

# Lifestyle Health Analysis

```
In [4]: import numpy as np
```

```
In [6]: import pandas as pd
```

```
In [108... lifestyle_data = pd.read_csv("C:/Users/digga/OneDrive/Desktop/power bi/bi2/synth
```

```
In [ ]: The dataset contains synthetic records of individuals, including lifestyle habit
```

## Phase 1: Dataset Understanding and Objective Framing

GOAL : Understand the structure, meaning, and potential of the dataset in order to define a clear business or health problem to solve.

### 1.1 what each column means?

Understand the meaning and role of each column to ensure accurate interpretation during analysis and visualization. This helps in identifying data types, potential groupings, and which features influence health indicators like BMI.

```
In [16]: lifestyle_data
```

Out[16]:

|      | ID   | Age | Gender | Height_cm | Weight_kg | BMI  | Smoker | Exercise_Freq     | Diet_Qu |
|------|------|-----|--------|-----------|-----------|------|--------|-------------------|---------|
| 0    | 1    | 56  | Other  | 177.6     | 37.3      | 11.8 | Yes    | NaN               |         |
| 1    | 2    | 69  | Other  | 169.3     | 70.7      | 24.7 | No     | 1-2<br>times/week |         |
| 2    | 3    | 46  | Female | 159.1     | 69.0      | 27.3 | No     | Daily             | Exc     |
| 3    | 4    | 32  | Male   | 170.6     | 76.4      | 26.3 | No     | 3-5<br>times/week | Exc     |
| 4    | 5    | 60  | Male   | 158.4     | 60.4      | 24.1 | No     | 3-5<br>times/week | Exc     |
| ...  | ...  | ... | ...    | ...       | ...       | ...  | ...    | ...               |         |
| 7495 | 7496 | 55  | Other  | 168.3     | 52.4      | 18.5 | Yes    | 1-2<br>times/week | Av      |
| 7496 | 7497 | 24  | Male   | 179.1     | 58.8      | 18.3 | No     | 3-5<br>times/week | Exc     |
| 7497 | 7498 | 61  | Other  | 160.2     | 80.0      | 31.2 | No     | 3-5<br>times/week |         |
| 7498 | 7499 | 40  | Female | 172.6     | 66.1      | 22.2 | No     | 1-2<br>times/week |         |
| 7499 | 7500 | 58  | Other  | 163.8     | 59.3      | 22.1 | No     | Daily             | Exc     |

7500 rows × 13 columns



In [18]: #There are 7500 rows and 13 columns in the dataset. Most columns are numerical e

# Below is a brief explanation of each column in the dataset:

```
# - `User_ID`: Unique identifier for each person in the dataset
# - `Age`: Age of the individual (in years)
# - `Gender`: Gender identity of the person (e.g., Male, Female, Other)
# - `Height_cm`: Height in centimeters
# - `Weight_kg`: Weight in kilograms
# - `BMI`: Body Mass Index, calculated using height and weight
# - `Smoking_Status`: Indicates if the person smokes (Yes or No)
# - `Physical_Activity_Level`: Frequency of physical activity (e.g., Daily, 3-5
# - `Diet_Quality`: Self-rated diet quality (e.g., Excellent, Good, Poor)
# - `Sleep_Hours`: Average hours of sleep per night
# - `Water_Intake_Liters`: Liters of water consumed per day
# - `Alcohol_Consumption`: Whether the person consumes alcohol (Yes or No)
# - `Stress_Level`: Self-reported stress level (e.g., Low, Moderate, High)
```

## 1.2 What's the shape and size of the Data ?

Get a high-level view of the dataset's structure, including the total number of rows (individuals) and columns (features), to estimate the scale of analysis and memory usage.

```
In [23]: print("Shape of the dataset :", lifestyle_data.shape)
```

```
Shape of the dataset : (7500, 13)
```

```
In [25]: lifestyle_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7500 entries, 0 to 7499
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID                     7500 non-null   int64
1   Age                   7500 non-null   int64
2   Gender                 7500 non-null   object
3   Height_cm             7500 non-null   float64
4   Weight_kg             7500 non-null   float64
5   BMI                   7500 non-null   float64
6   Smoker                7500 non-null   object
7   Exercise_Freq         5621 non-null   object
8   Diet_Quality          7500 non-null   object
9   Alcohol_Consumption   5608 non-null   object
10  Chronic_Disease       7500 non-null   object
11  Stress_Level          7500 non-null   int64
12  Sleep_Hours           7500 non-null   float64
dtypes: float64(4), int64(3), object(6)
memory usage: 761.8+ KB
```

### 1.3 What Is the Business or Health Question I Want to Answer?

Define a clear and focused health analysis objective based on the dataset. This helps guide future cleaning, feature creation, visualizations, and insights toward a meaningful outcome such as identifying risk factors, lifestyle patterns, or health recommendations.

```
In [ ]: # This project explores the relationship between lifestyle behaviors and health

# By identifying behavioral patterns and building a custom health risk scoring s

# ---

# ### Key Questions to Explore:
# - How do lifestyle habits like sleep, alcohol, stress, smoking, and diet impac
# - Are there specific combinations of behaviors that define healthy vs unhealth
# - Can we classify users into meaningful risk categories such as "Healthy", "At
# - Which lifestyle factors are most strongly associated with obesity or stress?

# ---

# ### Final Problem Statement:

# > Analyze how lifestyle choices such as sleep, diet, physical activity, alcoho
# > Use behavioral features to identify high-risk individuals and build a custom
```

### 1.4 Are there Multiple Types of Users (by Gender/Age/Habit)?

## Identify distinct user segments based on categorical attributes like gender, age groups

```
In [ ]: # Yes – the dataset includes a diverse population with multiple user types that

# ---

# ### Demographic Segments:
# - Gender → Male, Female, Other
# - Age Group → Categorized into:
#   - Child (<18)
#   - Young Adult (18–30)
#   - Adult (31–50)
#   - Senior (51+)

# ---

# ### Lifestyle-Based Segments:

# - Smoking Status → Smoker vs Non-Smoker
# - Alcohol Consumption → High / Moderate / Low / Unknown
# - Physical Activity → Sedentary / Moderate / Active / Not Reported
# - Diet Quality → Excellent / Good / Poor / Average
# - Sleep Duration → Short (<6 hrs) / Optimal (6–8 hrs) / Long (>8 hrs)
# - Stress Level → Scaled from 1 (Low) to 10 (High)

# ---

# ### Why Segmentation Matters:
# - Helps build user profiles like “Sleep-Deprived Smokers” or “Fit But Stressed”
# - Enables focused analysis of BMI, stress, and chronic disease risk by group
# - Supports data storytelling and stakeholder reporting

# ---

# → These segments will drive the creation of visualizations, health risk scorin
```

## 1.5 What Patterns Might Exist Between Lifestyle Choices and BMI?

Explore potential correlations between lifestyle factors (e.g., exercise, smoking, alcohol, diet, sleep) and BMI levels to uncover health risks, behavioral trends, and target areas for preventive care or intervention.

```
In [42]: # Lifestyle Patterns Related to BMI

# This section explores the potential relationships between lifestyle habits and

# ---

# #### Hypothesized Patterns:

# - Sleep Duration and BMI
#   → Both insufficient (<6 hours) and excessive sleep (>8 hours) may be linked

# - Water Intake and BMI
#   → Individuals with higher water intake are likely to have better hydration a
```

```

# - Physical Activity and BMI
#   → Higher frequency of exercise (3–5 times/week or daily) is generally associ

# - Smoking and Alcohol Consumption vs BMI
#   → Smoking may suppress appetite (leading to lower BMI), whereas high alcohol

# - Diet Quality and BMI
#   → Users reporting poor or average diet quality are expected to show higher B

# - Stress Levels and BMI
#   → Chronic stress can lead to emotional eating or sleep disturbances, both of

# ---

# These hypotheses will be tested in the next phase using univariate and bivariable

```

## Phase 2: Data Cleaning and Preprocessing

GOAL : Clean the dataset by handling missing values, correcting inconsistencies, and creating new features for better analysis.

### 2.1 checking for missing values in each column

This step is important because missing values can mess up your graphs, calculations, or even ML models later.

```
In [48]: lifestyle_data.isnull().sum()
```

```

Out[48]: ID                0
         Age                0
         Gender             0
         Height_cm          0
         Weight_kg          0
         BMI                0
         Smoker             0
         Exercise_Freq      1879
         Diet_Quality        0
         Alcohol_Consumption 1892
         Chronic_Disease     0
         Stress_Level        0
         Sleep_Hours         0
         dtype: int64

```

Check how many rows have Height or Weight as 0 (invalid)

```
In [51]: print("Height = 0:", (lifestyle_data['Height_cm'] == 0).sum())
         print("Weight = 0:", (lifestyle_data['Weight_kg'] == 0).sum())
         print("BMI = 0:", (lifestyle_data['BMI'] == 0).sum())
```

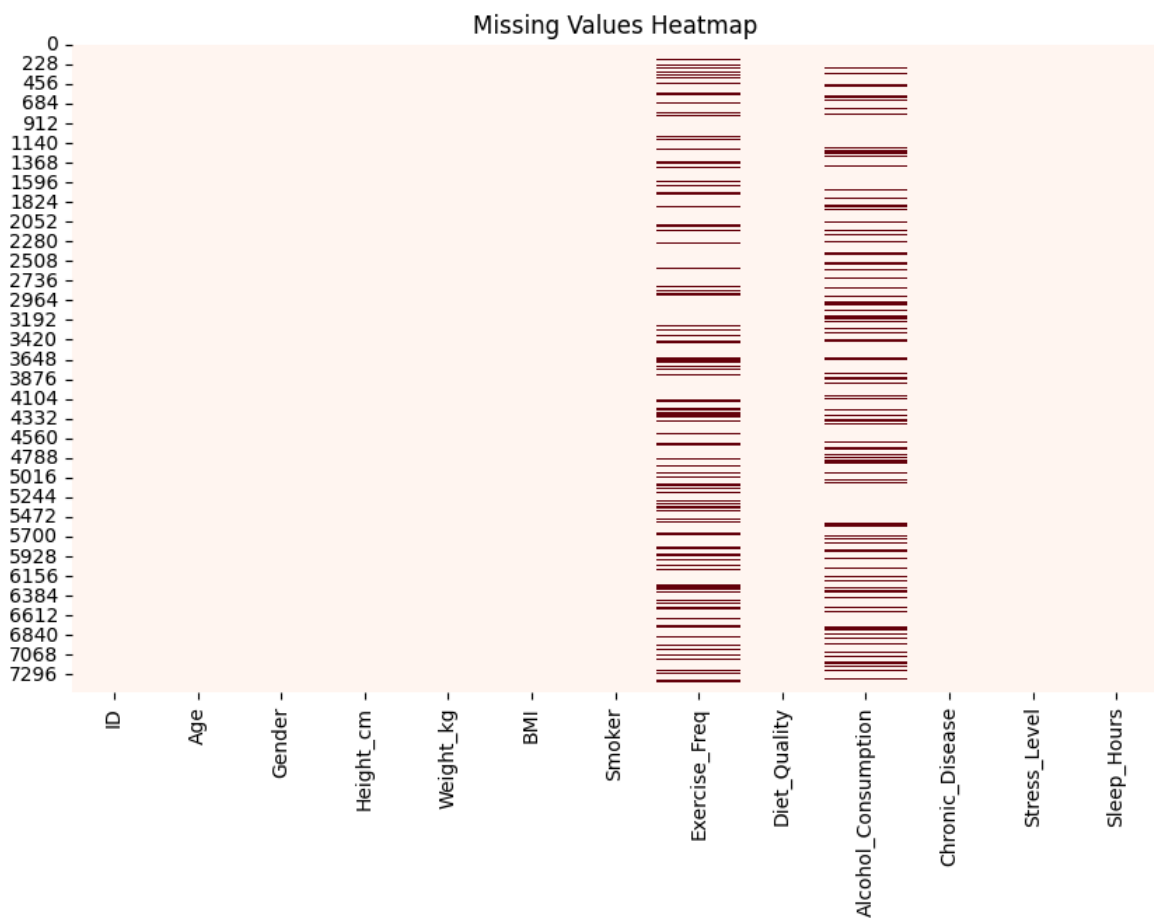
```

Height = 0: 0
Weight = 0: 0
BMI = 0: 0

```

```
In [53]: import seaborn as sns
         import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10,6))
sns.heatmap(lifestyle_data.isnull(), cbar=False, cmap="Reds")
plt.title("Missing Values Heatmap")
plt.savefig("Images/missing_values_heatmap.png")
plt.show()
plt.close()
```



## 2.2 Are There Any Duplicate Rows in the Dataset?

Duplicate records can distort insights, inflate counts, or mislead pattern detection. So before moving ahead, we'll check if any exact duplicate rows exist.

```
In [57]: lifestyle_data.duplicated().sum()
```

```
Out[57]: 0
```

```
In [59]: # We checked for exact duplicate entries using the `.duplicated().sum()` method.
# Result: The dataset contains 0 duplicate rows, indicating that each record is
# No data removal was needed in this step.
```

## 2.3 Are column names and data types clean and consistent?

Check if any column names need formatting (like extra spaces, inconsistent casing). Ensure all columns have the correct data types (e.g., numerical as int/float, categories as object)

```
In [63]: lifestyle_data.columns
```

```
Out[63]: Index(['ID', 'Age', 'Gender', 'Height_cm', 'Weight_kg', 'BMI', 'Smoker',  
              'Exercise_Freq', 'Diet_Quality', 'Alcohol_Consumption',  
              'Chronic_Disease', 'Stress_Level', 'Sleep_Hours'],  
             dtype='object')
```

```
In [65]: # We reviewed the dataset's column names and data types.  
  
# Column Names:  
# All column names are already well-structured and readable (e.g., `Age`, `Gende  
# No renaming or formatting was required.  
  
# Data Types:  
# - Numerical columns such as `Age`, `BMI`, `Height_cm`, and `Sleep_Hours` are s  
# - Categorical columns like `Gender`, `Smoker`, and `Stress_Level` are stored a  
  
# The data is clean and ready for further analysis.
```

## 2.4 Normalize Categorical Values

Ensure consistent formatting like 'Male' vs 'male', 'Yes' vs 'yes', etc.

```
In [70]: print("Gender:", lifestyle_data['Gender'].unique())
```

```
Gender: ['Other' 'Female' 'Male']
```

```
In [72]: print("smoker :", lifestyle_data['Smoker'].unique())
```

```
smoker : ['Yes' 'No']
```

```
In [74]: print("Exercise_Freq :", lifestyle_data['Exercise_Freq'].unique())
```

```
Exercise_Freq : [nan '1-2 times/week' 'Daily' '3-5 times/week']
```

```
In [76]: print("Diet_Quality : ", lifestyle_data['Diet_Quality'].unique())
```

```
Diet_Quality : ['Poor' 'Good' 'Excellent' 'Average']
```

```
In [78]: print("Alcohol_Consumption : " , lifestyle_data['Alcohol_Consumption'].unique())
```

```
Alcohol_Consumption : [nan 'High' 'Moderate' 'Low']
```

```
In [80]: print("Chronic_Disease : " , lifestyle_data['Chronic_Disease'].unique())
```

```
Chronic_Disease : ['No' 'Yes']
```

```
In [82]: print("Stress_Level:", lifestyle_data['Stress_Level'].unique())
```

```
Stress_Level: [ 9  2  3  6  1  7 10  4  5  8]
```

```
In [84]: cols_to_clean = ['Gender', 'Smoker', 'Exercise_Freq', 'Diet_Quality',  
                        'Alcohol_Consumption', 'Chronic_Disease', 'Stress_Level']
```

```
for col in cols_to_clean :  
    if lifestyle_data[col].dtype == 'object':  
        lifestyle_data[col] = lifestyle_data[col].str.strip().str.lower()
```

```
In [86]: for col in ['Gender', 'Smoker', 'Exercise_Freq', 'Diet_Quality',  
                  'Alcohol_Consumption', 'Chronic_Disease', 'Stress_Level']:
```

```
print(f"{col} → {lifestyle_data[col].unique()}")
```

Gender → ['other' 'female' 'male']

Smoker → ['yes' 'no']

Exercise\_Freq → [nan '1-2 times/week' 'daily' '3-5 times/week']

Diet\_Quality → ['poor' 'good' 'excellent' 'average']

Alcohol\_Consumption → [nan 'high' 'moderate' 'low']

Chronic\_Disease → ['no' 'yes']

Stress\_Level → [ 9 2 3 6 1 7 10 4 5 8]

```
In [88]: # We normalized all key categorical columns by converting their values to lowercase

# Cleaned Columns:
# - `gender`, `smoker`, `exercise_freq`, `diet_quality`
# - `alcohol_consumption`, `chronic_disease`, `stress_level`
#
# The `Diet_Quality` column had inconsistent abbreviations like `exc`, `ave`
# We standardized these using a mapping to their full forms:

# - `exc` → `excellent`
# - `ave` → `average`
# - `unk` → `unknown`

# This prevents issues like `Male` vs `male` being treated as different categories
# The categorical data is now clean, consistent, and ready for grouped analysis.
```

## 2.5 Feature Engineering

Add new , insightful columns based on existing data to help deeper analysis

### A. AGE GROUPING

```
In [110]: def age_group(Age):
            if (Age < 18) :
                return 'child'
            elif (18 <= Age <= 30):
                return 'younge_adult'
            elif (31 <= Age <= 50):
                return 'young_adult'
            else:
                return 'senior'

lifestyle_data['Age_group'] = lifestyle_data['Age'].apply(age_group)
```

### B. CREATE BMI CATEGORIES

```
In [112]: def bmi_category(BMI):
            if (BMI < 18.5):
                return 'underweight'
            elif (18.5 <= BMI < 25):
                return 'normal'
            elif (25 <= BMI < 30):
                return 'overweight'
            else:
                return 'obese'
```

```
lifestyle_data['BMI_Category'] = lifestyle_data['BMI'].apply(bmi_category)
```

### C. CATEGORIZE SLEEP QUALITY

```
In [114... def sleep_type(Sleep_Hours):
    if (Sleep_Hours <6) :
        return 'short_sleep'
    elif (6 <= Sleep_Hours <= 8):
        return 'optimal_sleep'
    else:
        return 'long_sleep'

lifestyle_data['sleep_type'] = lifestyle_data['Sleep_Hours'].apply(sleep_type)
```

```
In [116... print("Final Shape:", lifestyle_data.shape)
print("Final Columns:", lifestyle_data.columns)
lifestyle_data.head()
```

Final Shape: (7500, 16)

Final Columns: Index(['ID', 'Age', 'Gender', 'Height\_cm', 'Weight\_kg', 'BMI', 'Smoker', 'Exercise\_Freq', 'Diet\_Quality', 'Alcohol\_Consumption', 'Chronic\_Disease', 'Stress\_Level', 'Sleep\_Hours', 'Age\_group', 'BMI\_Category', 'sleep\_type'], dtype='object')

```
Out[116... 
```

|   | ID | Age | Gender | Height_cm | Weight_kg | BMI  | Smoker | Exercise_Freq  | Diet_Quality |
|---|----|-----|--------|-----------|-----------|------|--------|----------------|--------------|
| 0 | 1  | 56  | Other  | 177.6     | 37.3      | 11.8 | Yes    | NaN            | Poor         |
| 1 | 2  | 69  | Other  | 169.3     | 70.7      | 24.7 | No     | 1-2 times/week | Good         |
| 2 | 3  | 46  | Female | 159.1     | 69.0      | 27.3 | No     | Daily          | Excellent    |
| 3 | 4  | 32  | Male   | 170.6     | 76.4      | 26.3 | No     | 3-5 times/week | Excellent    |
| 4 | 5  | 60  | Male   | 158.4     | 60.4      | 24.1 | No     | 3-5 times/week | Excellent    |

## PHASE 3: Exploratory Data Analysis (EDA)

GOAL: Explore the data through visualizations to identify patterns, trends, and relationships between health and lifestyle factors.

```
In [120... import matplotlib.pyplot as plt
import seaborn as sns
```

### 3.1 What is the distribution of key numerical features ?

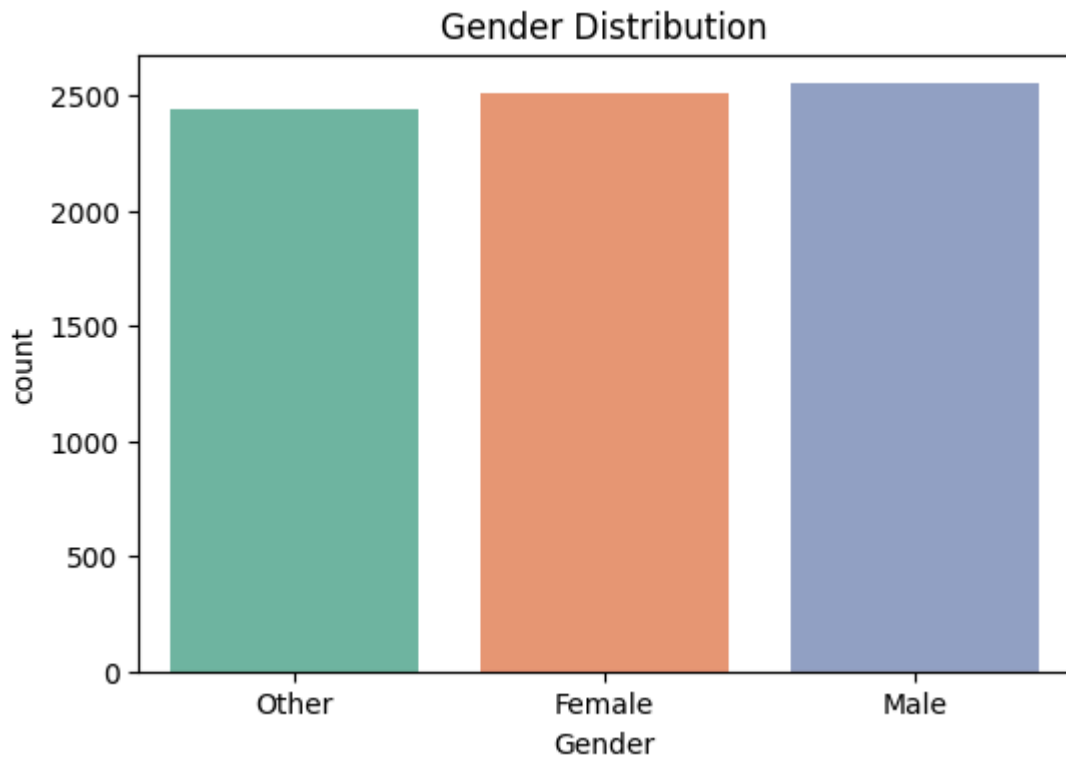
#### TYPE : UNIVARIATE ANALYSIS

To understand the central tendencies, spread, and shape of important numerical variables such as Sleep\_Hours, BMI, and Age.

This univariate analysis helps identify patterns, outliers, and skewness in individual variables, setting the foundation for deeper multivariate insights.

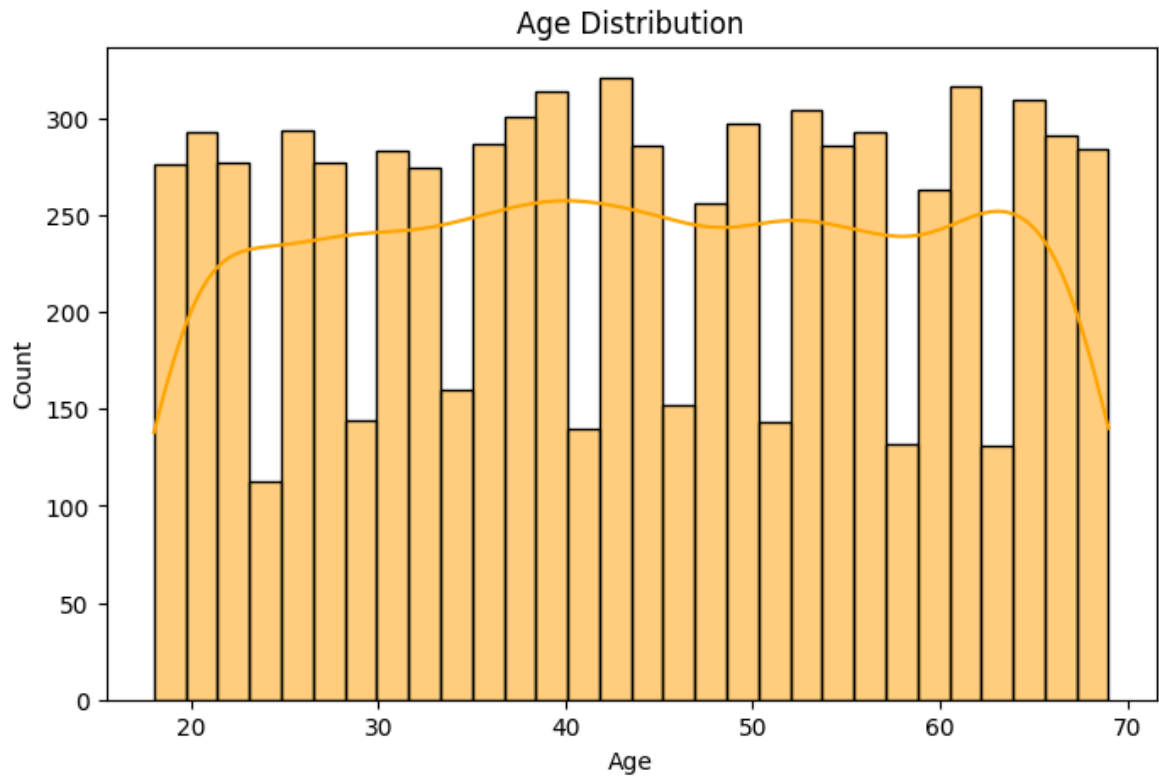
#### A. GENDER DISTRIBUTION

```
In [129... plt.figure(figsize=(6,4))
sns.countplot(data=lifestyle_data, x='Gender', hue = 'Gender', palette='Set2')
plt.title("Gender Distribution")
plt.savefig("Images/gender_distribution.png")
plt.show()
plt.close()
```



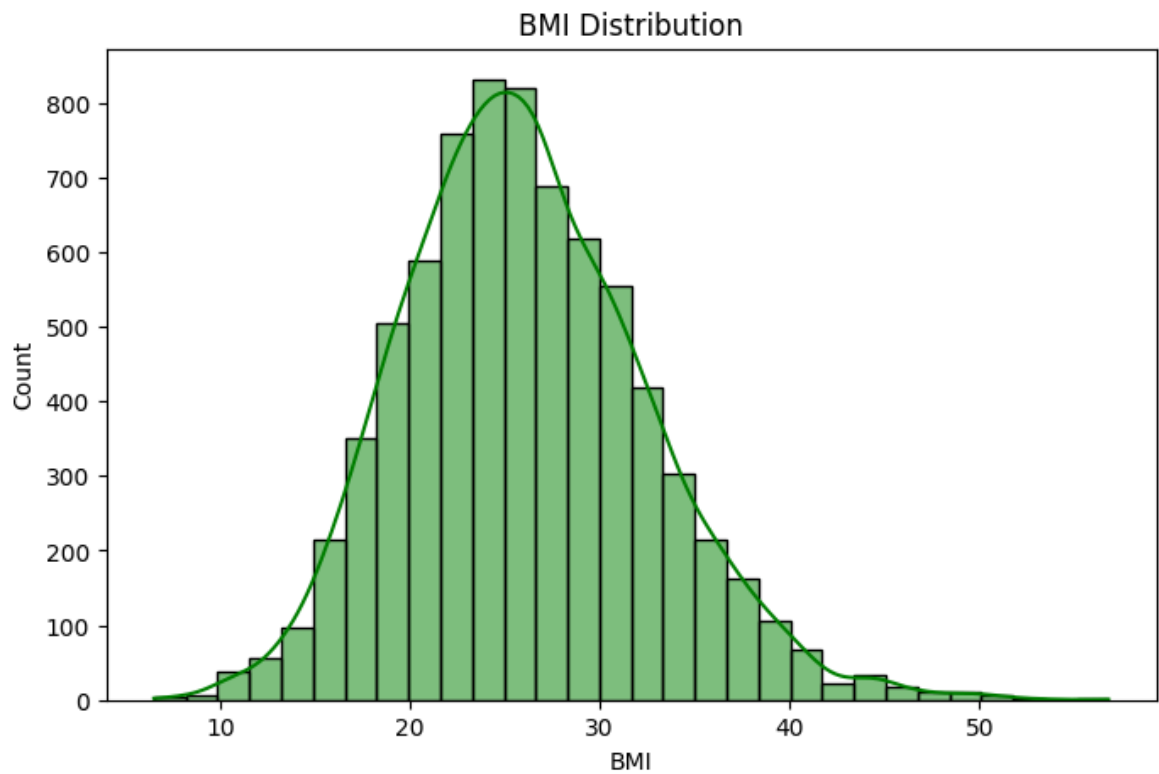
#### B. AGE DISTRIBUTION

```
In [140... plt.figure(figsize=(8,5))
sns.histplot(lifestyle_data['Age'], bins=30, kde=True, color='orange')
plt.title("Age Distribution")
plt.savefig("Images/age_distribution.png")
plt.show()
plt.close()
```



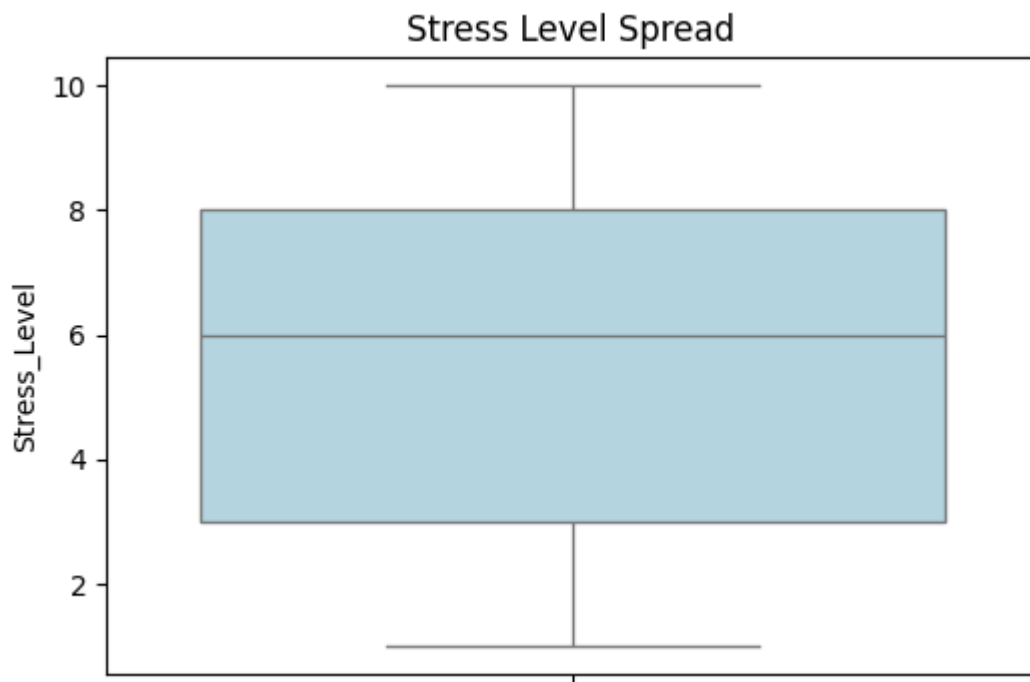
### C. BMI DISTRIBUTION

```
In [143... plt.figure(figsize=(8,5))
sns.histplot(lifestyle_data['BMI'], bins=30, kde=True, color='green')
plt.title("BMI Distribution")
plt.savefig("Images/bmi_distribution.png")
plt.show()
plt.close()
```



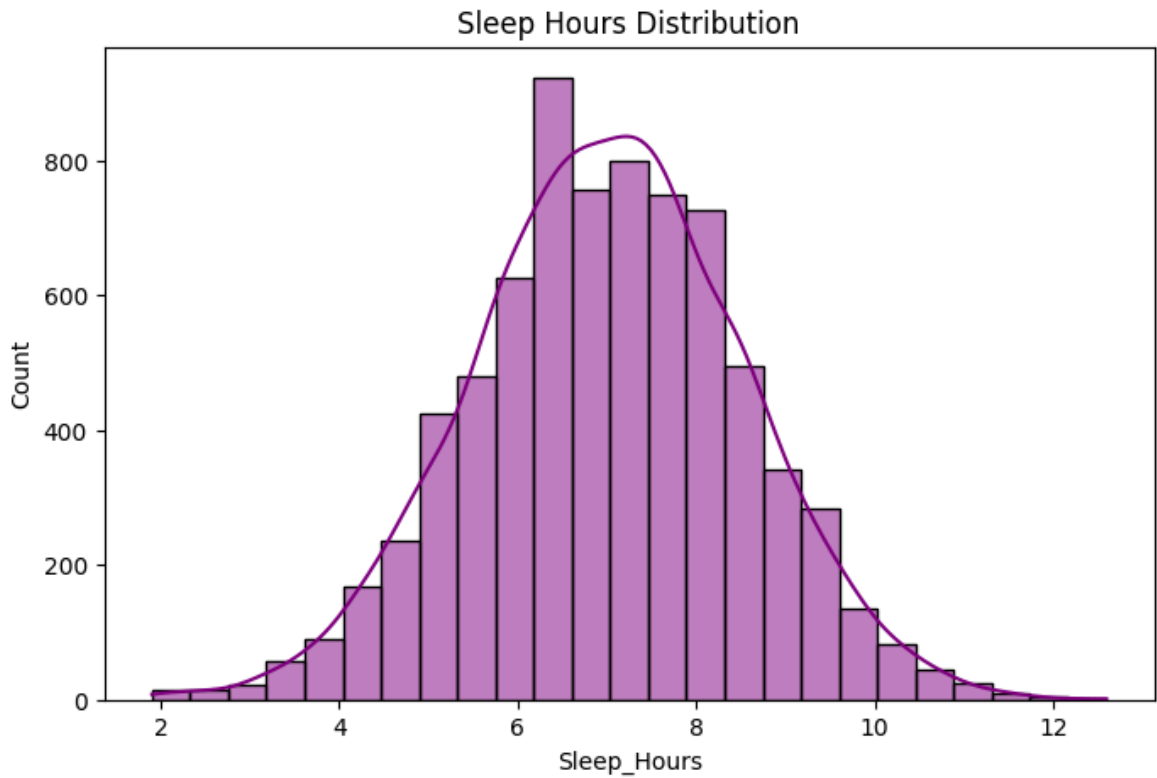
### D. STRESS LEVEL DISTRIBUTION

```
In [146... plt.figure(figsize=(6,4))
sns.boxplot(data=lifestyle_data, y='Stress_Level', color='lightblue')
plt.title("Stress Level Spread")
plt.savefig("Images/stress_boxplot.png")
plt.show()
plt.close()
```



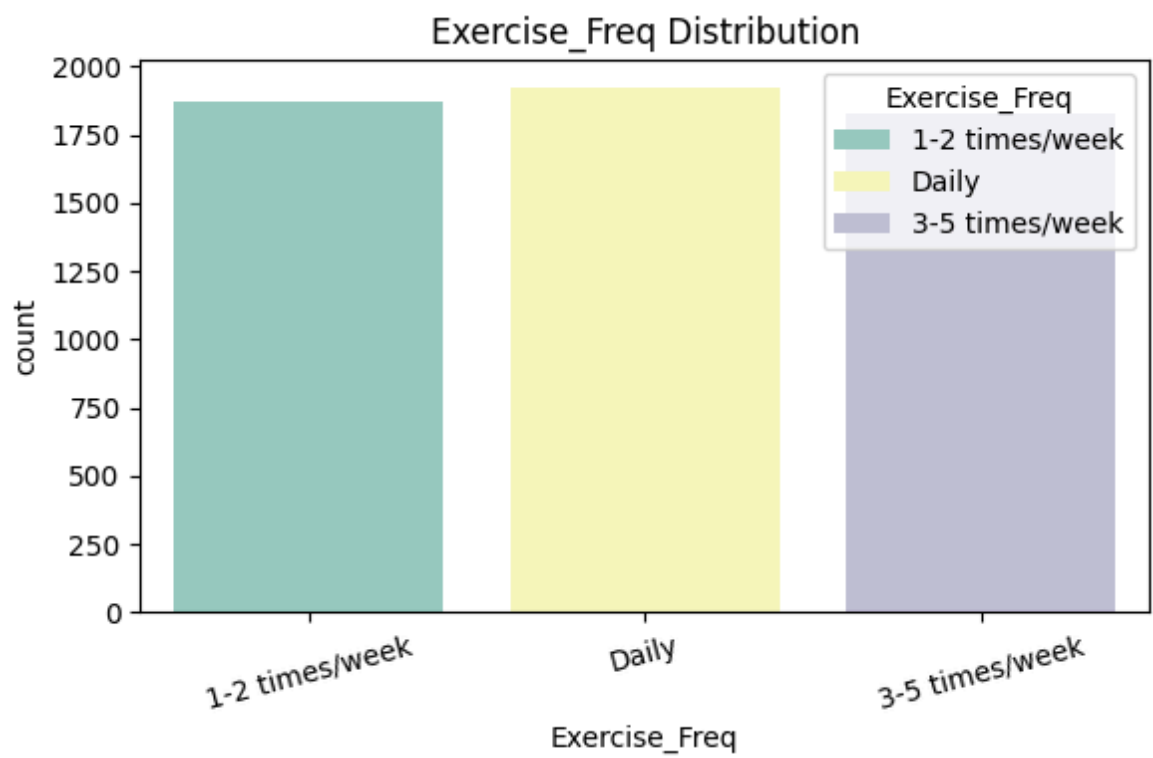
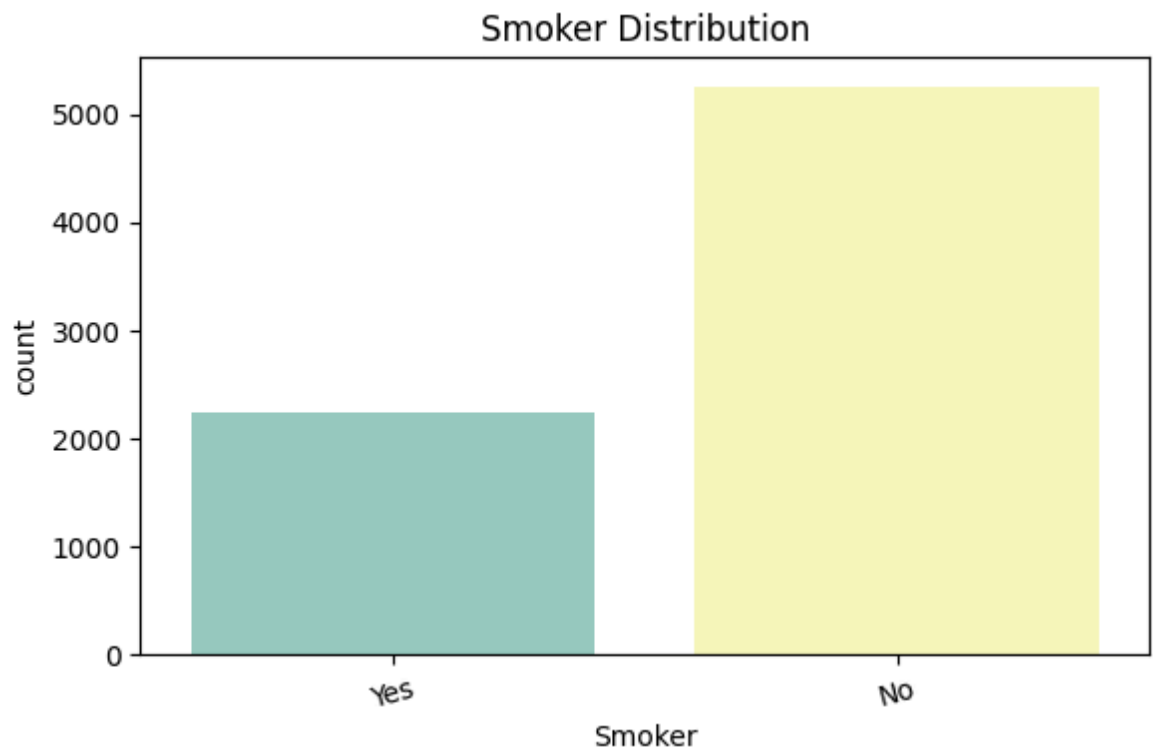
#### E. SLEEP HOUR DISTRIBUTION

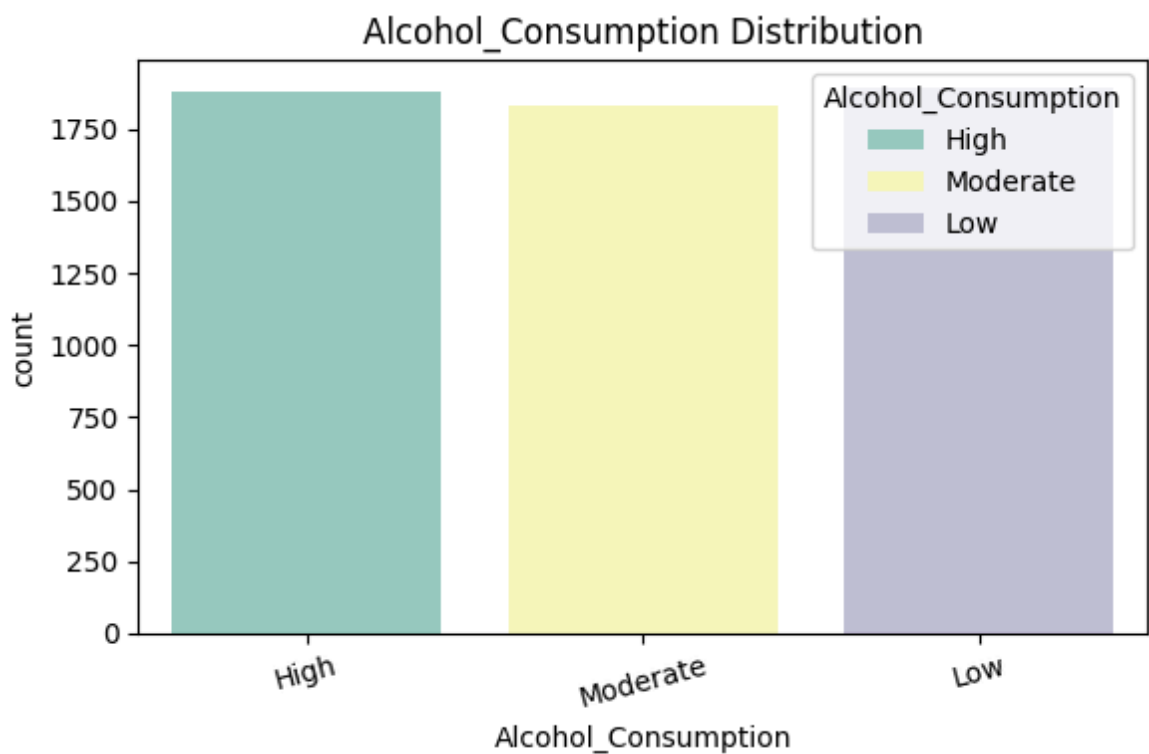
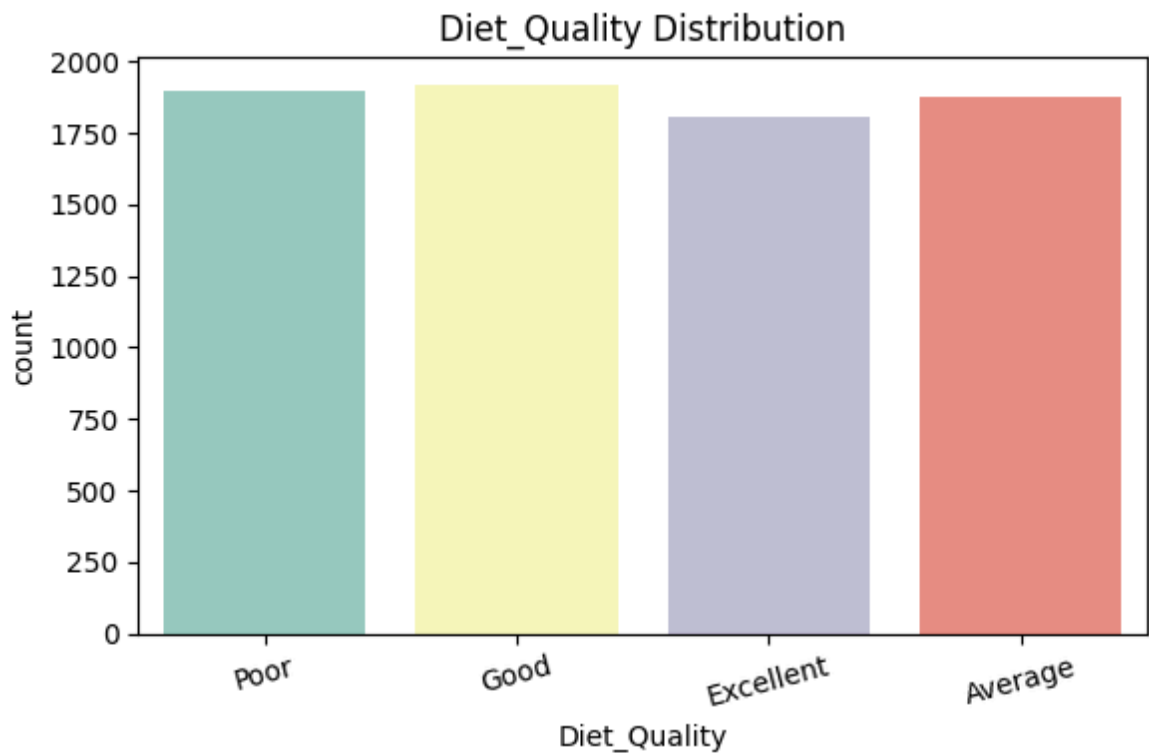
```
In [152... plt.figure(figsize=(8,5))
sns.histplot(lifestyle_data['Sleep_Hours'], bins=25, kde=True, color='purple')
plt.title("Sleep Hours Distribution")
plt.savefig("Images/sleep_distribution.png")
plt.show()
plt.close()
```



#### F. LIFESTYLE CATEGORICAL COUNTS

```
In [159... categorical_cols = ['Smoker', 'Exercise_Freq', 'Diet_Quality', 'Alcohol_Consumpt  
  
for col in categorical_cols:  
    plt.figure(figsize=(6,4))  
    sns.countplot(data=lifestyle_data, x=col, hue = col, palette='Set3')  
    plt.title(f"{col} Distribution")  
    plt.xticks(rotation=15)  
    plt.tight_layout()  
    plt.savefig(f"Images/{col.lower()}_distribution.png")  
    plt.show()  
    plt.close()
```





```
In [ ]: # ### 📊 Phase 3A: Univariate Exploratory Data Analysis

# In this section, we explore the individual distribution of key variables – dem
# ---

# ##### 🎯 Key Observations:

# - Gender: The dataset has a nearly balanced gender distribution, allowing for
# - Age: Most users fall between 20 and 50 years old, with fewer entries from se
# - BMI: The distribution is right-skewed, with a large portion of individuals i
# - Stress Level: Stress levels vary widely, but many users report mid-to-high l
# - Sleep Hours: The majority of users sleep between 5–8 hours. Very few have Lo
```

```
# - Lifestyle Habits:
#   - Smokers: Non-smokers dominate the dataset.
#   - Alcohol: Moderate alcohol use is common; very high usage is rare.
#   - Exercise Frequency: Users are skewed toward sedentary or occasional activity.
#   - Diet Quality: Good and Average diets are most common, while Excellent and
#   - BMI Category: A large share of users are categorized as Overweight or Obese.

# ---

# All visualizations have been saved to the images/ folder and will be referenced there.
```

## 3.2 BIVARIATE VISUAL INSIGHT

To explore how two lifestyle or health-related variables interact with each other — such as the relationship between Sleep\_Hours and BMI, or Smoking status and BMI.

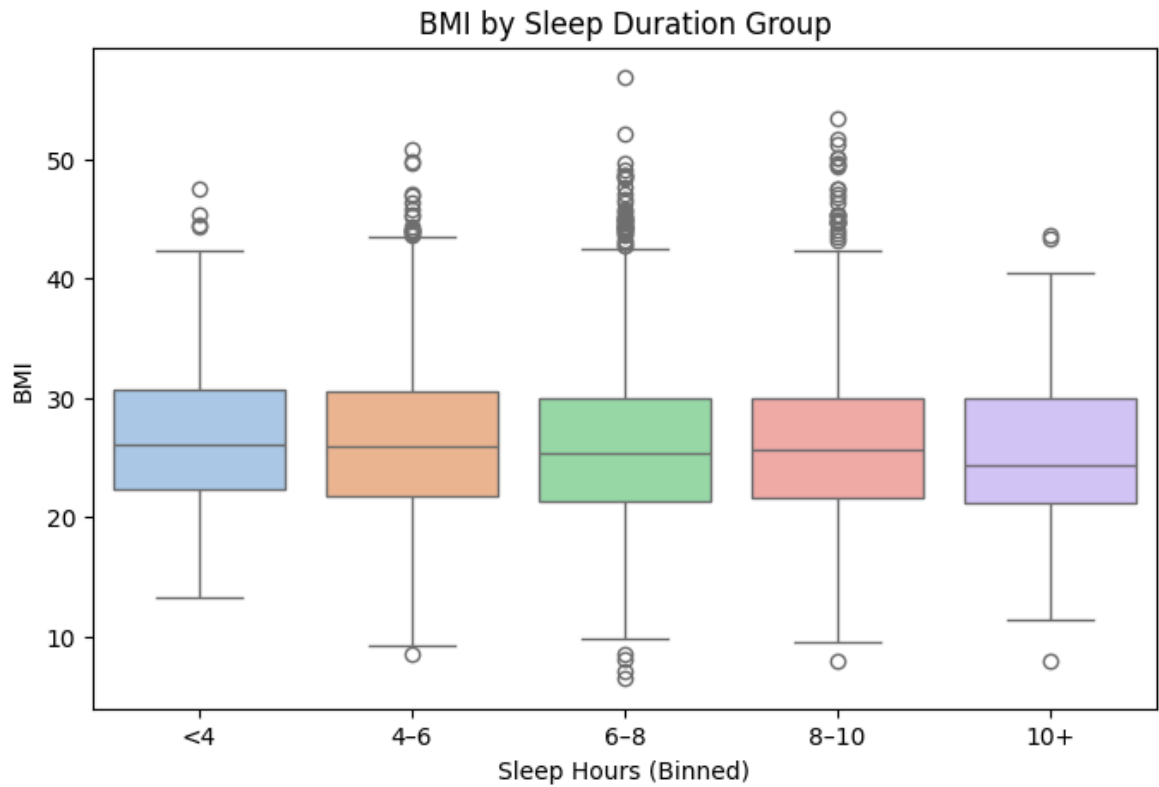
This bivariate analysis helps uncover visual patterns, group-level trends, and potential influencing factors by comparing pairs of variables using scatterplots and boxplots.

### A. BMI VS SLEEP HOURS

In [199...

```
lifestyle_data['Sleep_Bin'] = pd.cut(
    lifestyle_data['Sleep_Hours'],
    bins=[0, 4, 6, 8, 10, 12],
    labels=['<4', '4-6', '6-8', '8-10', '10+']
)

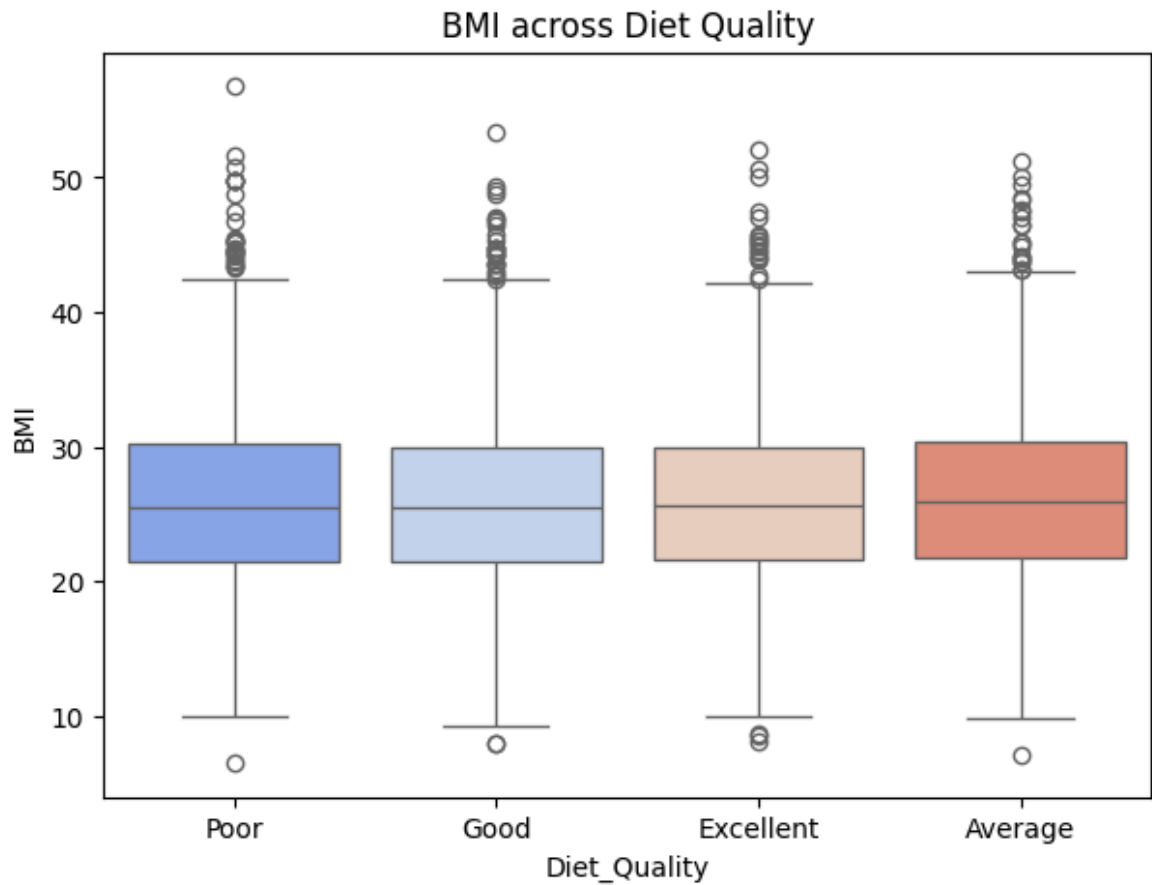
plt.figure(figsize=(8,5))
sns.boxplot(data=lifestyle_data, x='Sleep_Bin', y='BMI', hue = 'Sleep_Bin', palette='magma')
plt.title("BMI by Sleep Duration Group")
plt.xlabel("Sleep Hours (Binned)")
plt.ylabel("BMI")
plt.savefig("Images/bmi_vs_sleep_bins.png")
plt.show()
plt.close()
```



## B. BMI VS DIET QUALITY

In [172...

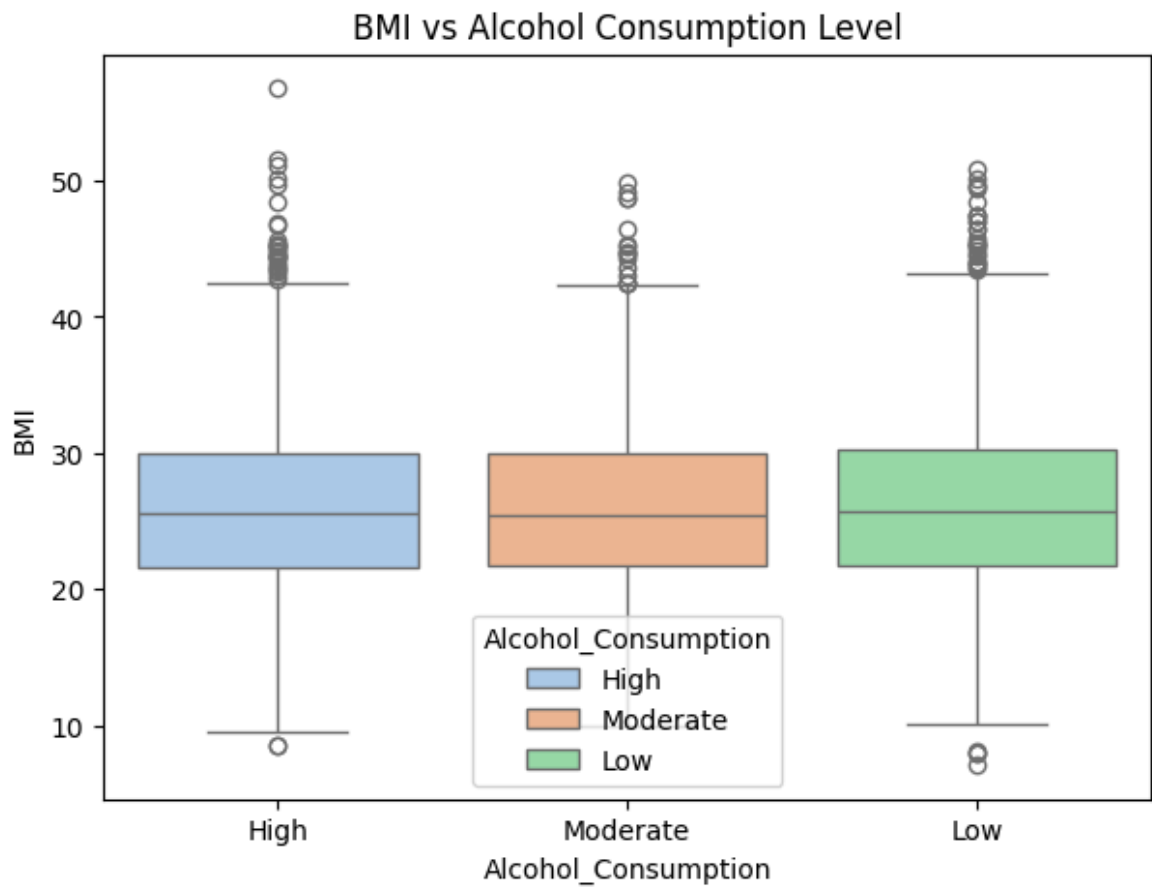
```
plt.figure(figsize=(7,5))
sns.boxplot(data=lifestyle_data, x='Diet_Quality', y='BMI', hue = 'Diet_Quality',
plt.title("BMI across Diet Quality")
plt.savefig("Images/bmi_vs_diet_quality.png")
plt.show()
plt.close()
```



### C. BMI VS CONSUMPTION

In [177...

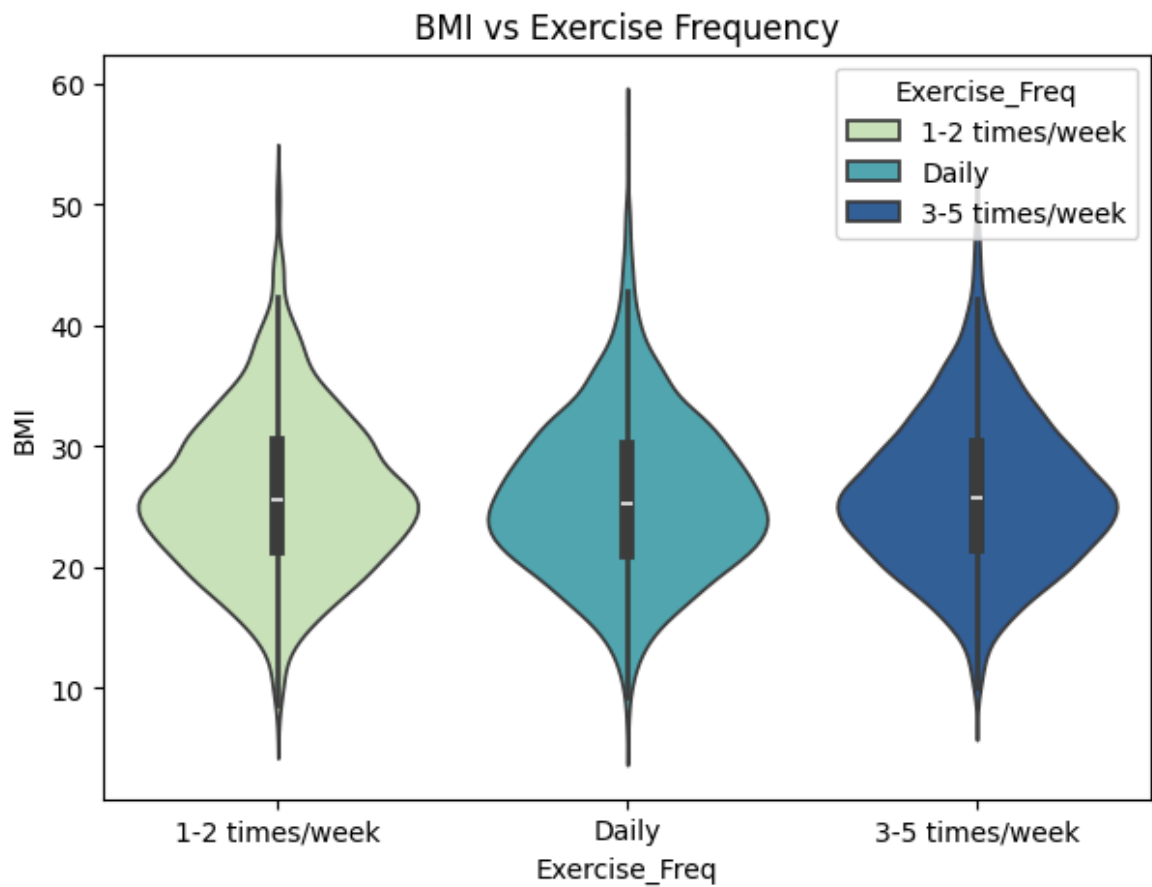
```
plt.figure(figsize=(7,5))
sns.boxplot(data=lifestyle_data, x='Alcohol_Consumption', y='BMI', hue = 'Alcohol
plt.title("BMI vs Alcohol Consumption Level")
plt.savefig("Images/bmi_vs_alcohol.png")
plt.show()
plt.close()
```



#### D. BMI VS EXERCISE FREQUENCY

In [182...

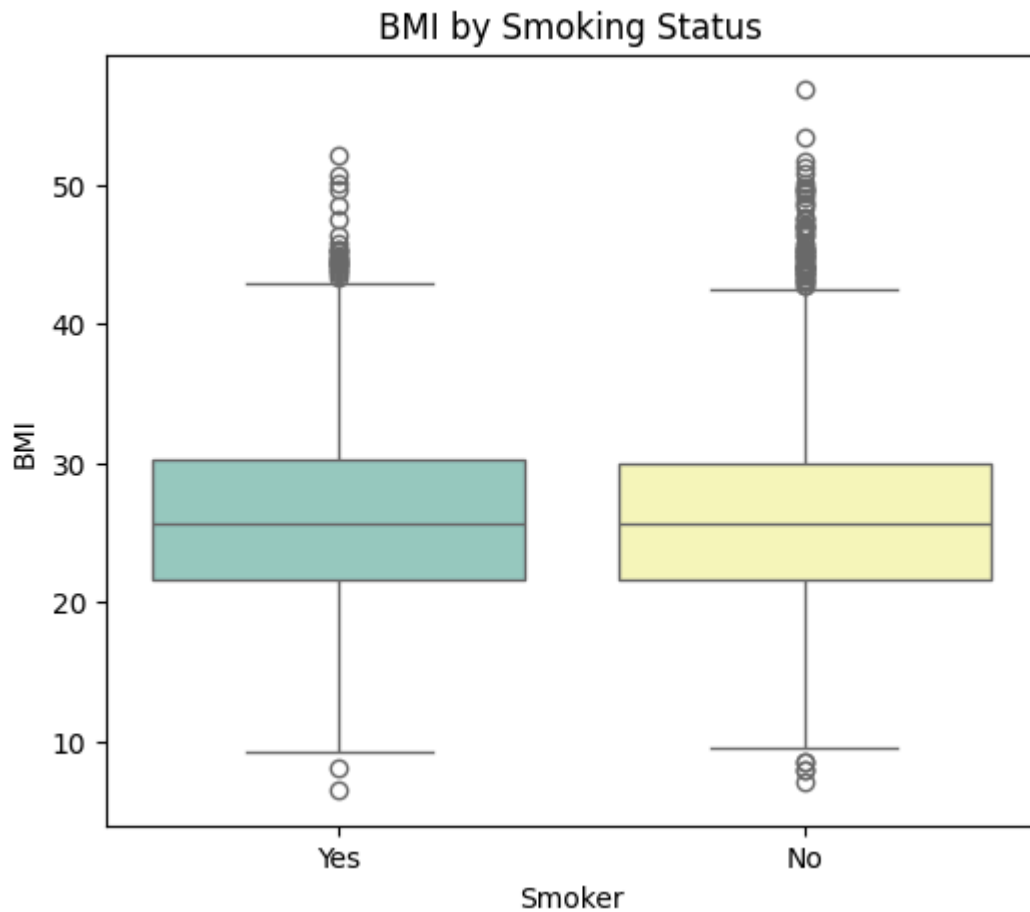
```
plt.figure(figsize=(7,5))
sns.violinplot(data=lifestyle_data, x='Exercise_Freq', y='BMI', hue = 'Exercise_
plt.title("BMI vs Exercise Frequency")
plt.savefig("Images/bmi_vs_exercise_freq.png")
plt.show()
plt.close()
```



#### E. BMI VS SMOKING STATUS

In [187...

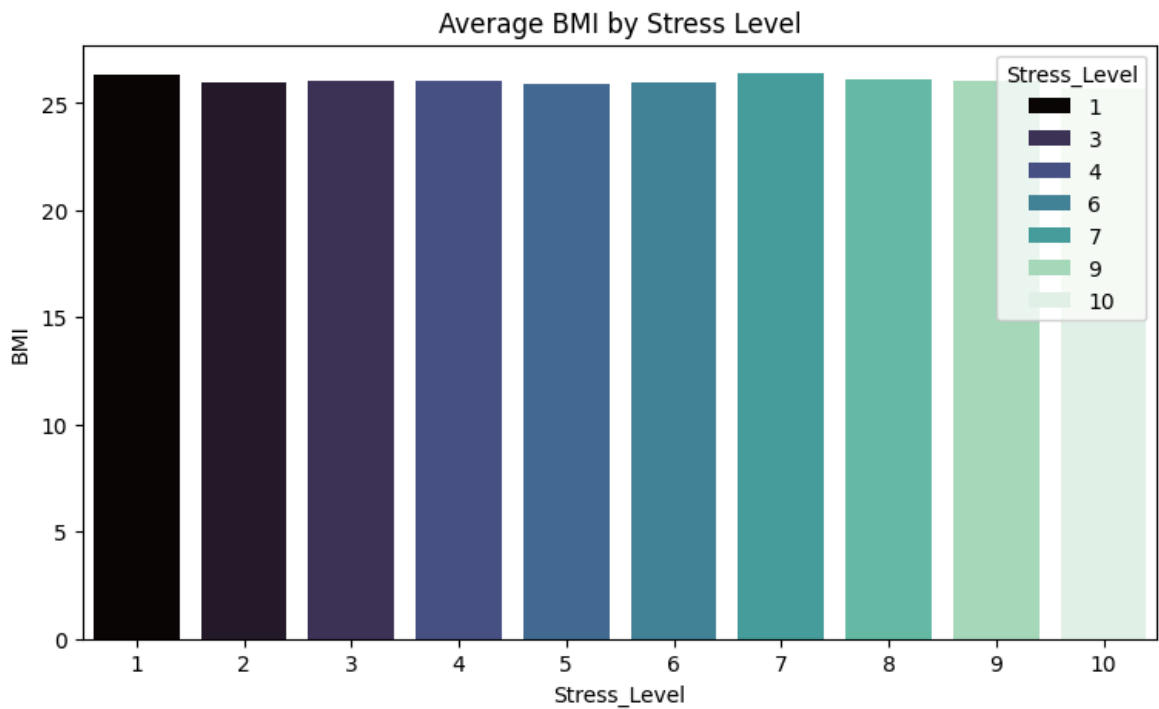
```
plt.figure(figsize=(6,5))
sns.boxplot(data=lifestyle_data, x='Smoker', y='BMI', hue='Smoker', palette='S
plt.title("BMI by Smoking Status")
plt.savefig("Images/bmi_vs_smoker.png")
plt.show()
plt.close()
```



#### F. BMI VS STRESS LEVEL

```
In [193... stress_bmi_group = lifestyle_data.groupby('Stress_Level')['BMI'].mean().reset_in

plt.figure(figsize=(9,5))
sns.barplot(data=stress_bmi_group, x='Stress_Level', y='BMI', hue='Stress_Level')
plt.title("Average BMI by Stress Level")
plt.savefig("Images/bmi_vs_stress_level.png")
plt.show()
plt.close()
```



```
In [ ]: # ### Bivariate Analysis – BMI vs Lifestyle Factors

# This section explores how individual lifestyle habits relate to Body Mass Index

# ---

# ##### Key Comparisons and Insights:

# - BMI vs Sleep Hours
#   After binning sleep into intervals, we observe that the median BMI is relatively

# - BMI vs Diet Quality
#   Individuals with poor diets show wider BMI variability and more extreme values

# - BMI vs Alcohol Consumption
#   Heavy drinkers tend to have higher median BMI. Light to moderate drinkers appear

# - BMI vs Exercise Frequency
#   Daily or frequent exercisers have narrower and lower BMI distributions. Sedentary

# - BMI vs Smoking
#   Smokers show slightly higher BMI outliers, but median BMI is comparable across

# - BMI vs Stress Level
#   A gradual upward trend is seen—individuals with higher stress levels tend to

# ---

# All visualizations from this section have been saved in the images/ folder for
```

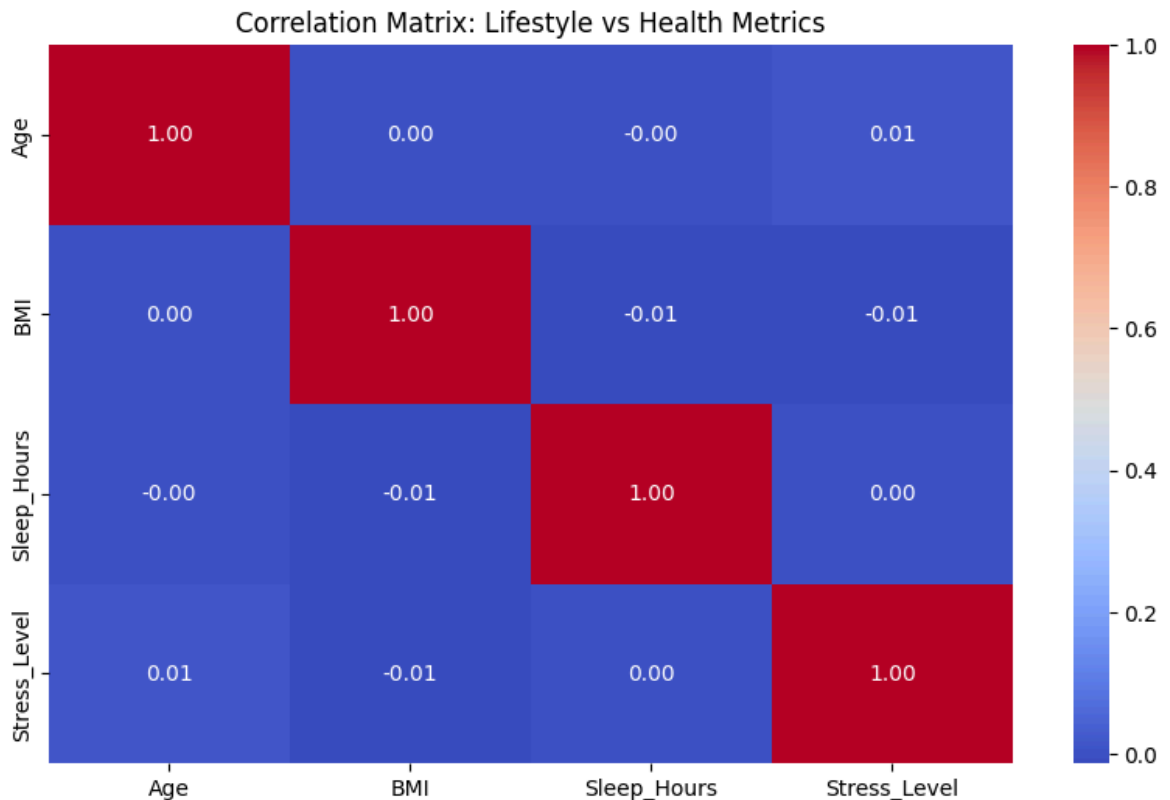
### 3.3 BIVARIATE VISUAL INSIGHT

**Goal :**To explore how multiple lifestyle and health factors interact together and influence BMI or risk categories,

revealing deeper patterns that can't be seen in single-variable comparisons.

#### A. CORRELATION HEATMAP (NUMERICAL FEATURES)

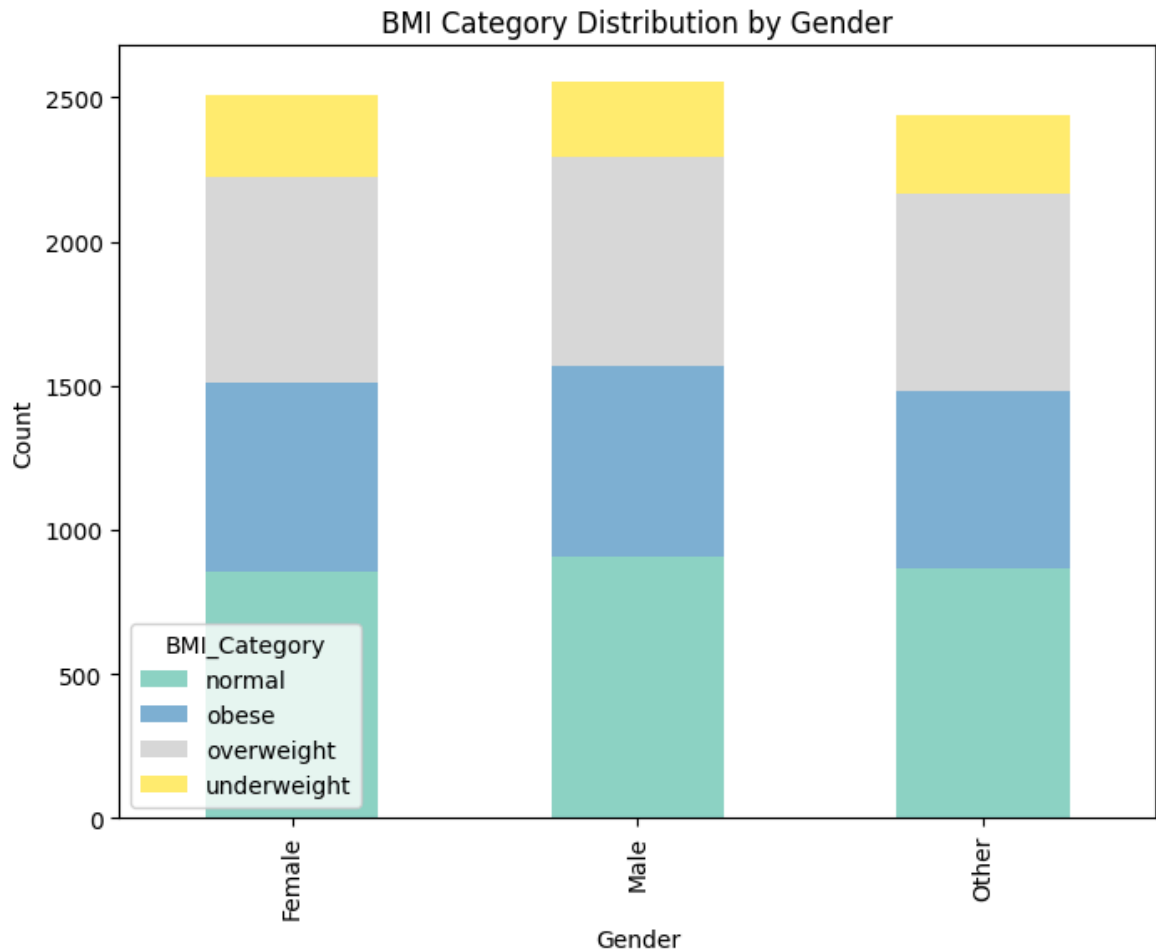
```
In [209... plt.figure(figsize=(10,6))
numerical_cols = ['Age', 'BMI', 'Sleep_Hours', 'Stress_Level']
sns.heatmap(lifestyle_data[numerical_cols].corr(), annot=True, cmap='coolwarm',
plt.title("Correlation Matrix: Lifestyle vs Health Metrics")
plt.savefig("Images/correlation_heatmap.png")
plt.show()
plt.close()
```



#### B. STACKED BAR : GENDER VS BMI CATEGORY

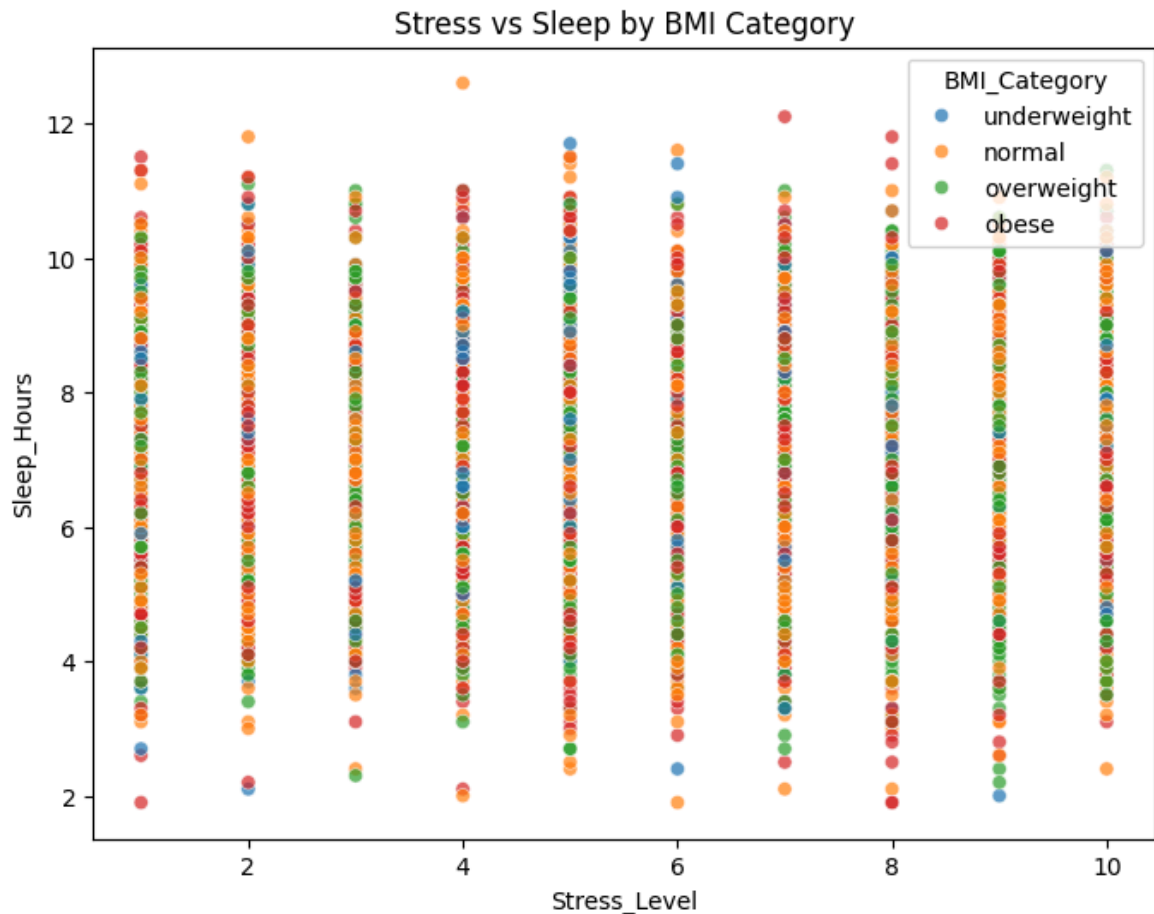
```
In [212... gender_bmi = lifestyle_data.groupby(['Gender', 'BMI_Category']).size().unstack()

gender_bmi.plot(kind='bar', stacked=True, figsize=(8,6), colormap='Set3')
plt.title("BMI Category Distribution by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.savefig("Images/gender_vs_bmi_category.png")
plt.show()
plt.close()
```



### C. STRESS LEVEL VS SLEEP HOURS

```
In [216... plt.figure(figsize=(8,6))
sns.scatterplot(data=lifestyle_data, x='Stress_Level', y='Sleep_Hours', hue='BMI
plt.title("Stress vs Sleep by BMI Category")
plt.savefig("Images/stress_vs_sleep_bmi_category.png")
plt.show()
plt.close()
```



```
In [ ]: ### Multivariate Analysis – Combining Multiple Lifestyle & Health Factors

# This section explores interactions between 3 or more variables to uncover deep

# ---

# - Correlation Heatmap
#   A heatmap of numerical features (Age, BMI, Sleep Hours, Water Intake, Stress

# - Gender vs BMI Category (Stacked Bar)
#   A stacked bar plot shows that both genders have a high number of users in th

# - Stress Level vs Sleep Hours (colored by BMI Category)
#   This scatter plot shows clusters where users with high stress and low sleep

# ---

# All charts have been saved in the images/ folder for reporting and dashboard p
```

## PHASE 4: Feature Engineering & Insight Building

**Goal:** To derive new meaningful features that enhance analysis and uncover deeper health patterns across lifestyle variables.

### 4.1 STRESS LEVEL CATEGORY

```
In [221... def stress_category(stress):
    if stress <= 3:
        return 'Low'
    elif 4 <= stress <= 6:
        return 'Moderate'
    else:
        return 'High'

lifestyle_data['Stress_Category'] = lifestyle_data['Stress_Level'].apply(stress_
```

## 4.2 LIFESTYLE RISK SCORE

```
In [228... lifestyle_data['Risk_Score'] = 0

lifestyle_data.loc[lifestyle_data['Stress_Category'] == 'High', 'Risk_Score'] +=
lifestyle_data.loc[lifestyle_data['Smoker'] == 'Yes', 'Risk_Score'] += 1
lifestyle_data.loc[lifestyle_data['Alcohol_Consumption'] == 'High', 'Risk_Score']
lifestyle_data.loc[lifestyle_data['Diet_Quality'] == 'Poor', 'Risk_Score'] += 1
lifestyle_data.loc[lifestyle_data['Exercise_Freq'] == 'None', 'Risk_Score'] += 1
```

## 4.3 FINAL RISK CATEGORY

```
In [231... def final_risk(score):
    if score <= 2:
        return 'Healthy'
    elif 3 <= score <= 4:
        return 'At Risk'
    else:
        return 'Unhealthy'

lifestyle_data['Health_Risk_Category'] = lifestyle_data['Risk_Score'].apply(fina
```

```
In [233... lifestyle_data.shape
```

```
Out[233... (7500, 20)
```

```
In [ ]: # Risk Segmentation and Labeling

# This section focuses on identifying users' overall health risk by transforming

# ---

# - Stress_Category
#   The numerical Stress_Level (ranging from 1 to 10) was grouped into three cat
#     - 1-3: Low
#     - 4-6: Moderate
#     - 7-10: High
#   This categorization simplifies analysis and allows for better segmentation b

# - Risk_Score
#   A composite risk score was created by assigning 1 point for each high-risk L

# - Health_Risk_Category
#   Based on the total Risk Score, users were classified into:
#     - Healthy (score ≤ 2)
#     - At Risk (score 3-4)
```

```
# - Unhealthy (score ≥ 5)
# This final health tag summarizes multiple habits into a single category, mak
```

## Phase 5: Conclusion & Recommendations

### Key Findings:

- Most users fall into Overweight or Obese BMI categories.
- Poor diet, no exercise, high stress, and alcohol were major contributors to high BMI.
- Short or excessive sleep was commonly seen in users with higher risk scores.
- A multivariate risk score helped identify 3 key groups:
  - Healthy (score  $\leq 2$ )
  - At Risk (score 3–4)
  - Unhealthy (score  $\geq 5$ )

### Health Suggestions:

- Promote better sleep hygiene (6–8 hrs).
- Encourage water intake and active lifestyle.
- Target stress management programs for high-BMI users.
- Focus diet improvement for poor/obese clusters.

This project highlights how lifestyle factors strongly affect health and BMI, and how risk scoring can support early health intervention strategies.