EXPLORATORY DATA ANALYSIS OF PUBLIC HEALTH : OBESITY & LIFESTYLE

1.Understanding the Data:

The dataset contains information related to public health, focusing on obesity and lifestyle factors. It includes various attributes for each individual, such as a unique identifier (ID), age, and gender. The Body Mass Index (BMI) is recorded to assess the body fat percentage based on height and weight. Additionally, the dataset captures lifestyle factors, including physical activity level categorised as low, moderate, or high, and the average daily calorie intake. It also includes smoking status (whether the individual is a smoker or non-smoker) and frequency of alcohol consumption (such as regularly, occasionally, or rarely). Furthermore, sleep patterns are tracked through the average number of hours slept per day. Finally, the overall health status of each individual is noted, indicating whether they are healthy, overweight, or obese. This comprehensive data can be used to analyse the relationships between lifestyle factors and obesity.

The dataset includes the following columns:
ID: A unique identifier for each individual.
Age: The age of the person.
Gender: The gender of the individual (Male/Female).
BMI: Body Mass Index, which measures body fat based on height and weight.
Physical Activity Level: The level of physical activity (e.g., Low, Moderate, High).
Daily Calorie Intake: The number of calories consumed daily.
Smoking Status: Whether the individual is a smoker or non-smoker.
Alcohol Consumption: The frequency of alcohol consumption (e.g., Regularly, Occasionally, Rarely).
Sleep Hours: The average number of hours slept per day.
Health Status: The overall health status (e.g., Healthy, Overweight, Obese)

2. Data collection:

The dataset "Public Health: Obesity and Lifestyle" represents data collected to analyse various factors influencing obesity and associated lifestyle habits. The data was gathered through multiple methods, including structured surveys and questionnaires, where participants provided information on their physical activity, smoking status, alcohol consumption, dietary habits, and sleep

patterns. Additionally, some data, such as body mass index (BMI), weight, and height, were obtained during health assessments or regular health check-ups, allowing for accurate measurement of obesity indicators.

Self-reported information was also incorporated, with participants detailing aspects like daily calorie intake and hours of sleep. While self-reported data may introduce some bias, it offers valuable insights into individual lifestyle behaviours. In some cases, electronic health records might have been used to supplement and validate the collected data, ensuring a comprehensive view of participants' health conditions.

The dataset's purpose is to facilitate understanding of how various lifestyle factors correlate with obesity, providing insights into public health trends and helping to identify key areas for lifestyle interventions. Through this data collection process, the dataset offers a foundation for analysing the relationships between lifestyle choices and health outcomes.

3.Data Cleaning: Data cleaning is a crucial step in preparing your dataset for analysis, ensuring the data is accurate, consistent, and free from errors. Here's a step-by-step guide on data cleaning in R, covering essential techniques using a dataset on obesity and lifestyle. Data cleaning is essential for ensuring reliable results in any data analysis. By handling missing values, correcting data types, removing duplicates, and addressing outliers, the dataset becomes ready for deeper analysis.

str(obesity):

```
str(obesity)
'data.frame':     90 obs. of  10 variables:
 $ ID                     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Age                    : int  25 45 30 60 35 50 28 40 55 38 ...
 $ Gender                 : chr  "Male" "Female" "Male" "Female" ...
 $ BMI                    : num  28.5 34.2 22 29 31.5 27 24.5 35 26 23.5 ...
 $ Physical_Activity_Level: chr  "Moderate" "Low" "High" "Low" ...
 $ Daily_Calorie_Intake   : int  2500 3000 1800 2600 2800 2400 2000 3200 2700
2200 ...
 $ Smoking_Status         : chr  "Non-Smoker" "Smoker" "Non-Smoker"
"Non-Smoker" ...
 $ Alcohol_Consumption    : chr  "Occasionally" "Regularly" "Rarely"
"Regularly" ...
 $ Sleep_Hours            : int  7 6 8 5 6 7 8 5 6 7 ...
```

```
 $ Health_Status          : chr  "Overweight" "Obese" "Healthy" "Overweight"
...
```

summary(obesity):

```
summary(obesity)
       ID              Age           Gender               BMI
 Min.   : 1.00   Min.   :21.00   Length:90          Min.   :21.00
 1st Qu.:23.25   1st Qu.:34.25   Class :character   1st Qu.:24.50
 Median :45.50   Median :43.50   Mode  :character   Median :27.75
 Mean   :45.50   Mean   :43.54                      Mean   :28.00
 3rd Qu.:67.75   3rd Qu.:53.00                      3rd Qu.:31.50
 Max.   :90.00   Max.   :66.00                      Max.   :36.00
 Physical_Activity_Level Daily_Calorie_Intake Smoking_Status
 Length:90               Min.   :1800         Length:90
 Class :character        1st Qu.:2200         Class :character
 Mode  :character        Median :2600         Mode  :character
                         Mean   :2593
                         3rd Qu.:3000
                         Max.   :3400
 Alcohol_Consumption  Sleep_Hours  Health_Status
 Length:90           Min.   :5.0   Length:90
 Class :character    1st Qu.:6.0   Class :character
 Mode  :character    Median :6.0   Mode  :character
                     Mean   :6.5
                     3rd Qu.:8.0
                     Max.   :8.0
```

Missing Values:

```
> any(is.na(obesity))
[1] FALSE
```

Duplicate values:

```
> sum(duplicated(obesity))
[1] 0
```

Convert categorical variables to factors:

obesity$Physical_Activity_Level <- as.factor(obesity$Physical_Activity_Level)

```
str(obesity)
'data.frame':      90 obs. of  10 variables:
 $ ID                     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Age                    : int  25 45 30 60 35 50 28 40 55 38 ...
 $ Gender                 : chr  "Male" "Female" "Male" "Female" ...
 $ BMI                    : num  28.5 34.2 22 29 31.5 27 24.5 35 26 23.5 ...
 $ Physical_Activity_Level: Factor w/ 3 levels "High","Low","Moderate": 3 2 1
2 3 2 1 2 3 1 ...
 $ Daily_Calorie_Intake   : int  2500 3000 1800 2600 2800 2400 2000 3200 2700
2200 ...
 $ Smoking_Status         : chr  "Non-Smoker" "Smoker" "Non-Smoker"
"Non-Smoker" ...
```

```
 $ Alcohol_Consumption    : chr  "Occasionally" "Regularly" "Rarely"
"Regularly" ...
 $ Sleep_Hours            : int  7 6 8 5 6 7 8 5 6 7 ...
 $ Health_Status          : chr  "Overweight" "Obese" "Healthy" "Overweight"
...
```

4. Data Transformation:

Data transformation techniques such as normalisation, binning, feature engineering, and reshaping help prepare the data for analysis by making it more structured and suitable for statistical modelling or machine learning.

Normalise the Age column: scales the data to a range of [0, 1].

```
obesity$Age_normalized <- (obesity$Age - min(obesity$Age)) /
(max(obesity$Age) - min(obesity$Age))
head(obesity)
```

```
head(obesity)
  ID Age Gender  BMI Physical_Activity_Level Daily_Calorie_Intake
Smoking_Status Alcohol_Consumption Sleep_Hours
1  1  25   Male 28.5                Moderate                 2500
Non-Smoker        Occasionally           7
2  2  45 Female 34.2                     Low                 3000
Smoker           Regularly           6
3  3  30   Male 22.0                    High                 1800
Non-Smoker              Rarely           8
4  4  60 Female 29.0                     Low                 2600
Non-Smoker           Regularly           5
5  5  35   Male 31.5                Moderate                 2800
Smoker        Occasionally           6
6  6  50 Female 27.0                     Low                 2400
Non-Smoker              Rarely           7
  Health_Status Age_normalized
1    Overweight     0.08888889
2         Obese     0.53333333
3       Healthy     0.20000000
4    Overweight     0.86666667
5         Obese     0.31111111
6    Overweight     0.64444444
```

Feature Engineering: Feature engineering helps to create new variables that may improve the model's performance.

```
obesity<- obesity%>%
  mutate(Age = cut(Age,
                   breaks = c(0, 12, 18, 60, Inf),
                   labels = c("Child", "Teenager", "Adult", "Senior"),
                   right = FALSE))
head(obesity)
```

```
ID    Age Gender  BMI Physical_Activity_Level
Daily_Calorie_Intake Smoking_Status
1  1  Adult   Male 28.5                Moderate
2500     Non-Smoker
2  2  Adult Female 34.2                     Low
3000         Smoker
3  3  Adult   Male 22.0                    High
1800     Non-Smoker
4  4 Senior Female 29.0                     Low
2600     Non-Smoker
5  5  Adult   Male 31.5                Moderate
2800         Smoker
6  6  Adult Female 27.0                     Low
2400     Non-Smoker
  Alcohol_Consumption Sleep_Hours Health_Status
Age_normalized
1        Occasionally           7    Overweight
0.08888889
2           Regularly           6         Obese
0.53333333
3              Rarely           8       Healthy
0.20000000
4           Regularly           5    Overweight
0.86666667
5        Occasionally           6         Obese
0.31111111
6              Rarely           7    Overweight
0.64444444
```

Disaggregate Data:

means breaking down a dataset into more detailed components. It involves

separating data into smaller segments based on specific categories, such as age groups, gender, or regions, rather than summarizing the entire dataset. Disaggregation helps identify patterns, trends, or disparities within subsets of the data, leading to deeper insights

```
disaggregated_data <- obesity %>%
  select(Alcohol_Consumption, BMI)
print(disaggregated_data)
```

```
Alcohol_Consumption  BMI
1         Occasionally 28.5
2            Regularly 34.2
3               Rarely 22.0
4            Regularly 29.0
5         Occasionally 31.5
6               Rarely 27.0
7               Rarely 24.5
8            Regularly 35.0
9         Occasionally 26.0
10              Rarely 23.5
11           Regularly 33.0
12              Rarely 21.0
13        Occasionally 30.0
14           Regularly 36.0
15              Rarely 24.0
16        Occasionally 28.0
17           Regularly 32.5
18              Rarely 26.0
19        Occasionally 25.5
20           Regularly 33.5
21              Rarely 23.0
22        Occasionally 29.5
23           Regularly 34.0
24              Rarely 22.5
25        Occasionally 27.0
26           Regularly 31.0
27              Rarely 28.0
28           Regularly 35.0
```

29	Occasionally	23.0
30	Regularly	32.5
31	Occasionally	26.5
32	Regularly	28.0
33	Rarely	23.5
34	Occasionally	30.5
35	Occasionally	29.0
36	Rarely	24.5
37	Regularly	34.5
38	Occasionally	25.0
39	Regularly	32.0
40	Rarely	22.0
41	Occasionally	28.5
42	Regularly	35.5
43	Rarely	23.0
44	Occasionally	30.0
45	Occasionally	26.0
46	Regularly	33.5
47	Rarely	21.5
48	Occasionally	27.0
49	Regularly	32.5
50	Rarely	22.5
51	Occasionally	29.5
52	Regularly	31.0
53	Rarely	24.0
54	Occasionally	28.5
55	Occasionally	26.0
56	Regularly	34.0
57	Rarely	21.0
58	Occasionally	27.5
59	Regularly	31.5
60	Rarely	22.5
61	Occasionally	27.0
62	Rarely	25.5
63	Occasionally	22.0
64	Regularly	33.0
65	Rarely	28.5
66	Occasionally	24.0

```
67           Regularly 30.5
68              Rarely 26.0
69           Regularly 34.0
70        Occasionally 23.5
71              Rarely 27.5
72        Occasionally 29.0
73              Rarely 22.5
74           Regularly 32.5
75        Occasionally 25.0
76           Regularly 35.0
77              Rarely 26.5
78        Occasionally 24.5
79           Regularly 30.0
80        Occasionally 27.0
81              Rarely 23.0
82           Regularly 33.5
83        Occasionally 28.0
84              Rarely 25.5
85           Regularly 32.0
86        Occasionally 23.0
87           Regularly 27.5
88        Occasionally 30.5
89              Rarely 21.5
90           Regularly 34.5
```

5. Data Integration : involves combining data from different sources or datasets to create a unified view for analysis. The goal is to merge datasets with shared information or related attributes so that the combined data can be used for deeper insights, correlations, and comprehensive analysis.

```
combined_data <- merge(obesity, lifestyle, by="Daily_Calorie_Intake")
print(head(combined_data))
Daily_Calorie_Intake ID.x   Age Gender BMI
Physical_Activity_Level Smoking_Status.x
1                 1800    3 Adult   Male  22
High       Non-Smoker
2                 1800    3 Adult   Male  22
High       Non-Smoker
```

```
3                1800    3 Adult   Male  22
High       Non-Smoker
4                1800    3 Adult   Male  22
High       Non-Smoker
5                1800    3 Adult   Male  22
High       Non-Smoker
6                1800    3 Adult   Male  22
High       Non-Smoker
  Alcohol_Consumption.x Sleep_Hours.x Health_Status
Age_normalized ID.y
1               Rarely             8       Healthy
0.2   32
2               Rarely             8       Healthy
0.2   64
3               Rarely             8       Healthy
0.2   54
4               Rarely             8       Healthy
0.2   84
5               Rarely             8       Healthy
0.2   23
6               Rarely             8       Healthy
0.2    3
  Physical_Activity Smoking_Status.y Alcohol_Consumption.y
Sleep_Hours.y
1              High       Non-smoker                  None
9
2              High       Non-smoker                  None
8
3              High       Non-smoker                  None
9
4              High       Non-smoker                  None
9
5              High       Non-smoker                  None
9
6              High       Non-smoker                  None
8
```

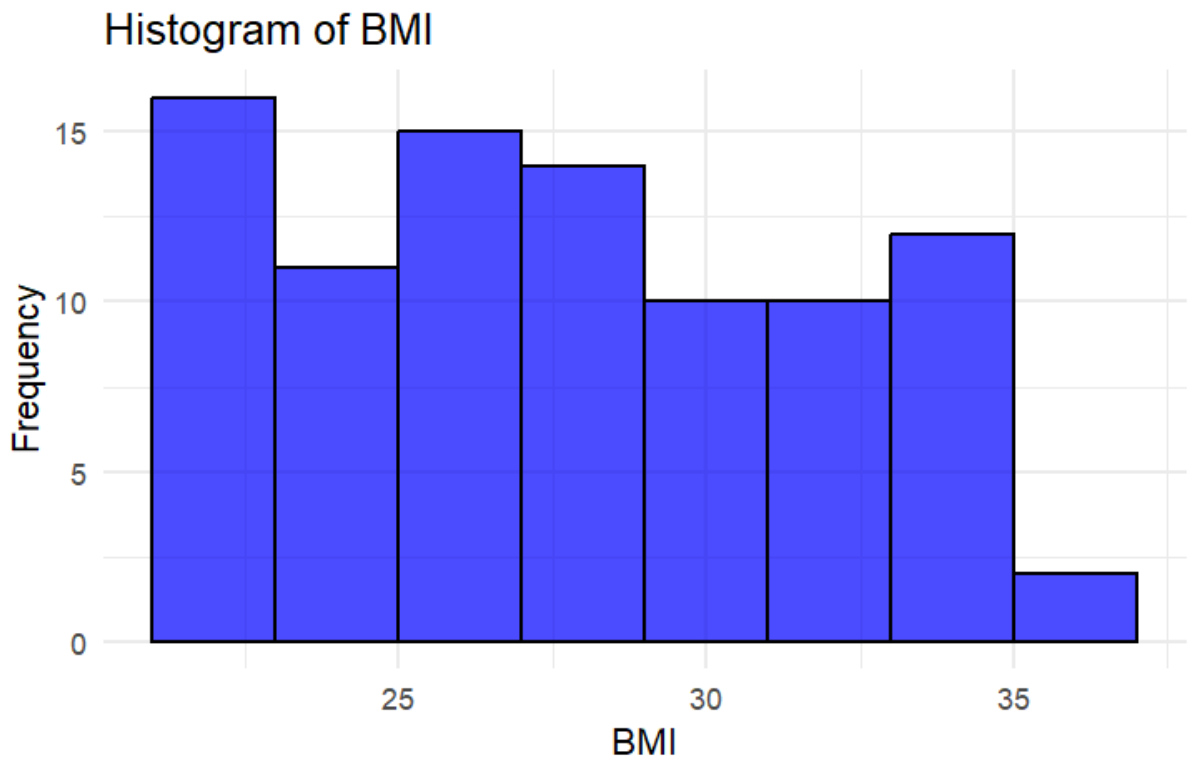
6. Data Exploration :is the process of analysing and summarising datasets to uncover insights, patterns, and relationships within the data. It involves using visualisations and statistical techniques to understand the data's underlying structure, detect anomalies, check assumptions, and establish relationships.

Univariate Analysis:

Univariate analysis focuses on understanding one variable at a time. It helps to get a basic idea of the data by looking at the central value, spread, and distribution of the variable. You can do this using summary statistics and visualisations like histograms, box plots, and density plots.

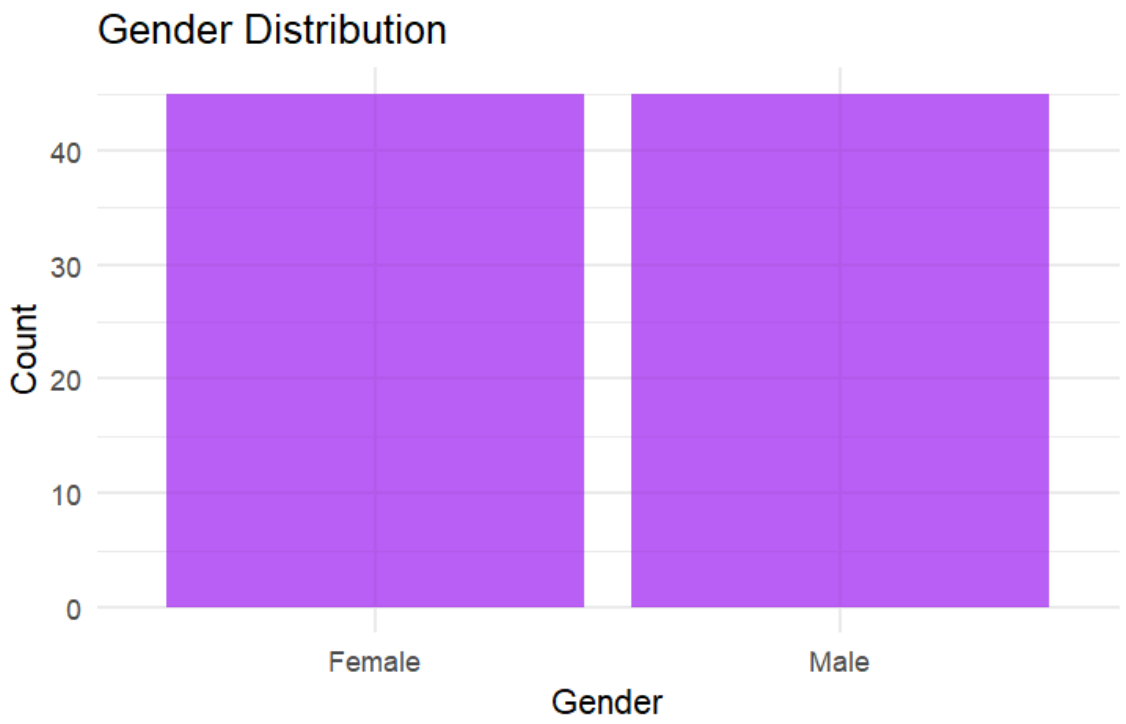
Histogram for BMI: Visualise the distribution of a continuous variable like BMI

```
ggplot(obesity, aes(x = BMI)) +
  geom_histogram(binwidth = 2, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of BMI", x = "BMI", y = "Frequency") +
  theme_minimal()
```



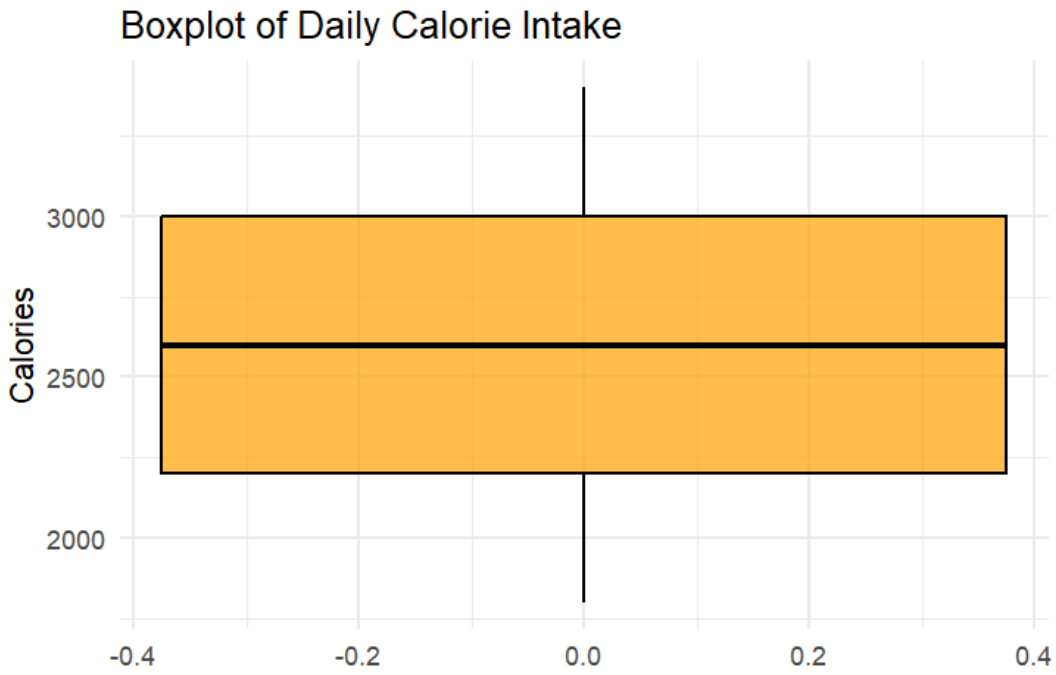
Bar Chart for Categorical Variables (Gender):Show the frequency distribution for a categorical variable like gender

```
ggplot(obesity, aes(x = Gender)) +
  geom_bar(fill = "purple", alpha = 0.7) +
  labs(title = "Gender Distribution", x = "Gender", y = "Count") +
  theme_minimal()
```



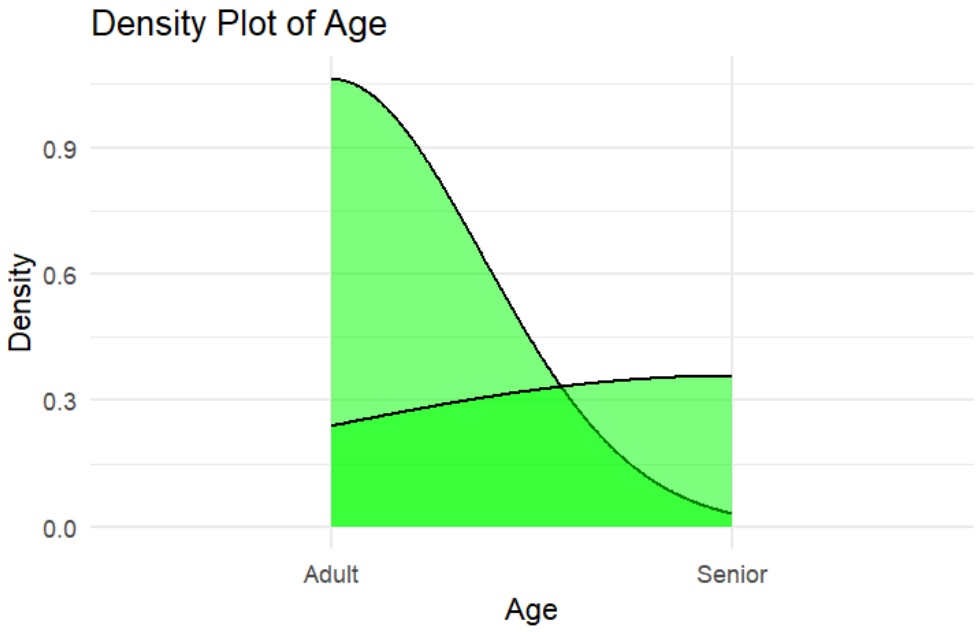
Boxplot for Daily calorie intake: Create a boxplot to visualize the distribution and identify outliers.

```
ggplot(obesity, aes(y = Daily_Calorie_Intake)) +
  geom_boxplot(fill = "orange", color = "black", alpha = 0.7) +
  labs(title = "Boxplot of Daily Calorie Intake", y = "Calories") +
  theme_minimal()
```



Density Plot for Age: To get a smooth curve representation of the data distribution:

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

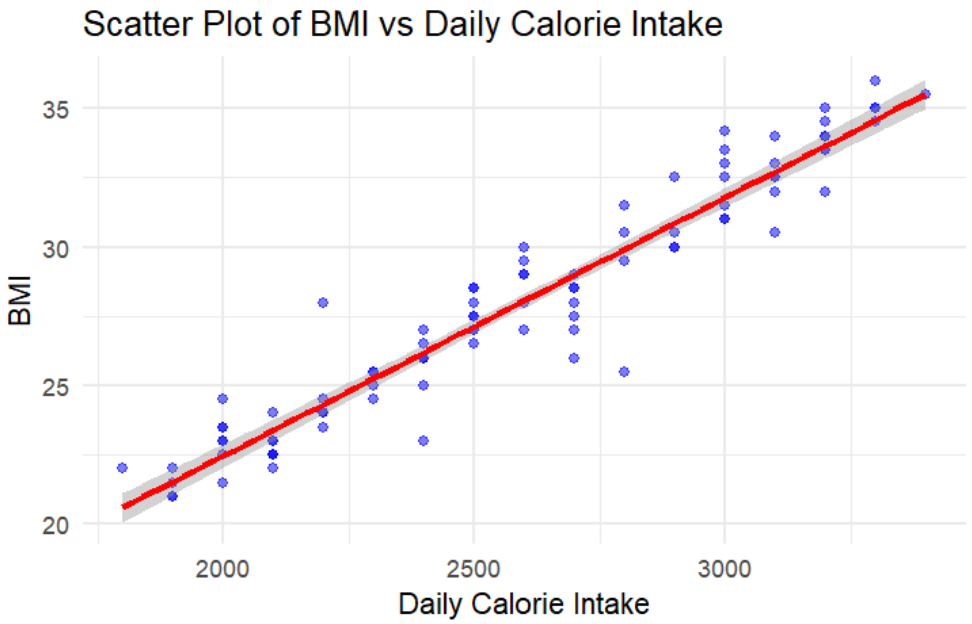


Bivariate Analysis:

Bivariate analysis examines the relationship between two variables to understand if they are correlated and how they interact. It often involves visualisations like scatter plots, box plots, or correlation coefficients.

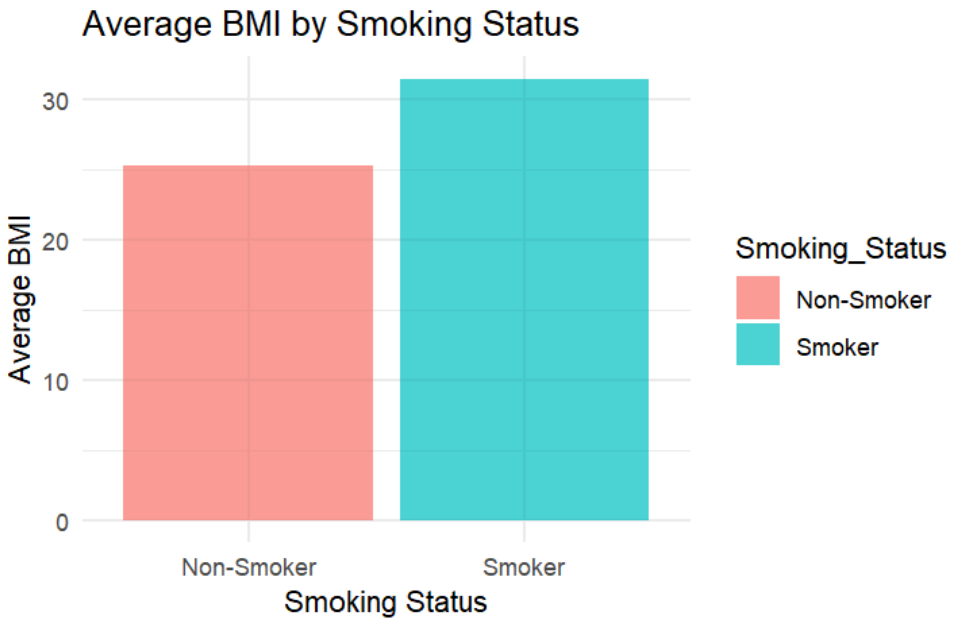
Scatter plot: Relationship between BMI and Daily Calorie Intake

```
ggplot(obesity, aes(x = Daily_Calorie_Intake, y = BMI)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "Scatter Plot of BMI vs Daily Calorie Intake",
       x = "Daily Calorie Intake",
       y = "BMI") +
  theme_minimal() +
  geom_smooth(method = "lm", color = "red")
```



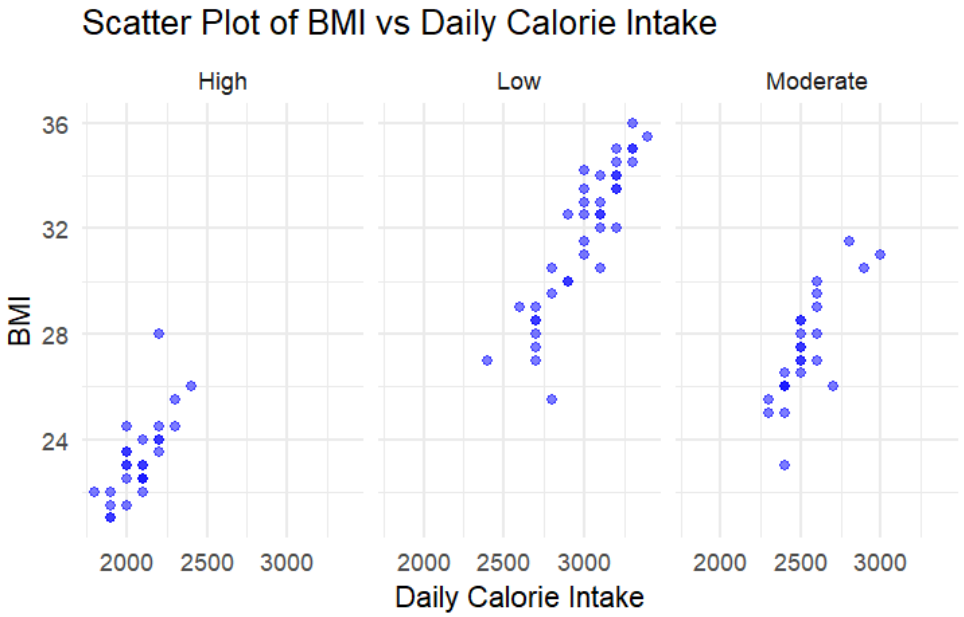
Barplot: Average BMI by Smoking Status

```
ggplot(obesity, aes(x = Smoking_Status, y = BMI, fill = Smoking_Status)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge", alpha = 0.7) +
  labs(title = "Average BMI by Smoking Status",
       x = "Smoking Status",
       y = "Average BMI") +
  theme_minimal()
```



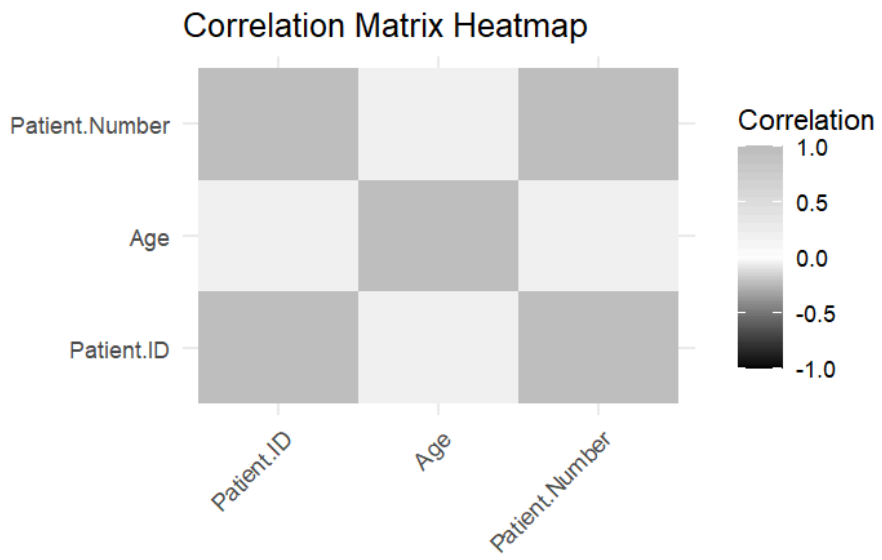
Faceted Scatter Plots: BMI vs Daily Calorie Intake Faceted by Physical Activity

```
ggplot(obesity, aes(x = Daily_Calorie_Intake, y = BMI)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "Scatter Plot of BMI vs Daily Calorie Intake",
       x = "Daily Calorie Intake",
       y = "BMI") +
  theme_minimal() +
  facet_wrap(~Physical_Activity_Level)
```



Correlation Heatmap: You can visualize the correlation between numeric variables using a heatmap.

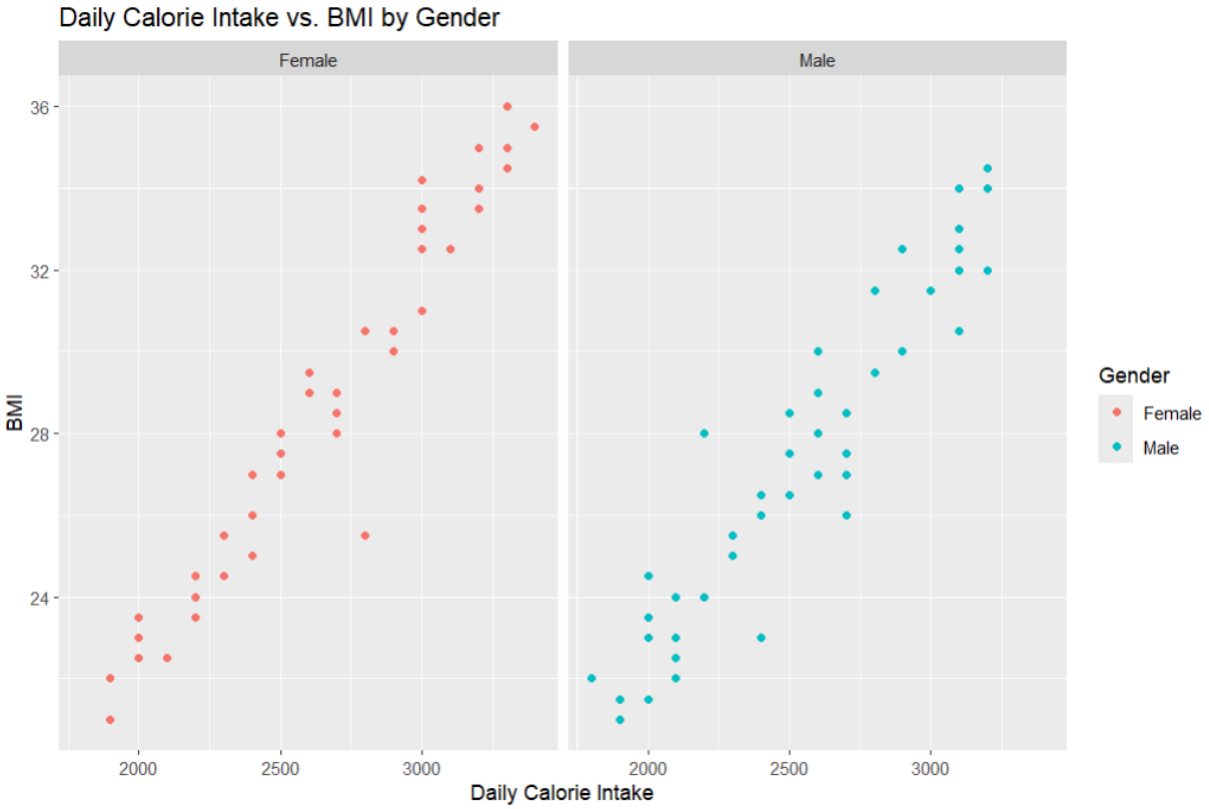
```
cor_matrix <- cor(obesity[, c("Age", "BMI", "Daily Calorie Intake")])  # Select numeric columns
cor_melted <- melt(cor_matrix)
ggplot(cor_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "black", high = "grey", mid = "white",
                       limit = c(-1, 1), name="Correlation") +
  theme_minimal() +
  labs(title = "Correlation Matrix Heatmap", x = "", y = "") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



MULTIVARIATE: Multivariate analysis examines the interactions between multiple variables simultaneously, allowing for a deeper understanding of complex relationships in the data. It can involve techniques like multiple regression, principal component analysis (PCA), or multivariate visualizations.

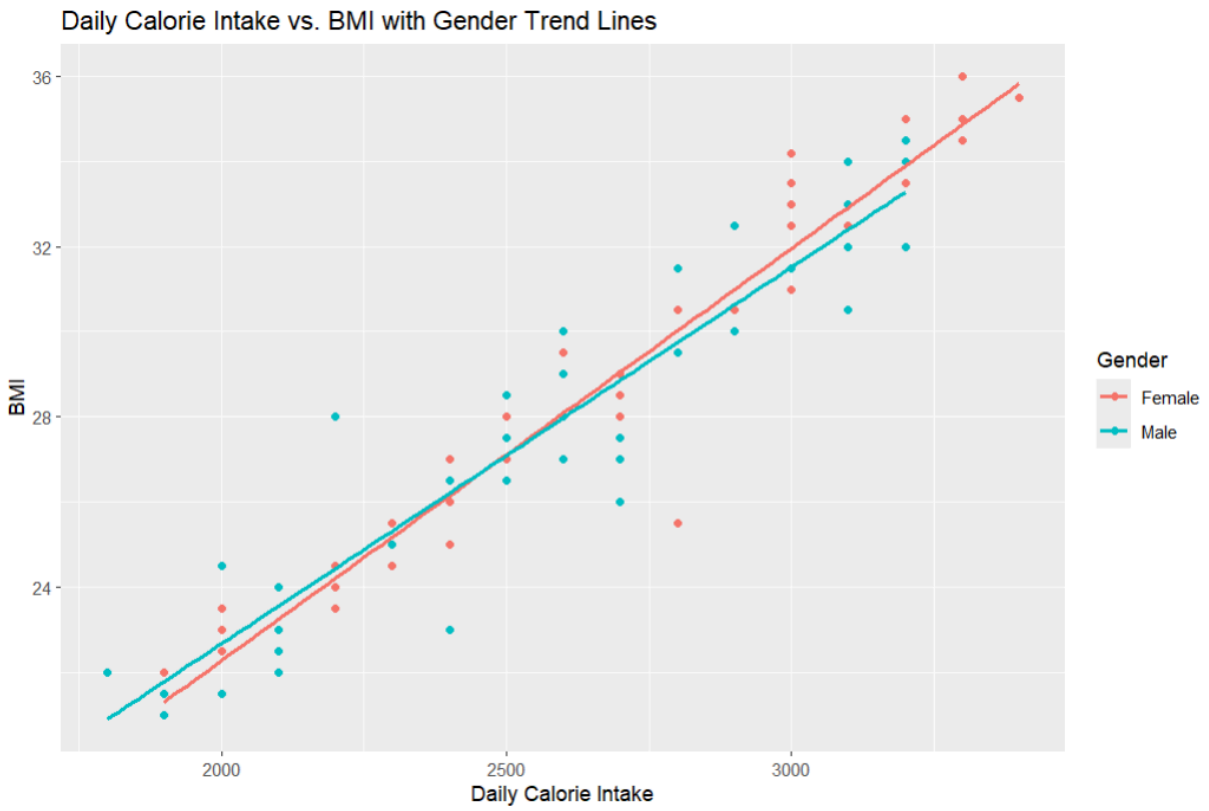
Scatter plot with faceting by Gender

```
library(ggplot2)
ggplot(public, aes(x = Daily_Calorie_Intake, y = BMI, color = Gender)) +
  geom_point() +
  facet_grid(~Gender) +
  ggtitle("Daily Calorie Intake vs. BMI by Gender") +
  xlab("Daily Calorie Intake") +
  ylab("BMI")
```



Smooth line plot with grouping by Gender

```
ggplot(public, aes(x = Daily_Calorie_Intake, y = BMI, color = Gender)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +  # Add linear trend lines
  ggtitle("Daily Calorie Intake vs. BMI with Gender Trend Lines") +
  xlab("Daily Calorie Intake") +
  ylab("BMI")
```



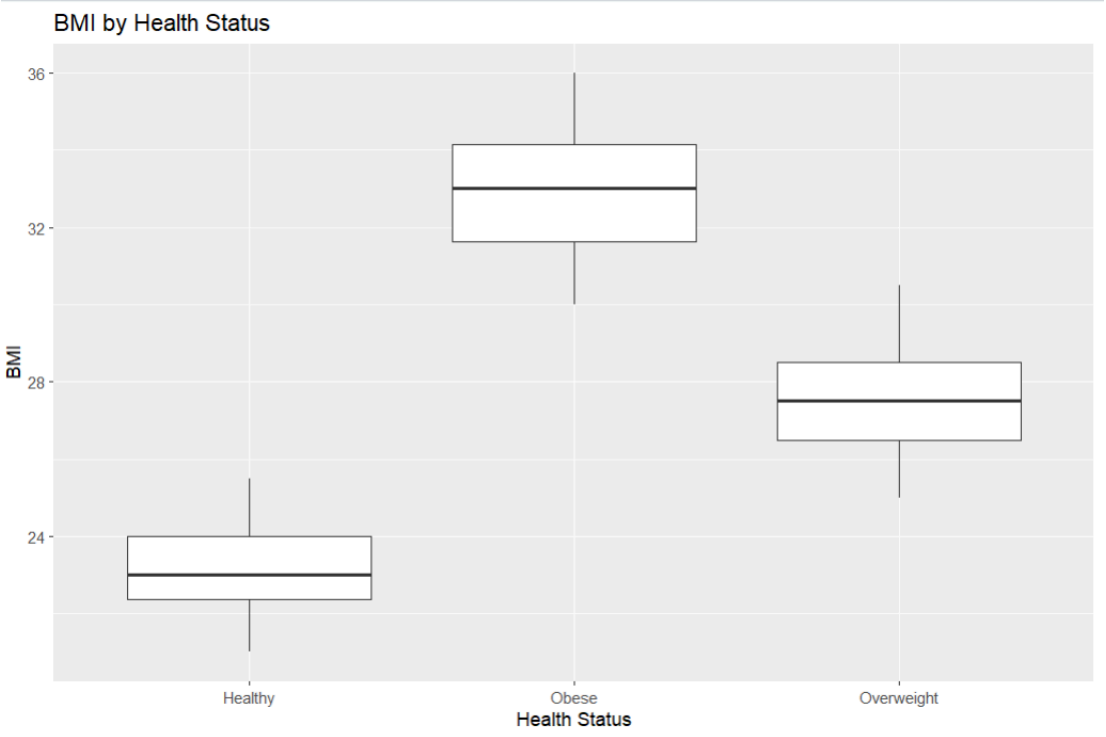
7. Data Visualization :

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualisation tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Boxplot for BMI by Health Status:

The code creates a boxplot using `ggplot2` to visualize the distribution of BMI across different Health Status categories. Each box represents the range of BMI values for a specific Health Status, showing the median, quartiles, and potential outliers. The plot helps identify differences in BMI distribution across health groups.

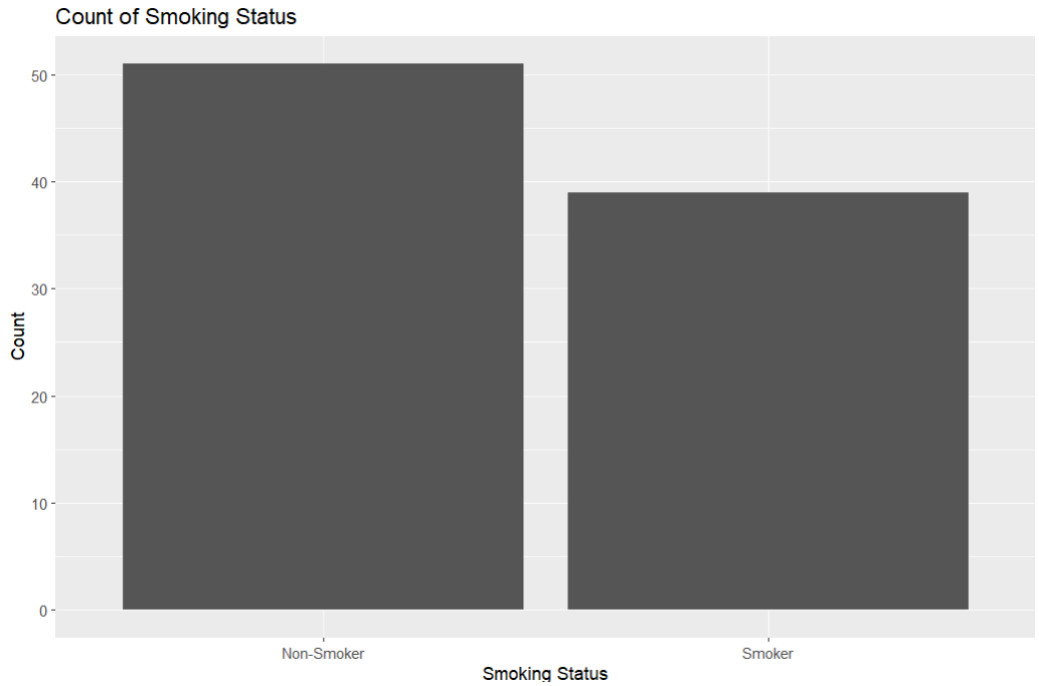
```
library(ggplot2)
ggplot(public, aes(x = Health_Status, y = BMI)) +
  geom_boxplot() +
  ggtitle("BMI by Health Status") +
  xlab("Health Status") +
  ylab("BMI")
```



Bar plot for Smoking Status

The code generates a bar chart to display the count of individuals in each Smoking Status category using `ggplot2`. The x-axis represents different smoking statuses, while the y-axis shows the corresponding counts. The plot includes a title and axis labels for clarity.

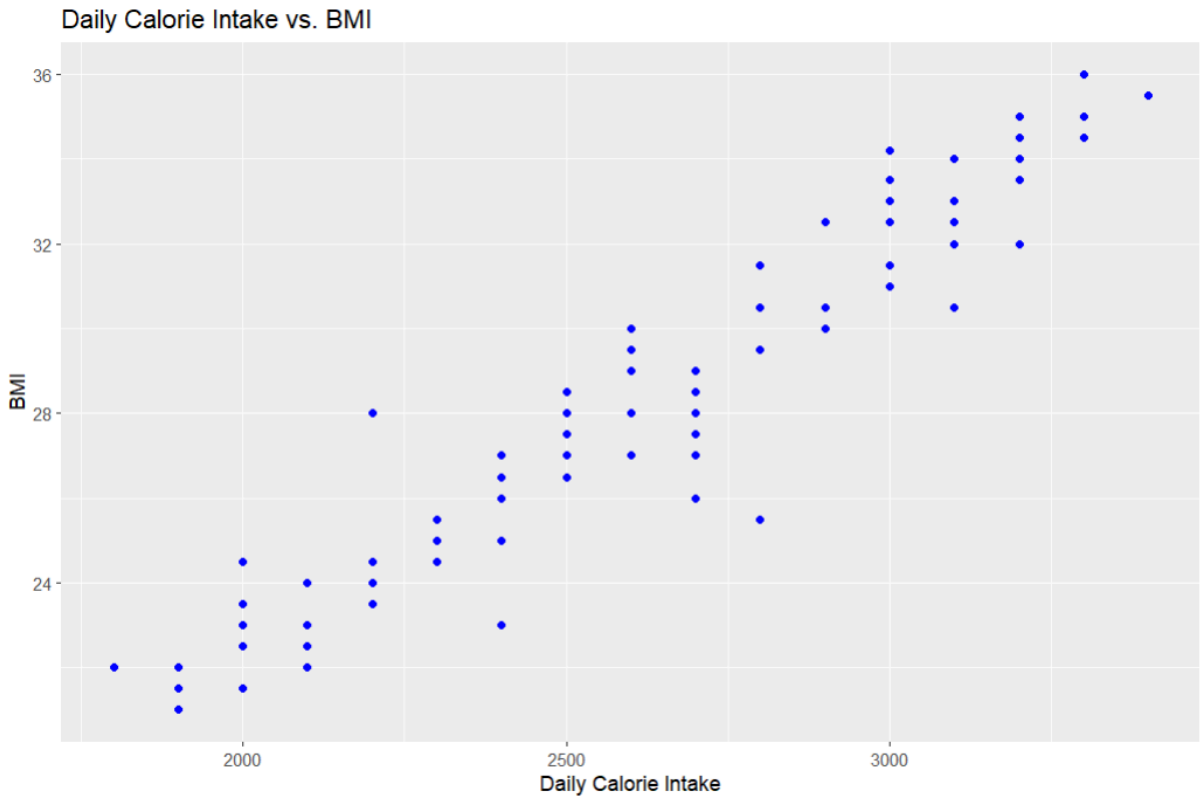
```
ggplot(public, aes(x = Smoking_Status)) +
  geom_bar() +
  ggtitle("Count of Smoking Status") +
  xlab("Smoking Status") +
  ylab("Count")
```



Scatter plot for Daily Calorie Intake vs. BMI

The code creates a scatter plot using ggplot2 to visualize the relationship between Daily Calorie Intake and BMI. Data points are displayed in blue, with Daily Calorie Intake on the x-axis and BMI on the y-axis. The plot is titled and labeled to provide clear context for the variables.

```
ggplot(health_data, aes(x = Daily_Calorie_Intake, y = BMI)) +
  geom_point(color = "blue") +
  ggtitle("Daily Calorie Intake vs. BMI") +
  xlab("Daily Calorie Intake") +
  ylab("BMI")
```



8. Descriptive Statistics :

Calculate summary statistics like mean, median, standard deviation.
The summary statistics provide insights into the data distribution for three variables: BMI, Daily Calorie Intake, and Sleep Hours. For BMI, the mean is approximately 28, with a standard deviation of 4.22, indicating moderate variability. Daily Calorie Intake has a mean around 2593, with a standard deviation of 433.9, showing wider variability. Sleep Hours have a mean of 6.5 and lower variability, with a standard deviation of 1.12. Median values are close to the means for all variables, suggesting near-normal distribution. Variances further confirm the spread of each variable

Summary statistics for BMI

summary(health_data$BMI)

```
> summary(public$BMI)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  21.00   24.50   27.75   28.00   31.50   36.00
```

Summary statistics for Daily Calorie Intake

summary(health_data$Daily_Calorie_Intake)

```
> summary(public$Daily_Calorie_Intake)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
   1800    2200    2600    2593    3000    3400
```

Summary statistics for Sleep Hours

summary(health_data$Sleep_Hours)

```
> summary(public$Sleep_Hours)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    5.0     6.0     6.0     6.5     8.0     8.0
```

Mean of BMI:

mean(public$BMI)

```
> mean(public$BMI)
[1] 27.99667
```

Mean of Daily Calorie Intake:

mean(public$Daily_Calorie_Intake)

```
> mean(public$Daily_Calorie_Intake)
[1] 2593.333
>
```

Mean of Sleep Hours:

mean(public$Sleep_Hours)

```
> mean(public$Sleep_Hours)
[1] 6.5
```

Median of BMI

median(health_data$BMI)

```
> median(public$BMI)
[1] 27.75
```

 # Median of Daily Calorie Intake

median(health_data$Daily_Calorie_Intake)

```
> median(public$Daily_Calorie_Intake)
[1] 2600
```

 # Median of Sleep Hours

median(health_data$Sleep_Hours)

```
> median(public$Sleep_Hours)
[1] 6
```

 # Standard deviation of BMI

sd(health_data$BMI)

```
> sd(public$BMI)
[1] 4.216992
```

 # Standard deviation of Daily Calorie Intake

sd(health_data$Daily_Calorie_Intake)

```
> sd(public$Daily_Calorie_Intake)
[1] 433.9005
```

 # Standard deviation of Sleep Hours

sd(health_data$Sleep_Hours)

```
> sd(public$Sleep_Hours)
[1] 1.124298
```

 # Variance of BMI

var(health_data$BMI)

```
> var(public$BMI)
[1] 17.78302
```

 # Variance of Daily Calorie Intake

var(health_data$Daily_Calorie_Intake)

```
> var(public$Daily_Calorie_Intake)
[1] 188269.7
```

 # Variance of Sleep Hours

var(health_data$Sleep_Hours)

```
> var(public$Sleep_Hours)
[1] 1.264045
```

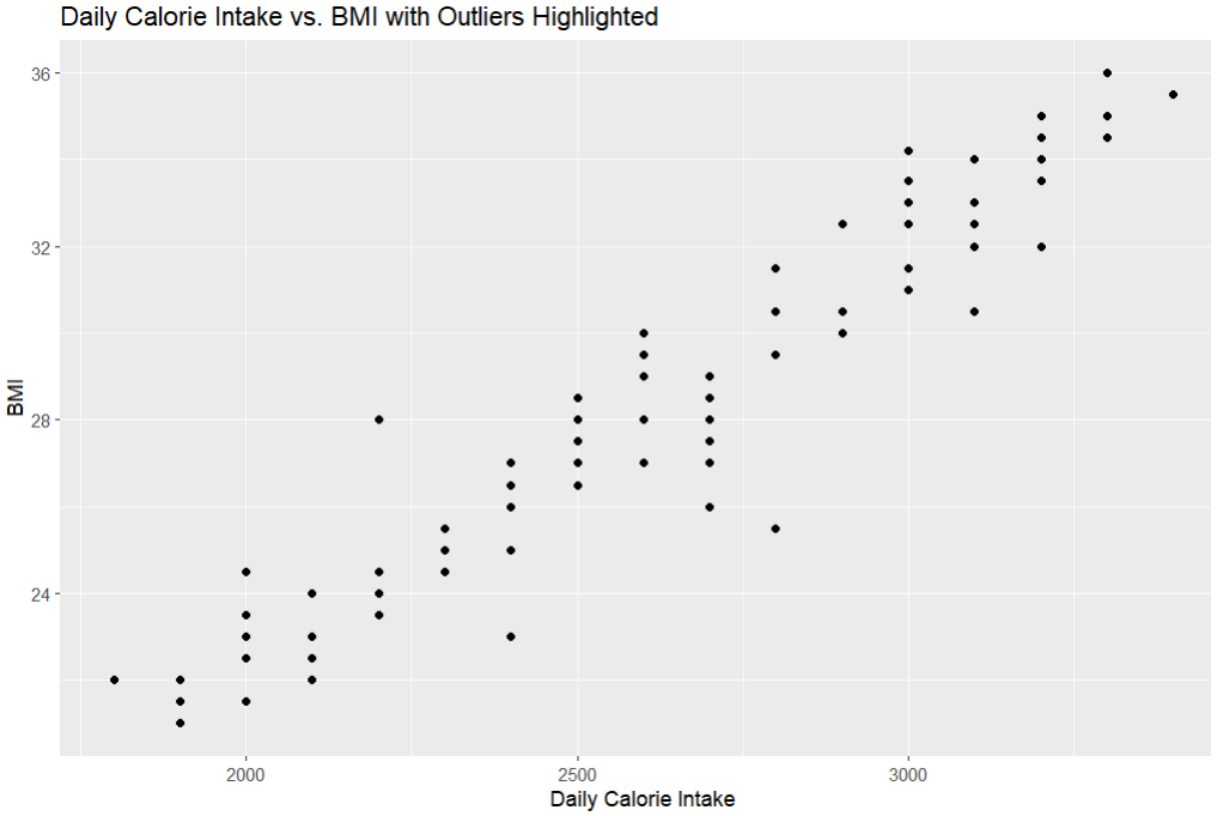
9. Identify Patterns and Outliers

Use visualizations and statistical methods to identify patterns or outliers in the data.

The code uses R's dplyr and ggplot2 packages to analyze and visualize data. It first identifies outliers by filtering rows where "Daily_Calorie_Intake" is either above 3500 or below 1500. Then, it creates a scatter plot with "Daily_Calorie_Intake" on the x-axis and "BMI" on the y-axis, using ggplot2. The main plot shows all data points, while outliers are highlighted in red for distinction. The plot is titled and labeled for clarity.

```
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr))
```

```
outliers <- public %>% filter(Daily_Calorie_Intake > 3500 |
Daily_Calorie_Intake < 1500)
ggplot(public, aes(x = Daily_Calorie_Intake, y = BMI)) +
  geom_point() +  # Regular points
  geom_point(data = outliers, aes(x = Daily_Calorie_Intake, y = BMI), color =
"red", size = 3) +  # Highlight outliers
  ggtitle("Daily Calorie Intake vs. BMI with Outliers Highlighted") +
  xlab("Daily Calorie Intake") +
  ylab("BMI")
```



10. Hypothesis Testing

The t.test() function compares the means of two groups (smokers vs. non-smokers) for the BMI variable. It tests the null hypothesis that there is no significant difference in mean BMI between smokers and non-smokers. The output provides the test statistic (t-value), degrees of freedom, p-value, and confidence interval. A low p-value (typically < 0.05) indicates a significant difference in mean BMI between the two groups. If the p-value is higher, it suggests no statistically significant difference.

```
# Hypothesis test: t-test to compare BMI between smokers and
non-smokers
library(dplyr)
t_test_result <- t.test(BMI ~ Smoking_Status, data = public)
t_test_result
```

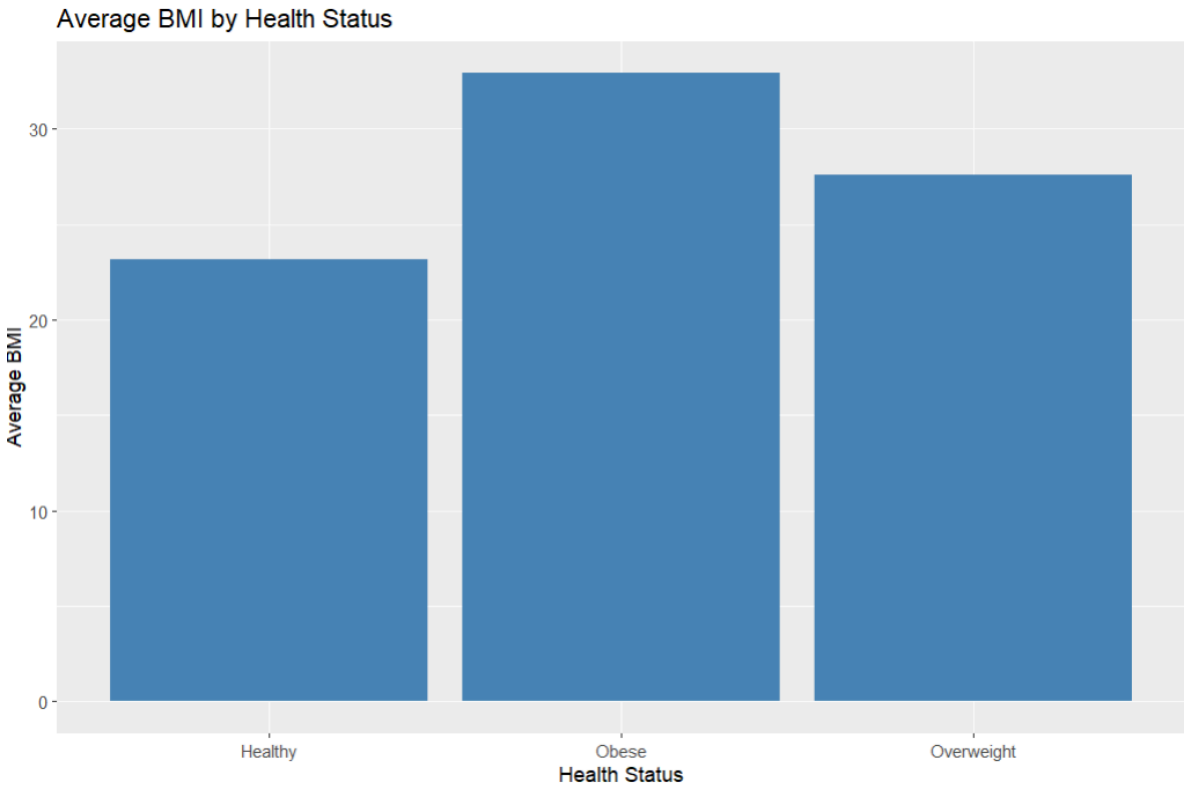
```
> t_test_result

      Welch Two Sample t-test

data:  BMI by Smoking_Status
t = -10.083, df = 84.521, p-value = 3.716e-16
alternative hypothesis: true difference in means between group Non-Smoker
and group Smoker is not equal to 0
95 percent confidence interval:
 -7.385322 -4.952234
sample estimates:
mean in group Non-Smoker     mean in group Smoker
                25.32353                 31.49231
```

11. Data Summarization :

Summarize key findings and insights using tables and visualizations.

This code calculates and visualizes the average BMI across different health statuses. Using dplyr, it groups the data by Health_Status and computes the mean BMI for each group. The resulting summary is stored in avg_bmi_summary. The ggplot2 code creates a bar plot with Health_Status on the x-axis and the average BMI on the y-axis, using a steel blue color to fill the bars. The plot is titled "Average BMI by Health Status" and has labeled axes for clarity

```
library(dplyr)
library(ggplot2)
avg_bmi_summary <- public %>%
  group_by(Health_Status) %>%
  summarise(avg_BMI = mean(BMI))
avg_bmi_summary
ggplot(avg_bmi_summary, aes(x = Health_Status, y = avg_BMI)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  ggtitle("Average BMI by Health Status") +
  xlab("Health Status") +
  ylab("Average BMI")
```


Average BMI by Health Status
Average BMI
30
20
10
0
Healthy
Obese
Overweight
Health Status