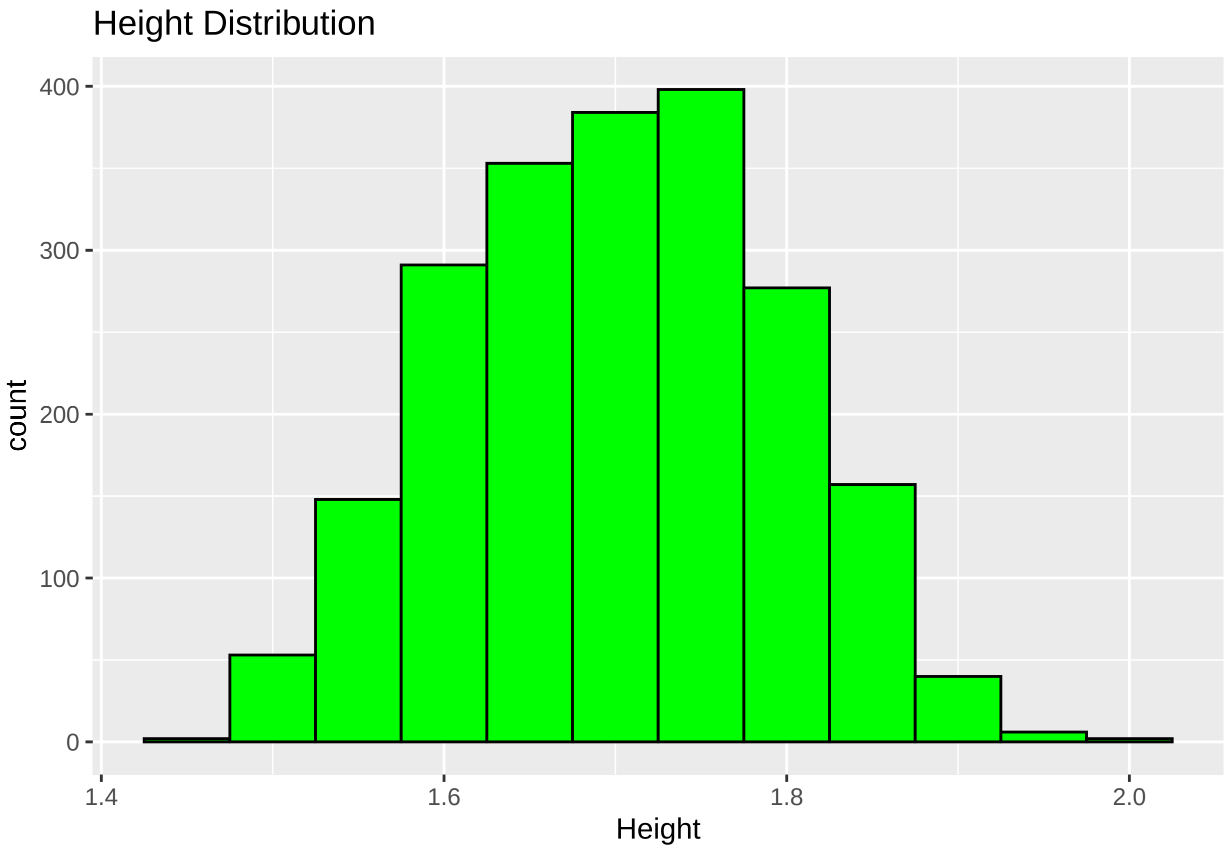
Histogram for Age variable

```r
ggplot(data, aes(x = Age)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  ggtitle("Age Distribution")
```



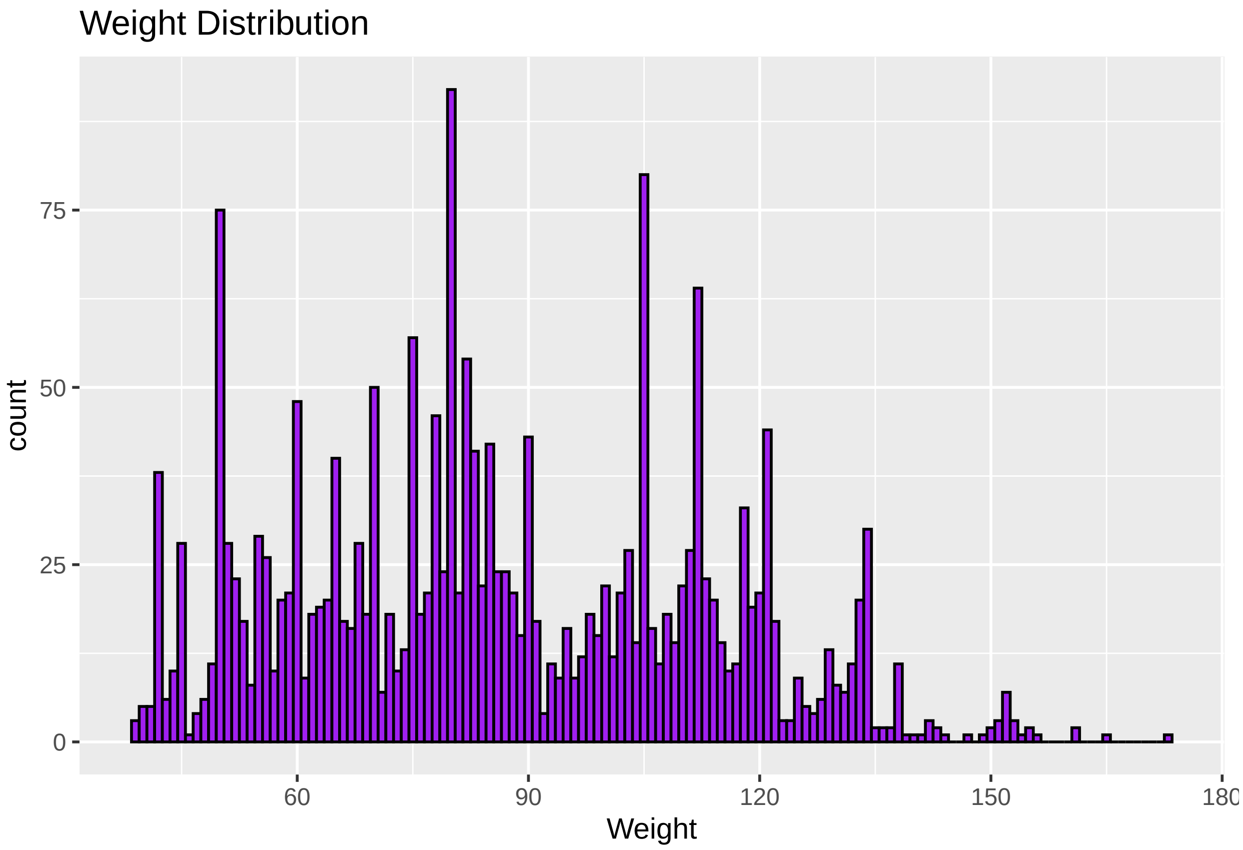
Histogram for Height variable

```
ggplot(data, aes(x = Height)) +
  geom_histogram(binwidth = 0.05, fill = "green", color = "black") +
  ggtitle("Height Distribution")
```



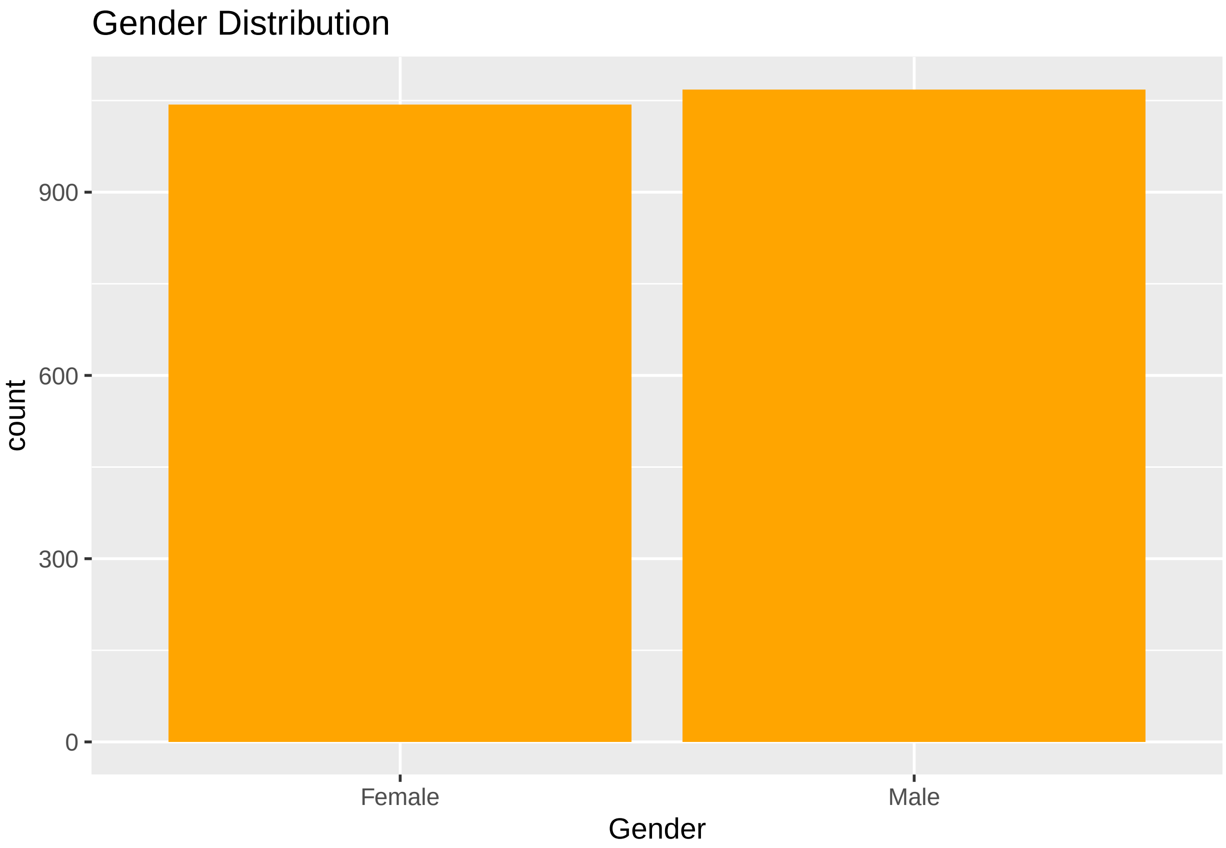
Histogram for Weight variable

```r
ggplot(data, aes(x = Weight)) +
  geom_histogram(binwidth = 1, fill = "purple", color = "black") +
  ggtitle("Weight Distribution")
```



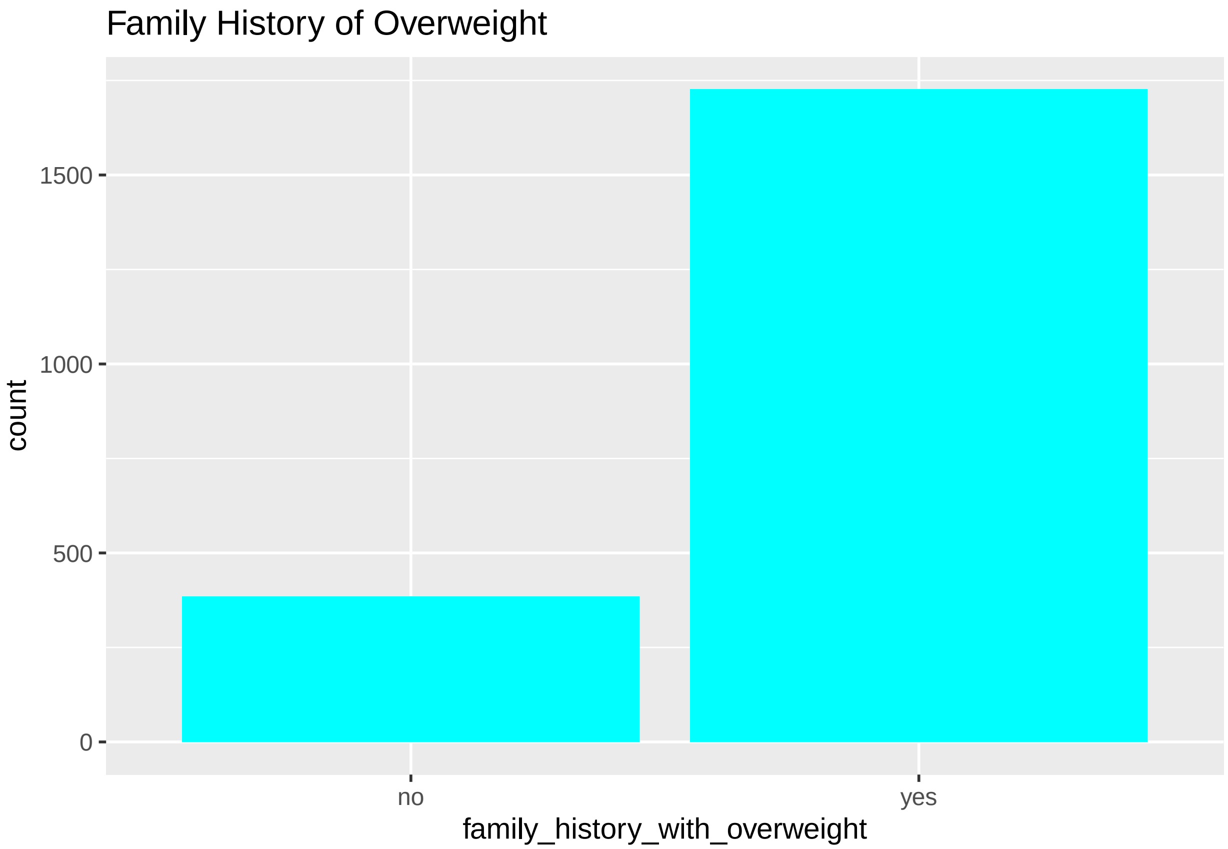
Bar chart for Gender variable

```
ggplot(data, aes(x = Gender)) +
  geom_bar(fill = "orange") +
  ggtitle("Gender Distribution")
```



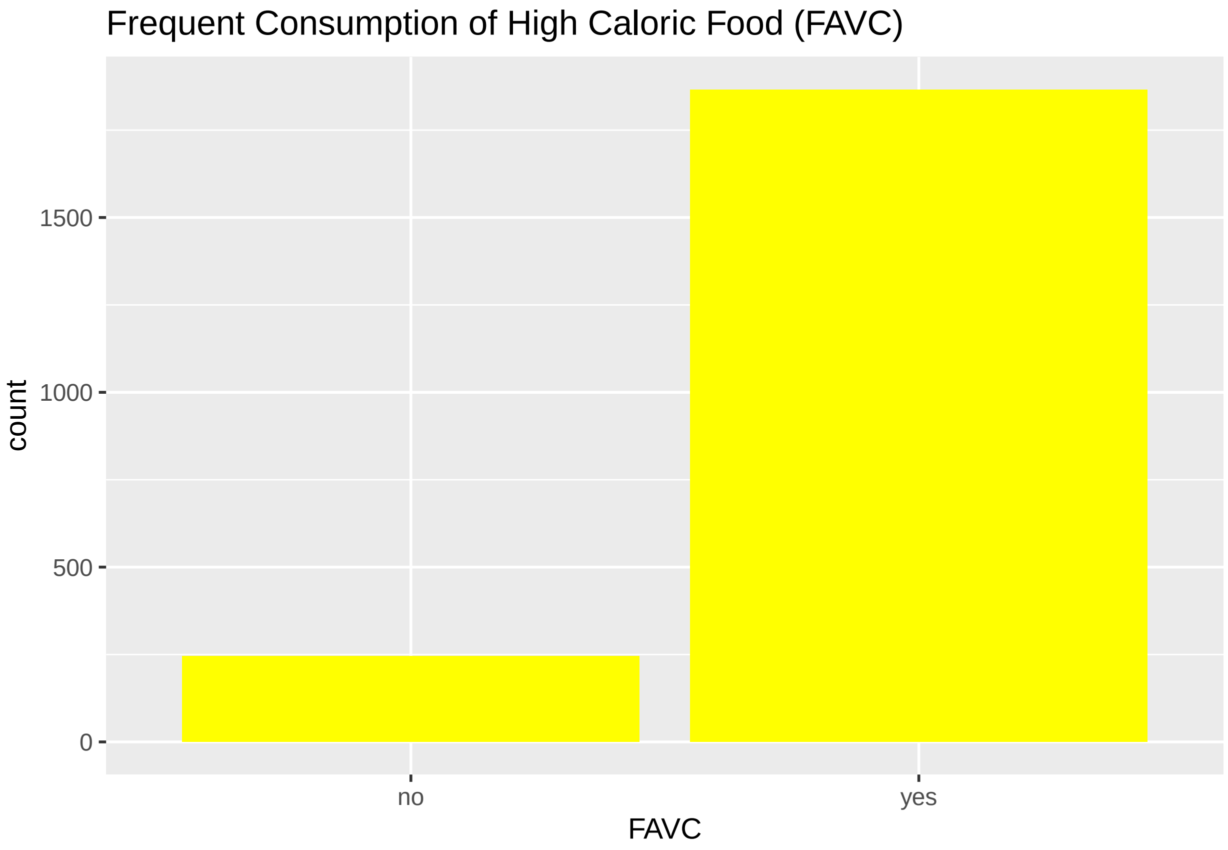
Bar chart for family__history__with__overweight variable

```
ggplot(data, aes(x = family_history_with_overweight)) +
  geom_bar(fill = "cyan") +
  ggtitle("Family History of Overweight")
```



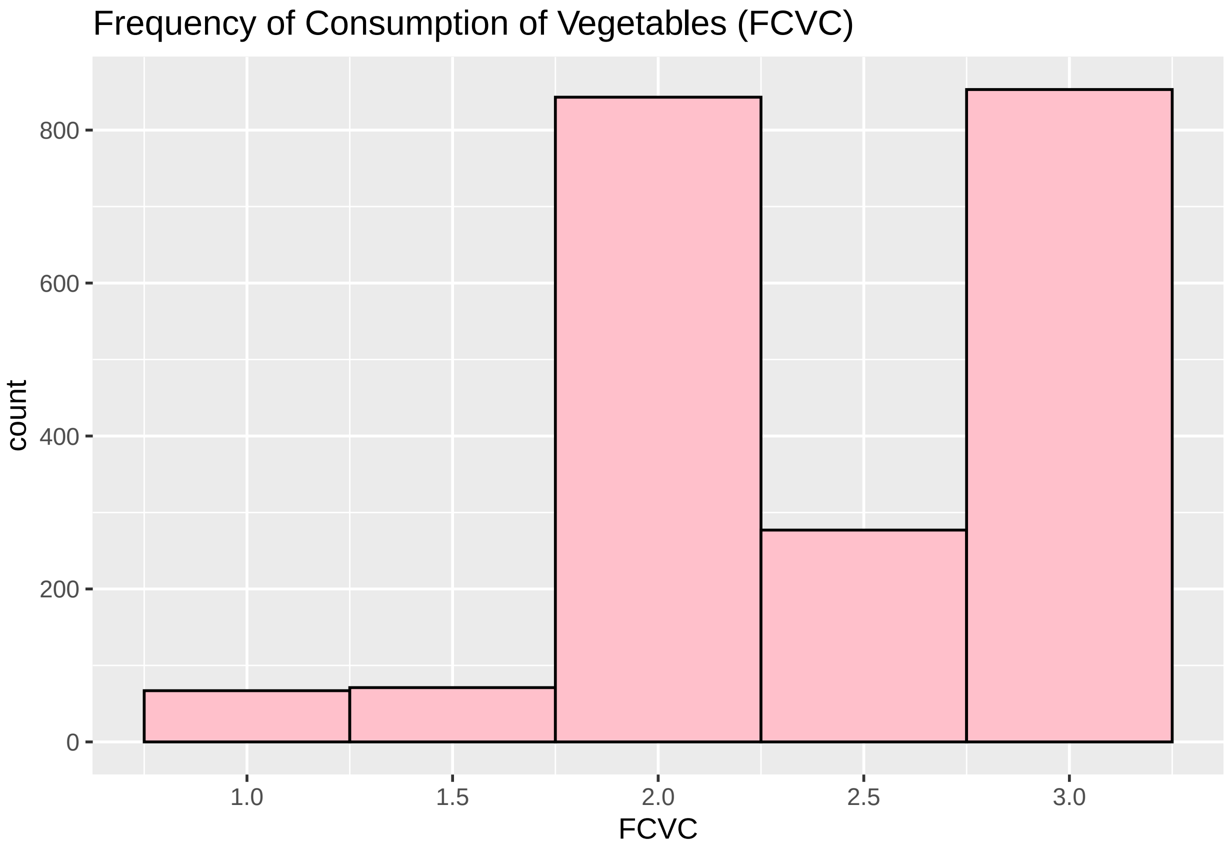
Bar chart for FAVC (Frequent Consumption of High Caloric Food)

```r
ggplot(data, aes(x = FAVC)) +
  geom_bar(fill = "yellow") +
  ggtitle("Frequent Consumption of High Caloric Food (FAVC)")
```



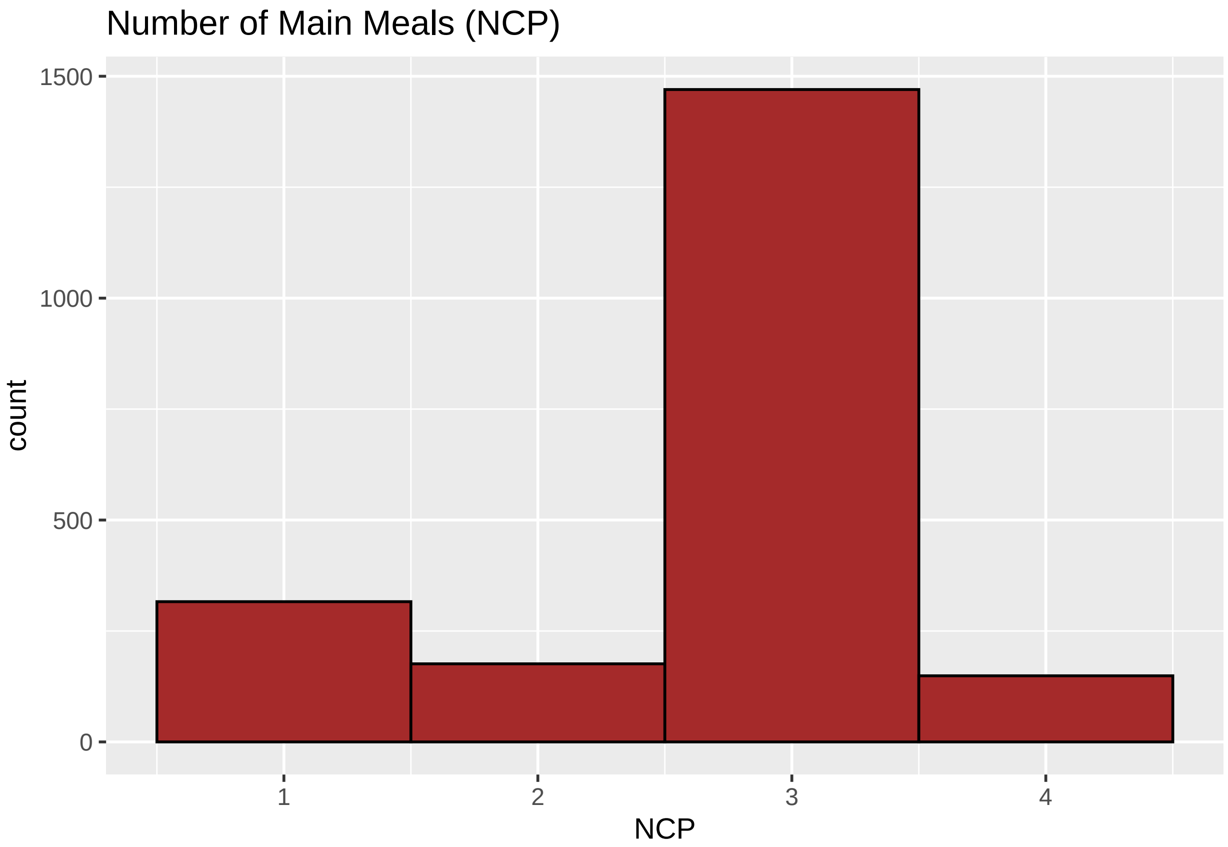
Histogram for FCVC (Frequency of Consumption of Vegetables)

```
ggplot(data, aes(x = FCVC)) +
  geom_histogram(binwidth = 0.5, fill = "pink", color = "black") +
  ggtitle("Frequency of Consumption of Vegetables (FCVC)")
```



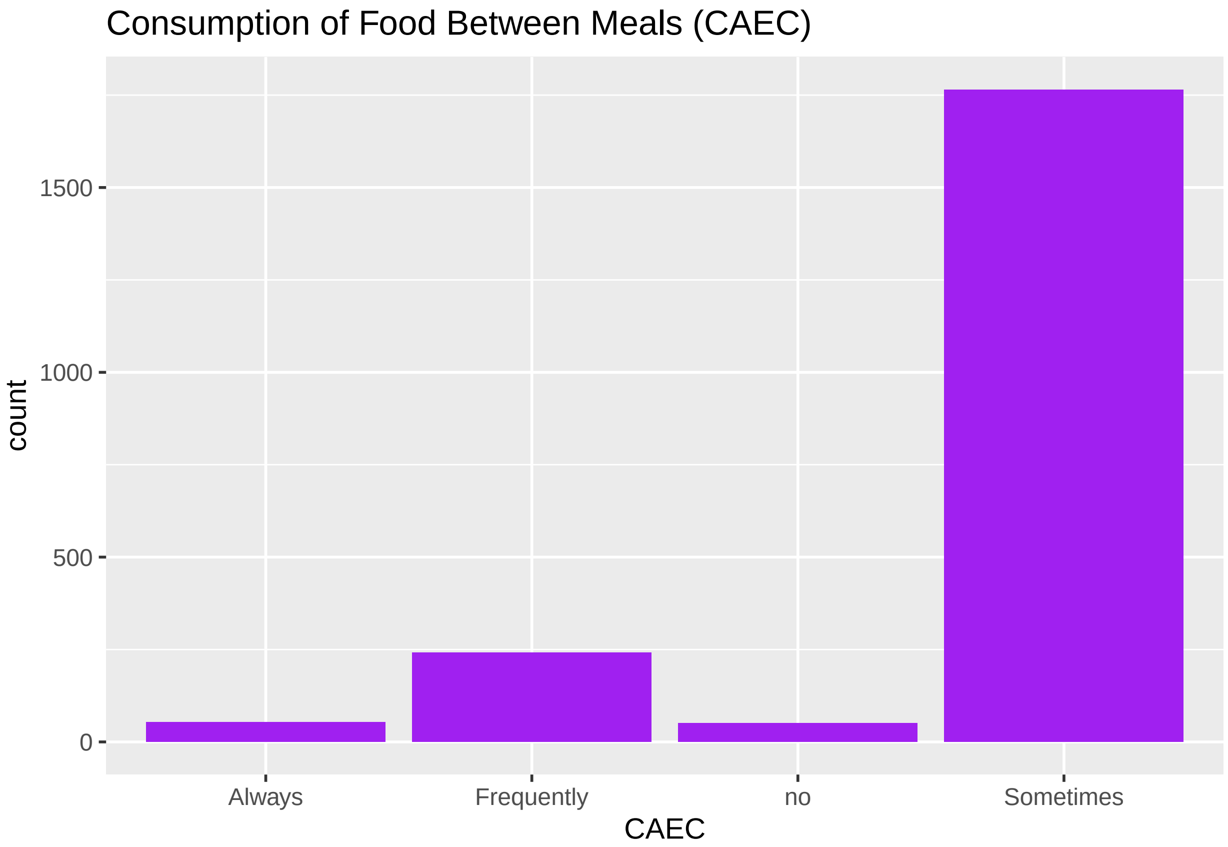
Histogram for NCP (Number of Main Meals)

```
ggplot(data, aes(x = NCP)) +
  geom_histogram(binwidth = 1, fill = "brown", color = "black") +
  ggtitle("Number of Main Meals (NCP)")
```



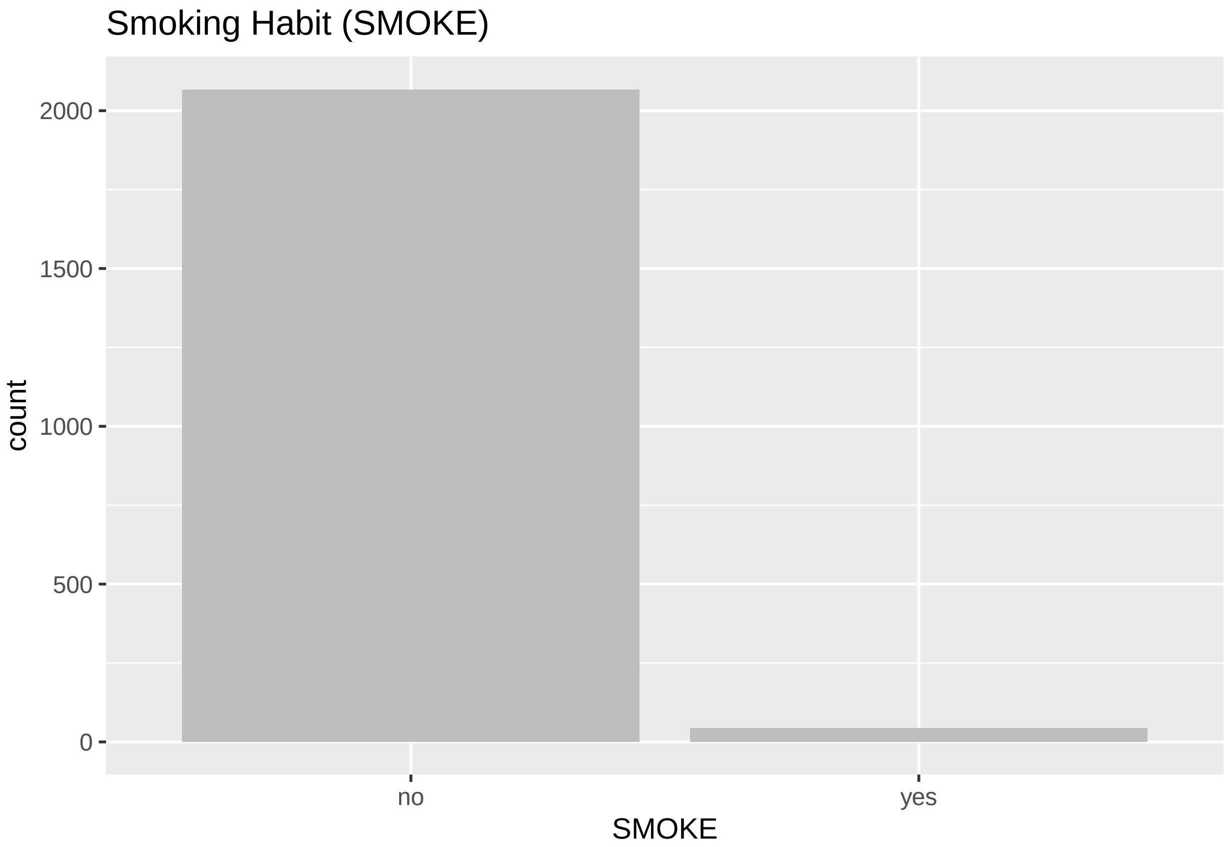
Bar chart for CAEC (Consumption of Food Between Meals)

```
ggplot(data, aes(x = CAEC)) +
  geom_bar(fill = "purple") +
  ggtitle("Consumption of Food Between Meals (CAEC)")
```



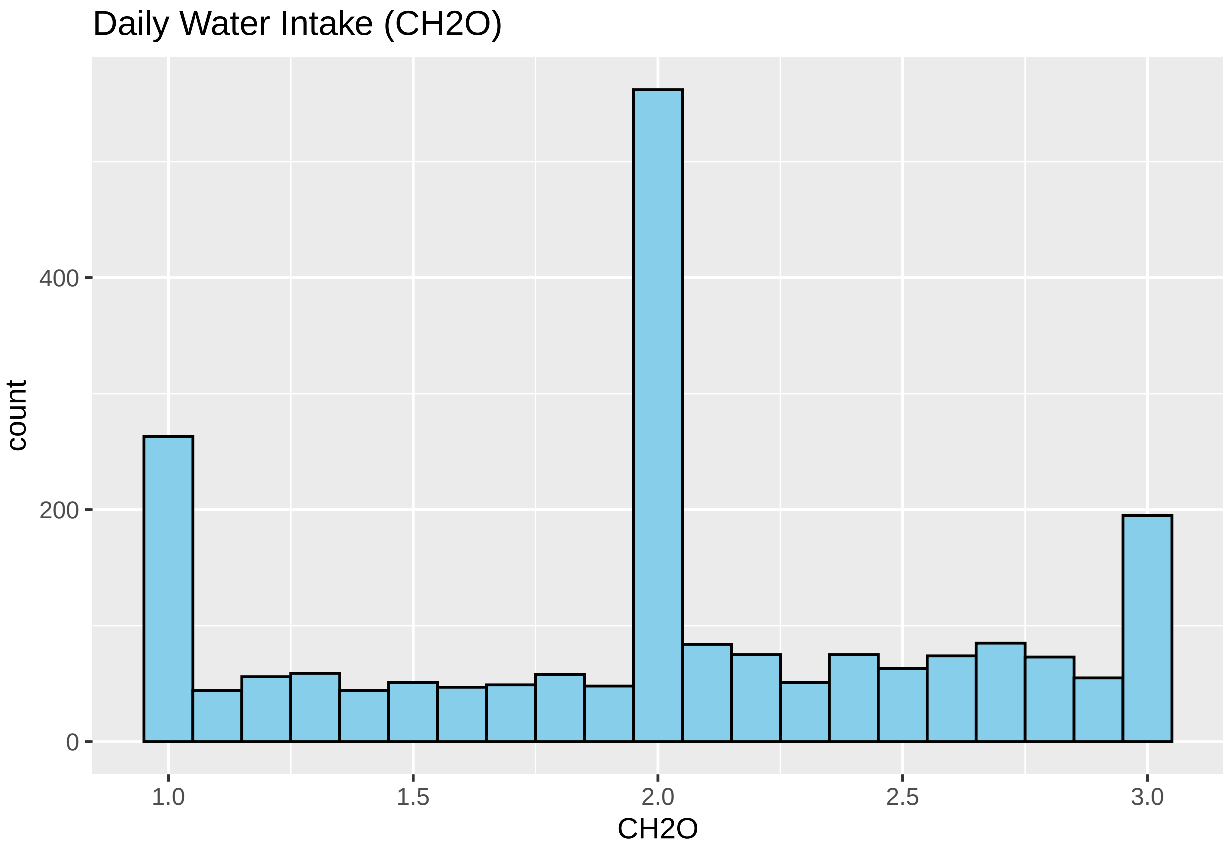
Bar chart for SMOKE variable

```
ggplot(data, aes(x = SMOKE)) +
  geom_bar(fill = "grey") +
  ggtitle("Smoking Habit (SMOKE)")
```



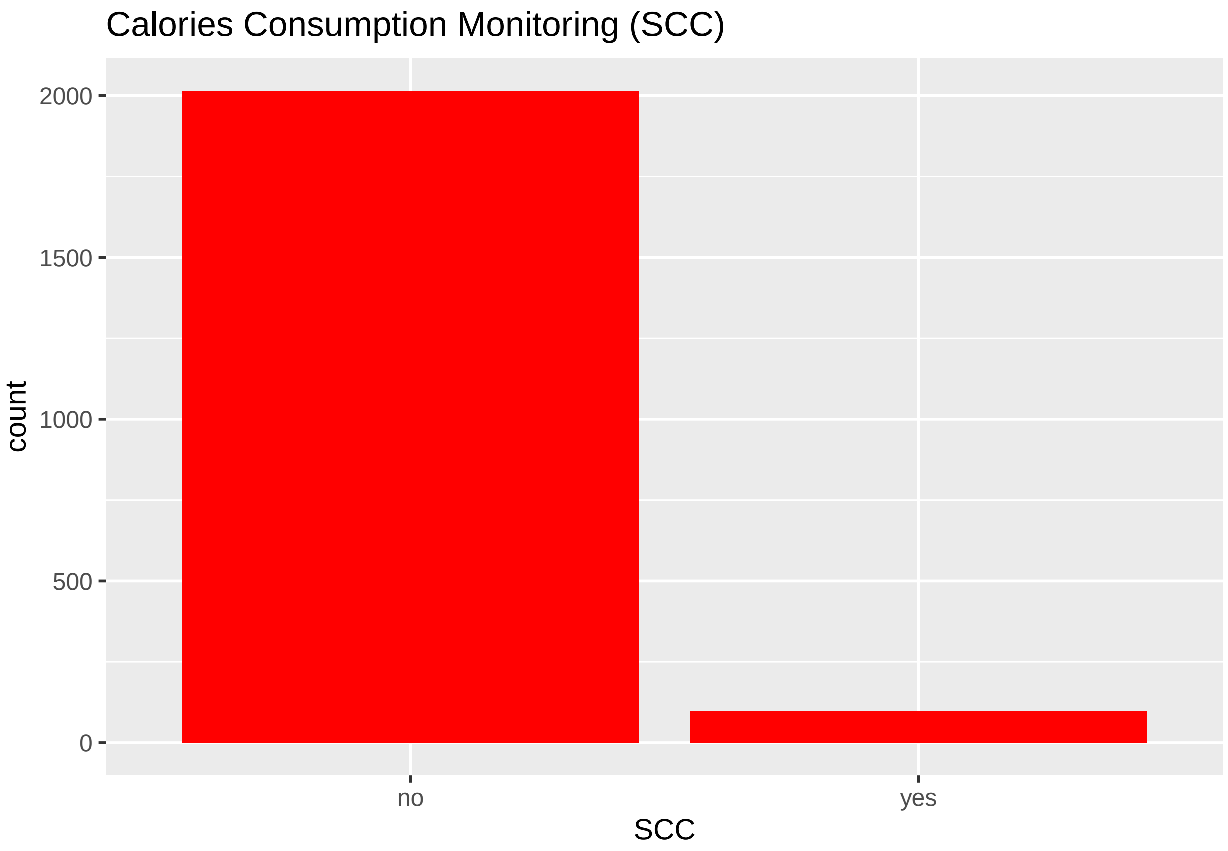
Histogram for CH2O (Daily Water Intake)

```
ggplot(data, aes(x = CH2O)) +
  geom_histogram(binwidth = 0.1, fill = "skyblue", color = "black") +
  ggtitle("Daily Water Intake (CH2O)")
```



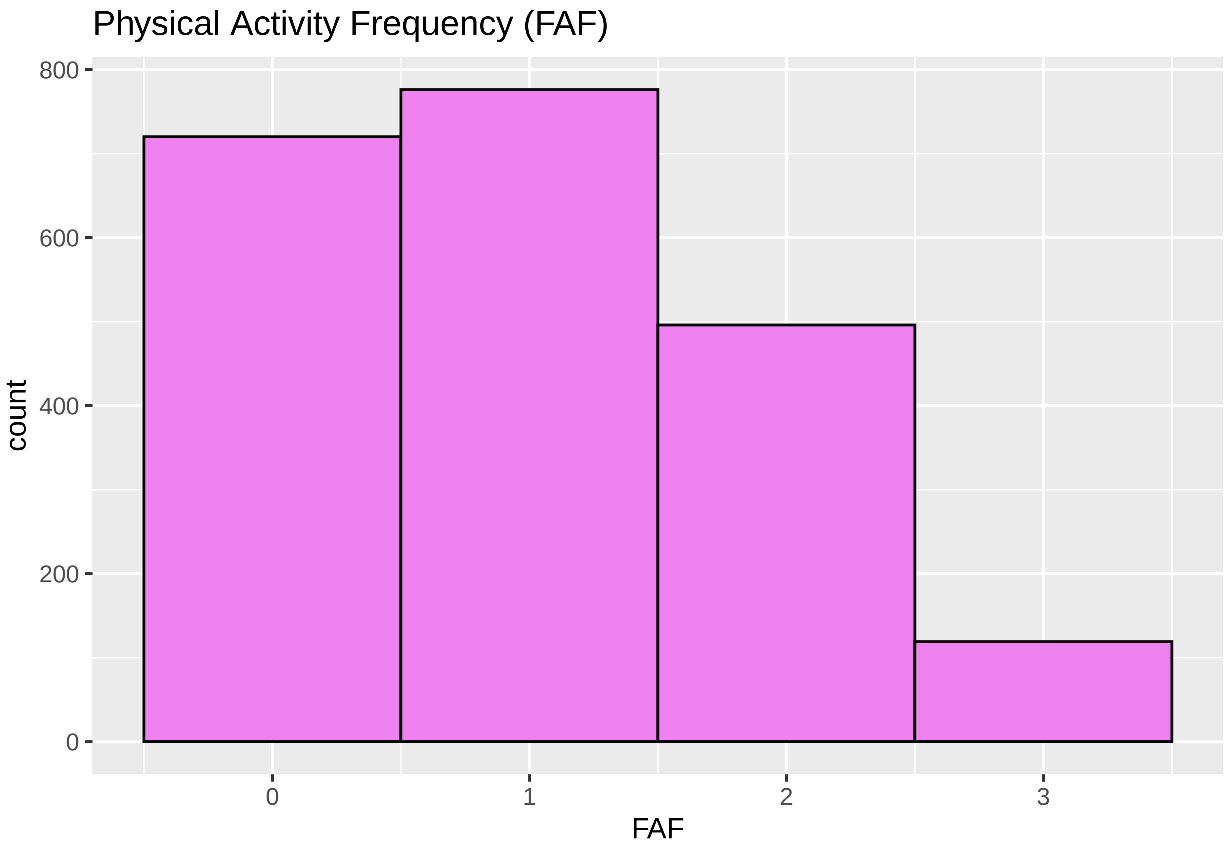
Bar chart for SCC (Calories Consumption Monitoring)

```r
ggplot(data, aes(x = SCC)) +
  geom_bar(fill = "red") +
  ggtitle("Calories Consumption Monitoring (SCC)")
```



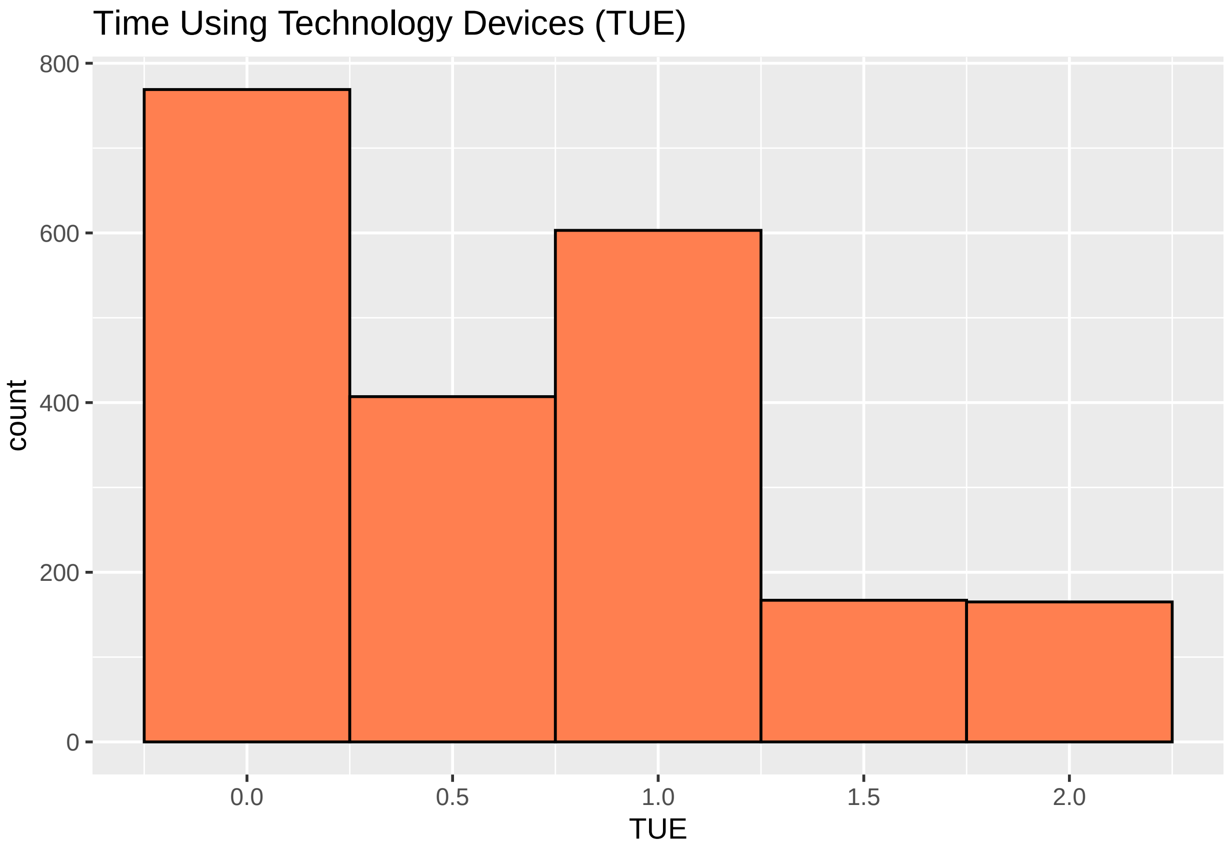
Histogram for FAF (Physical Activity Frequency)

```
ggplot(data, aes(x = FAF)) +
  geom_histogram(binwidth = 1, fill = "violet", color = "black") +
  ggtitle("Physical Activity Frequency (FAF)")
```



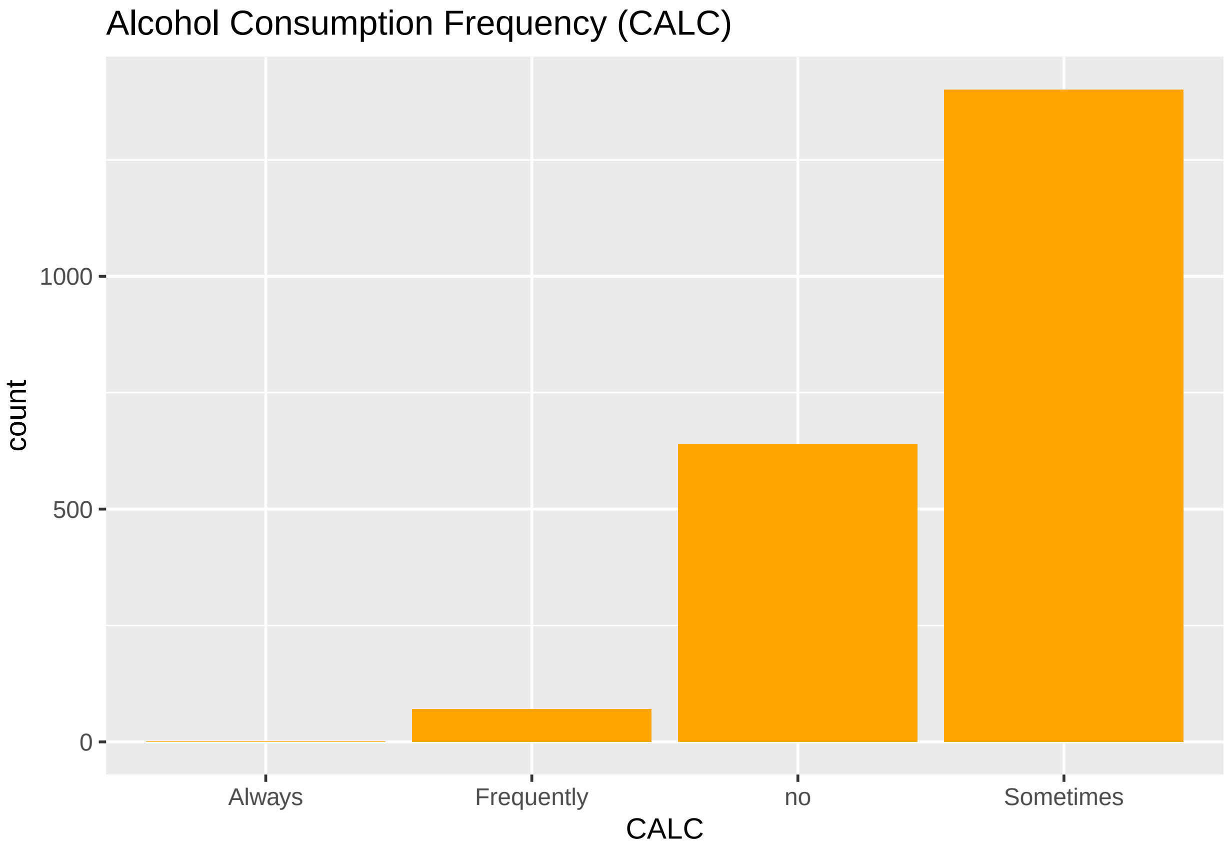
Histogram for TUE (Time Using Technology Devices)

```
ggplot(data, aes(x = TUE)) +
  geom_histogram(binwidth = 0.5, fill = "coral", color = "black") +
  ggtitle("Time Using Technology Devices (TUE)")
```



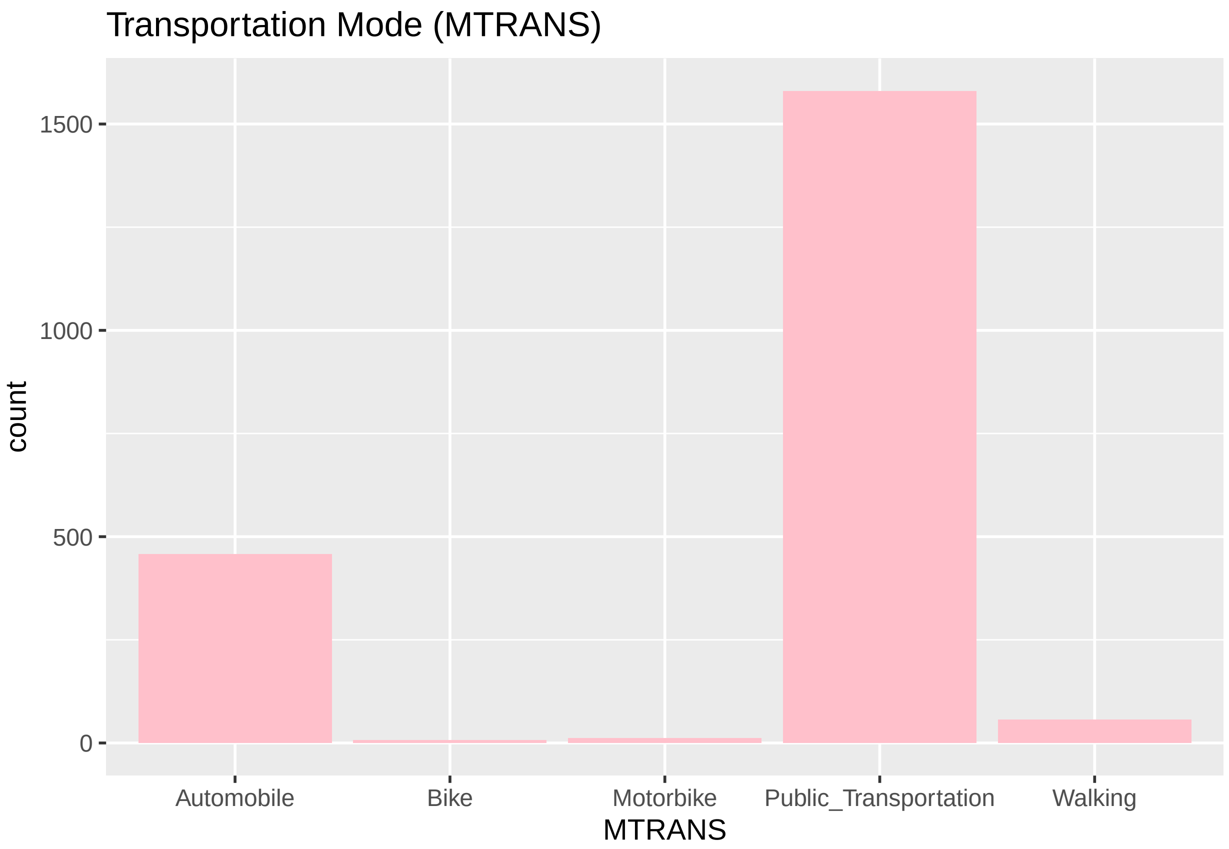
Bar chart for CALC (Alcohol Consumption Frequency)

```
ggplot(data, aes(x = CALC)) +
  geom_bar(fill = "orange") +
  ggtitle("Alcohol Consumption Frequency (CALC)")
```



Bar chart for MTRANS (Transportation Used)

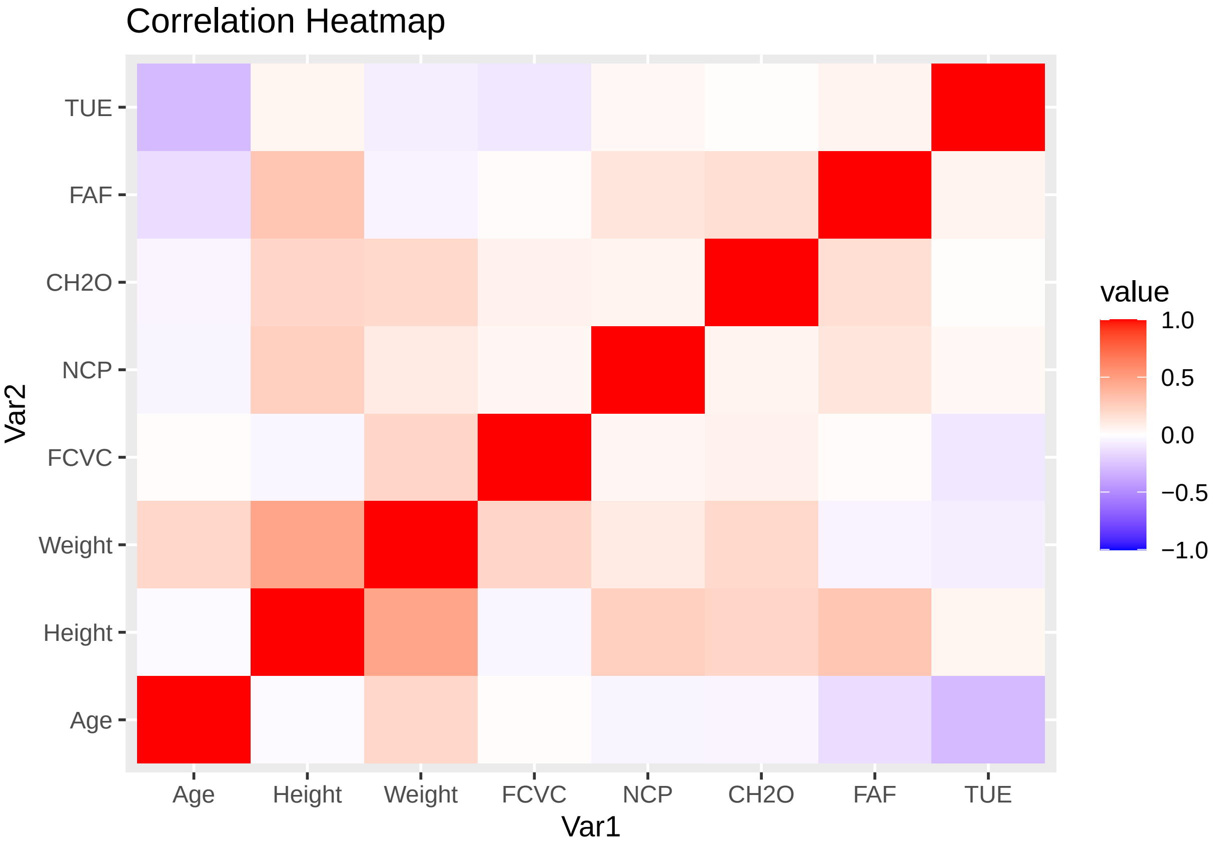
```
ggplot(data, aes(x = MTRANS)) +
  geom_bar(fill = "pink") +
  ggtitle("Transportation Mode (MTRANS)")
```



```r
# Correlation matrix (Korelasyon matrisi)
cor_matrix <- cor(data[, sapply(data, is.numeric)], use = "complete.obs")
print(cor_matrix)
```

```
##                Age      Height      Weight        FCVC         NCP        CH2O
## Age     1.00000000 -0.02595813  0.20256010  0.01629089 -0.04394373 -0.04530386
## Height -0.02595813  1.00000000  0.46313612 -0.03812106  0.24367173  0.21337592
## Weight  0.20256010  0.46313612  1.00000000  0.21612471  0.10746899  0.20057539
## FCVC    0.01629089 -0.03812106  0.21612471  1.00000000  0.04221630  0.06846147
## NCP    -0.04394373  0.24367173  0.10746899  0.04221630  1.00000000  0.05708800
## CH2O   -0.04530386  0.21337592  0.20057539  0.06846147  0.05708800  1.00000000
## FAF    -0.14493833  0.29470900 -0.05143627  0.01993940  0.12950431  0.16723649
## TUE    -0.29693059  0.05191167 -0.07156136 -0.10113485  0.03632557  0.01196534
##                FAF         TUE
## Age    -0.14493833 -0.29693059
## Height  0.29470900  0.05191167
## Weight -0.05143627 -0.07156136
## FCVC    0.01993940 -0.10113485
## NCP     0.12950431  0.03632557
## CH2O    0.16723649  0.01196534
## FAF     1.00000000  0.05856207
## TUE     0.05856207  1.00000000
```

```r
# Heatmap (simple heatmap using ggplot)
library(reshape2)
cor_melted <- melt(cor_matrix)
ggplot(data = cor_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1)) +
  ggtitle("Correlation Heatmap")
```



As a result of examining and visualizing the data, the variables that are considered to directly affect obesity and are relatively easier to address with solutions have been identified as follows;

- FCVC (Frequency of Vegetable Consumption): Increase vegetable consumption.
- FAVC (Consumption of High-Calorie Food): Reduce the consumption of high-calorie foods.
- FAF (Physical Activity Time): Increase physical activity.
- CH2O (Water Consumption): Increase daily water consumption.
- TUE (Technology Usage Time): Reduce technology usage time to create more time for physical activity.
- In this context, we can begin exploring the obesity situation by focusing on the positive changes in these variables.

4- Model

Obesity status (NObeyesdad) is typically determined using a metric like Body Mass Index (BMI). BMI is a parameter that helps estimate a person's body fat percentage based on their height and weight. The following steps demonstrate how to calculate BMI and use this metric for determining obesity status:

BMI Calculation; BMI is calculated using the following formula:

BMI=Weight (kg)/Height (m)2

Where;

Weight is in kilograms (kg), Height is in meters (m). 2. BMI Categories: Based on the BMI result, an individual's obesity status is typically classified into the following categories:

Normal weight: 18.5 - 24.9 Overweight: 25 - 29.9 Obese: 30 and above Underweight: Below 18.5

```r
category_map <- list()

# Convert categorical variables to numeric and store the mappings
data[] <- lapply(data, function(x) {
  if (is.character(x)) {
    # Convert to factor and store mappings
    factor_x <- factor(x)

    # Save the mappings (column name and numeric value for each level)
    category_map[[deparse(substitute(x))]] <- setNames(seq_along
                                                       (levels(factor_x)),
                                                       levels(factor_x))

    # Convert to numeric
    return(as.numeric(factor_x))
  } else {
    return(x)  # No change for numeric data
  }
})
```

```r
data$BMI <- data$Weight / (data$Height^2)  # BMI hesaplama

# Select overweight and obese individuals (BMI >= 25)
overweight_obese_data <- data[data$BMI >= 27.45, ]

# Display the number of overweight or obese individuals
cat("Number of overweight or obese individuals: ",
    nrow(overweight_obese_data), "\n")
```

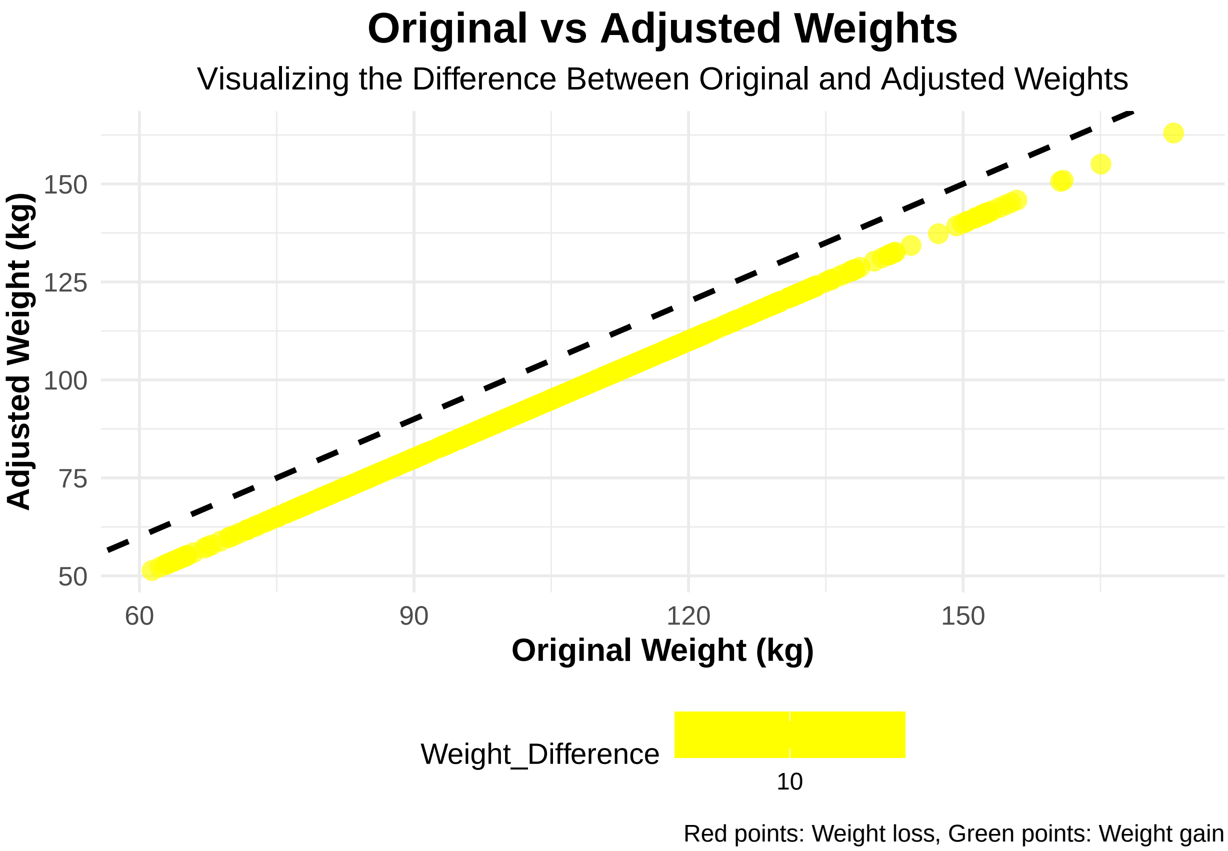
```
## Number of overweight or obese individuals:  1210
```

```r
# Standardization
overweight_obese_data_scaled <- overweight_obese_data
overweight_obese_data_scaled[, c("FCVC", "FAVC", "FAF", "CH2O", "TUE")] <-
  scale(overweight_obese_data[, c("FCVC", "FAVC", "FAF", "CH2O", "TUE")])

# Recreate model using the standardized data
model_scaled <- lm(Weight ~ FCVC + FAVC + FAF + CH2O + TUE,
                   data = overweight_obese_data_scaled)

# Predict weight changes for each observation in the dataset
predicted_weight_changes_scaled <- predict(model_scaled,
                                           newdata = overweight_obese_data_scaled)

# Handle NA values if present
predicted_weight_changes_scaled[is.na(predicted_weight_changes_scaled)] <- 0

# Limit the weight change predictions between -10 and 10
predicted_weight_changes_scaled <-
  pmin(pmax(predicted_weight_changes_scaled, -10), 10)
```

```
# Calculate the adjusted weights by applying the predicted weight changes
overweight_obese_data$Adjusted_Weight <-
  overweight_obese_data$Weight - predicted_weight_changes_scaled

# New data for prediction based on mean values with specific adjustments
new_data <- data.frame(
  FCVC = mean(overweight_obese_data$FCVC, na.rm = TRUE) * 1.02,
  FAVC = mean(overweight_obese_data$FAVC, na.rm = TRUE) * 0.98,
  FAF = mean(overweight_obese_data$FAF, na.rm = TRUE) * 1.05,
  CH2O = mean(overweight_obese_data$CH2O, na.rm = TRUE) * 1.02,
  TUE = mean(overweight_obese_data$TUE, na.rm = TRUE) * 0.95
)

# Predict the weight changes using the new data
predicted_new_weight_change <- predict(model_scaled, newdata = new_data)

# Calculate the new predicted weight by applying the weight change
new_predicted_weight <-
  mean(overweight_obese_data$Weight, na.rm = TRUE) - predicted_new_weight_change

# Display the first few rows of the original and adjusted weight
head(overweight_obese_data[, c("Weight", "Adjusted_Weight")])
```

```
## # A tibble: 6 x 2
##   Weight Adjusted_Weight
##    <dbl>           <dbl>
## 1   89.8            79.8
## 2  105              95
## 3   99              89
## 4   78              68
## 5   82              72
## 6   80              70
```

```
# Add the Weight_Difference column
overweight_obese_data$Weight_Difference <-
  overweight_obese_data$Weight - overweight_obese_data$Adjusted_Weight

# Create the plot
ggplot(overweight_obese_data, aes(x = Weight, y = Adjusted_Weight)) +
  geom_point(aes(color = Weight_Difference), size = 3, alpha = 0.7) +
  # Color the points based on the difference
  scale_color_gradient2(low = "red", mid = "yellow",
                      high = "green", midpoint = 0) +
  # Show the difference using color
  geom_abline(slope = 1, intercept = 0, color = "black",
            linetype = "dashed", linewidth = 1) +  # Add the Y = X line
  labs(
    title = "Original vs Adjusted Weights",
    subtitle = "Visualizing the Difference Between Original and Adjusted Weights",
    x = "Original Weight (kg)",
    y = "Adjusted Weight (kg)",
    caption = "Red points: Weight loss, Green points: Weight gain"
  ) +
  theme_minimal() +  # Use minimal theme
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    # Adjust the title style
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    # Adjust the subtitle style
    axis.title = element_text(size = 12, face = "bold"),
    # Make axis titles bold
    axis.text = element_text(size = 10),
    # Adjust the axis text size
    plot.caption = element_text(size = 9, hjust = 1)
    # Adjust the caption text size
  ) +
  theme(legend.position = "bottom")  # Place the legend at the bottom
```



Original vs Adjusted Weights

Visualizing the Difference Between Original and Adjusted Weights

Red points: Weight loss, Green points: Weight gain

5-Model Testing

```
# Splitting the data into Training and Testing Sets
set.seed(123)  # For reproducibility of results
library(caret)
```

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

```
train_index <- createDataPartition(overweight_obese_data$Weight,
                                   p = 0.8, list = FALSE)
train_data <- overweight_obese_data[train_index, ]
test_data <- overweight_obese_data[-train_index, ]
```

```r
# Creating model using the training data
model_train <- lm(Weight ~ FCVC + FAVC + FAF + CH2O + TUE, data = train_data)

# Summary of the model trained on the training data
summary(model_train)
```

```
##
## Call:
## lm(formula = Weight ~ FCVC + FAVC + FAF + CH2O + TUE, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.990 -11.569  -1.485  13.150  53.117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.7254     5.0468   1.333    0.183
## FCVC         14.0693     1.0108  13.919  < 2e-16 ***
## FAVC         25.6381     2.0906  12.263  < 2e-16 ***
## FAF           3.4866     0.6673   5.225 2.14e-07 ***
## CH2O          4.7735     0.8750   5.455 6.22e-08 ***
## TUE          -0.2230     0.9223  -0.242    0.809
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.09 on 964 degrees of freedom
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3013
## F-statistic: 84.55 on 5 and 964 DF,  p-value: < 2.2e-16
```

```r
# Making predictions using the test data
predicted_weights <- predict(model_train, newdata = test_data)

# Calculating the difference between actual and predicted weights
comparison <- data.frame(
  Actual = test_data$Weight,
  Predicted = predicted_weights,
  Difference = test_data$Weight - predicted_weights
)

# Displaying the first few rows
head(comparison)
```

```
##   Actual Predicted   Difference
## 1   89.8  70.04910  19.75090261
## 2  105.0 121.05707 -16.05706908
## 3   82.0 104.98310 -22.98310477
## 4   80.0  80.06269  -0.06268947
## 5   91.0  95.68722  -4.68722399
## 6   80.0  95.68722 -15.68722399
```

```r
# Calculating performance metrics
RMSE_value <- sqrt(mean((predicted_weights - test_data$Weight)^2))
# Root Mean Squared Error (RMSE)
MAE_value <- mean(abs(predicted_weights - test_data$Weight))
# Mean Absolute Error (MAE)
```

```r
R_squared_value <- 1 - sum((test_data$Weight - predicted_weights)^2) /
  sum((test_data$Weight - mean(test_data$Weight))^2)  # R-squared

# Printing the results
cat("RMSE:", RMSE_value, "\n")
```

```
## RMSE: 16.36437
```

```r
cat("MAE:", MAE_value, "\n")
```

```
## MAE: 13.33857
```

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

```
## R-squared: 0.252711
```

```r
# Correlation Analysis (between actual and predicted values)
correlation <- cor(test_data$Weight, predicted_weights)
cat("Correlation between actual and predicted weights:", correlation, "\n")
```

```
## Correlation between actual and predicted weights: 0.5053967
```

The R^2 and coefficient values of the regression model developed as a result of the study were examined. Upon review, it was observed that the R^2 value was low. During the model development process, various other variables were tested to build the most successful model; however, the performance of these alternative models was also found to be low. Therefore, the selected variables for the model were prioritized based on their practical adjustability in real life and their direct influence on an individual's weight. In this context, the focus of the model shifted from its ability to explain the weight variable to exploring the potential impacts of lifestyle changes on weight.

6-Results

The five variables used in the model were chosen because they are factors that can be directly influenced in daily life. It has been determined that improving these factors can lead to better living standards and healthier outcomes. These selected variables are expected to support individuals in achieving a healthy weight. However, it should not be overlooked that other variables, beyond those included in this model, also play a significant role in healthy weight management. Therefore, alternative models can be developed using different variables, and positive changes in these variables can also contribute to individuals reaching their healthy weight goals.