In [14]:

```
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
data.head()
```

Out[14]:

	Student_ID	Age	Gender	University_Year	Sleep_Duration	Study_Hours	Screen_Time
0	1	24	Other	2nd Year	7.7	7.9	3.4
1	2	21	Male	1st Year	6.3	6.0	1.9
2	3	22	Male	4th Year	5.1	6.7	3.9
3	4	24	Other	4th Year	6.3	8.6	2.8
4	5	20	Male	4th Year	4.7	2.7	2.7

In [24]:

```
data.shape
```

Out[24]:

```
(500, 14)
```

In [15]:

```
data.tail()
```

Out[15]:

	Student_ID	Age	Gender	University_Year	Sleep_Duration	Study_Hours	Screen_Ti
495	496	24	Male	2nd Year	5.1	9.3	
496	497	20	Male	2nd Year	8.9	7.7	
497	498	21	Male	3rd Year	5.7	6.4	
498	499	18	Female	2nd Year	4.9	0.5	
499	500	21	Male	3rd Year	7.9	11.6	

In [19]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Student_ID           500 non-null    int64  
 1   Age                  500 non-null    int64  
 2   Gender               500 non-null    object 
 3   Year                 500 non-null    object 
 4   SleepHours           500 non-null    float64
 5   StudyHours           500 non-null    float64
 6   ScreenHours          500 non-null    float64
 7   CaffeineUnits        500 non-null    int64  
 8   Physical_Activity    500 non-null    int64  
 9   Sleep_Quality        500 non-null    int64  
 10  Weekday_Sleep_Start  500 non-null    float64
 11  Weekend_Sleep_Start  500 non-null    float64
 12  Weekday_Sleep_End    500 non-null    float64
 13  Weekend_Sleep_End    500 non-null    float64
dtypes: float64(7), int64(5), object(2)
memory usage: 54.8+ KB
```

In [20]:

```
data.describe()
```

Out[20]:

	Student_ID	Age	SleepHours	StudyHours	ScreenHours	CaffeineUnits	Ph
count	500.000000	500.00000	500.000000	500.000000	500.000000	500.000000	
mean	250.500000	21.53600	6.472400	5.981600	2.525000	2.462000	
std	144.481833	2.33315	1.485764	3.475725	0.859414	1.682325	
min	1.000000	18.00000	4.000000	0.100000	1.000000	0.000000	
25%	125.750000	20.00000	5.100000	2.900000	1.800000	1.000000	
50%	250.500000	21.00000	6.500000	6.050000	2.600000	2.000000	
75%	375.250000	24.00000	7.800000	8.800000	3.300000	4.000000	
max	500.000000	25.00000	9.000000	12.000000	4.000000	5.000000	

In [23]:

```
data.columns
```

Out[23]:

```
Index(['Student_ID', 'Age', 'Gender', 'Year', 'SleepHours', 'StudyHours',
       'ScreenHours', 'CaffeineUnits', 'Physical_Activity', 'Sleep_Quality',
       'Weekday_Sleep_Start', 'Weekend_Sleep_Start', 'Weekday_Sleep_End',
       'Weekend_Sleep_End'],
      dtype='object')
```

In [6]:

```
#DATA MUNGING
# Renaming columns
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
data.rename(columns={
    'University_Year': 'Year',
    'Sleep_Duration': 'SleepHours',
    'Study_Hours': 'StudyHours',
    'Screen_Time': 'ScreenHours',
    'Caffeine_Intake': 'CaffeineUnits'
}, inplace=True)

# Parsing dates if applicable (assuming a 'Date' column exists)
if 'Date' in data.columns:
    data['Date'] = pd.to_datetime(data['Date'])

# Converting numerical columns to float for calculations to prevent any errors
data['SleepHours'] = data['SleepHours'].astype(float)
data['StudyHours'] = data['StudyHours'].astype(float)

print("Preview of munged data:")
data.head(7)
```

```
Preview of munged data:
```

Out[6]:

	Student_ID	Age	Gender	Year	SleepHours	StudyHours	ScreenHours	CaffeineUnits
0	1	24	Other	2nd Year	7.7	7.9	3.4	2
1	2	21	Male	1st Year	6.3	6.0	1.9	5
2	3	22	Male	4th Year	5.1	6.7	3.9	5
3	4	24	Other	4th Year	6.3	8.6	2.8	4
4	5	20	Male	4th Year	4.7	2.7	2.7	0
5	6	25	Other	1st Year	4.9	12.0	3.2	3
6	7	22	Female	2nd Year	6.5	11.7	3.4	1

In [8]:

```
#DATA CLEANING
# Identifying missing values in each column
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
missing_counts = data.isnull().sum()
print("Missing values per column:")
print(missing_counts)
data.fillna(10)

# Filling missing sleep hours with the median
if 'SleepHours' in data.columns:
    median_sleep = data['SleepHours'].median()
    data['SleepHours'].fillna(median_sleep, inplace=True)
    print(f"Filled missing SleepHours with median: {median_sleep}")

# Removes rows with missing values
data.dropna()

#Check for duplicate rows
data.duplicated()

#removes duplicate rows
```

```
data.drop_duplicates()

# Standardizing categorical data (e.g., Gender to 'Male', 'Female', 'Other')
if 'Gender' in data.columns:
    data['Gender'] = data['Gender'].str.strip().str.capitalize()

# Cleaned dataset preview
print("Preview of cleaned data:")
data.head()
```

```
Missing values per column:
Student_ID             0
Age                    0
Gender                 0
University_Year        0
Sleep_Duration         0
Study_Hours            0
Screen_Time            0
Caffeine_Intake        0
Physical_Activity      0
Sleep_Quality          0
Weekday_Sleep_Start    0
Weekend_Sleep_Start    0
Weekday_Sleep_End      0
Weekend_Sleep_End      0
dtype: int64
Preview of cleaned data:
```

Out[8]:

	Student_ID	Age	Gender	University_Year	Sleep_Duration	Study_Hours	Screen_Time
0	1	24	Other	2nd Year	7.7	7.9	3.4
1	2	21	Male	1st Year	6.3	6.0	1.9
2	3	22	Male	4th Year	5.1	6.7	3.9
3	4	24	Other	4th Year	6.3	8.6	2.8
4	5	20	Male	4th Year	4.7	2.7	2.7

In [7]:

```
#DATA FILTERING
# Filtering students who sleep less than 6 hours
less_sleep = data[data['SleepHours'] < 6]
print(f"Students sleeping less than 6 hours: {less_sleep.shape[0]} records.")

# Filtering second-year students with high study hours (>5 hours)
high_study_second_year = data[(data['Year'] == '2') & (data['StudyHours'] > 5)]
print(f"Second-year students studying more than 5 hours: {high_study_second_year

# Combining multiple filters
poor_sleep_and_high_study = data[(data['SleepHours'] < 6) & (data['StudyHours']
print("Subset: Students with poor sleep and high study hours")
poor_sleep_and_high_study.head()
```

```
Students sleeping less than 6 hours: 198 records.
Second-year students studying more than 5 hours: 0 records.
Subset: Students with poor sleep and high study hours
```

Out[7]:

	Student_ID	Age	Gender	Year	SleepHours	StudyHours	ScreenHours	CaffeineUni
2	3	22	Male	4th Year	5.1	6.7	3.9	
5	6	25	Other	1st Year	4.9	12.0	3.2	
9	10	19	Other	2nd Year	5.8	8.2	2.0	
14	15	25	Female	4th Year	4.9	10.4	2.3	
16	17	21	Female	3rd Year	4.7	8.9	3.8	

In [11]:

```
#DATA AGGREGATION
# Aggregating data by 'Year' to calculate averages
aggregated_data = data.groupby('Year').agg({
    'SleepHours': 'mean',
    'StudyHours': 'mean',
    'ScreenHours': 'mean'
}).reset_index()

# Renaming columns for comfortability
aggregated_data.rename(columns={
    'SleepHours': 'AvgSleepHours',
    'StudyHours': 'AvgStudyHours',
    'ScreenHours': 'AvgScreenHours'
}, inplace=True)
print("Aggregated data by Year:")
print(aggregated_data)

# Aggregating by Gender and Year to compare metrics
gender_year_agg = data.groupby(['Gender', 'Year']).agg({
    'SleepHours': 'mean',
    'StudyHours': 'mean'
}).reset_index()
print("Aggregated data by Gender and Year:")
print(gender_year_agg)
```

```
Aggregated data by Year:
       Year  AvgSleepHours  AvgStudyHours  AvgScreenHours
0  1st Year       6.493600       5.804000        2.448800
1  2nd Year       6.561832       6.081679        2.600000
2  3rd Year       6.489394       6.429545        2.450000
3  4th Year       6.324107       5.534821        2.610714
Aggregated data by Gender and Year:
    Gender      Year  SleepHours  StudyHours
0   Female  1st Year    6.702326    4.804651
1   Female  2nd Year    6.597368    5.892105
2   Female  3rd Year    6.551111    5.895556
3   Female  4th Year    6.265000    5.550000
4     Male  1st Year    6.238636    5.897727
5     Male  2nd Year    6.274074    6.087037
6     Male  3rd Year    6.568750    6.604167
7     Male  4th Year    6.350000    5.427500
8    Other  1st Year    6.552632    6.826316
9    Other  2nd Year    6.925641    6.258974
10   Other  3rd Year    6.320513    6.830769
11   Other  4th Year    6.365625    5.650000
```

In [12]:

```
#DATA MERGING USING SAMPLE DATA
# Sample data for merging
student_data = pd.DataFrame({
    'StudentID': [1, 2, 3, 4],
    'SleepHours': [6.5, 5.2, 7.0, 4.8],
    'StudyHours': [2.0, 5.0, 3.5, 4.0]
})

course_data = pd.DataFrame({
    'StudentID': [1, 2, 3, 5],
    'Course': ['Math', 'Physics', 'Chemistry', 'Biology']
})

# Merging datasets on 'StudentID'
merged_data = pd.merge(student_data, course_data, on='StudentID', how='inner')

# Display merged data
print("Merged Data:")
print(merged_data)
```

```
Merged Data:
   StudentID  SleepHours  StudyHours     Course
0          1         6.5         2.0       Math
1          2         5.2         5.0    Physics
2          3         7.0         3.5  Chemistry
```

In [11]:

```
#DATA RESHAPING
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
# Pivot: Summarize Sleep_Duration per Weekday_Sleep_Start
pivot_sleep = data.pivot(index='Weekday_Sleep_Start', columns='Student_ID', valu

# Pivot: Summarize Study_Hours per Student_ID
pivot_study = data.pivot(index='Student_ID', columns='Gender', values='Study_Hou

# Melt: Convert to long format to compare multiple metrics
melted_data = data.melt(
    id_vars=['Student_ID', 'Gender', 'University_Year'],  # Fixed columns
    value_vars=['Sleep_Duration', 'Study_Hours', 'Screen_Time'],  # Metrics to u
    var_name='Metric',
    value_name='Hours'
)
print("Pivoted Data - Sleep Duration per Weekday Start Time:")
print(pivot_sleep)

print("\nPivoted Data - Study Hours per Gender:")
print(pivot_study)

print("\nMelted Data - Long Format:")
print(melted_data)
```

```
Pivoted Data - Sleep Duration per Weekday Start Time:
Student_ID           1    2    3    4    5    6    7    8    9    10   ...  \
Weekday_Sleep_Start                                                    ...   
1.08                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
1.17                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
1.22                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
1.24                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
1.28                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
...                  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...   
21.82                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
21.86                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
21.88                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
21.89                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   
21.93                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   

Student_ID           491  492  493  494  495  496  497  498  499  500  
Weekday_Sleep_Start                                                    
1.08                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
1.17                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
1.22                 NaN  NaN  NaN  NaN  NaN  NaN  8.9  NaN  NaN  NaN  
1.24                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
1.28                 NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
...                  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  
21.82                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
21.86                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
21.88                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
21.89                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  
21.93                NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  

[452 rows x 500 columns]

Pivoted Data - Study Hours per Gender:
Gender      Female  Male  Other
Student_ID                     
1              NaN   NaN    7.9
2              NaN   6.0    NaN
3              NaN   6.7    NaN
4              NaN   NaN    8.6
5              NaN   2.7    NaN
...            ...   ...    ...
496            NaN   9.3    NaN
497            NaN   7.7    NaN
498            NaN   6.4    NaN
499            0.5   NaN    NaN
500            NaN  11.6    NaN

[500 rows x 3 columns]

Melted Data - Long Format:
      Student_ID  Gender University_Year          Metric  Hours
0              1   Other        2nd Year  Sleep_Duration    7.7
1              2    Male        1st Year  Sleep_Duration    6.3
2              3    Male        4th Year  Sleep_Duration    5.1
3              4   Other        4th Year  Sleep_Duration    6.3
4              5    Male        4th Year  Sleep_Duration    4.7
...          ...     ...             ...             ...    ...
1495         496    Male        2nd Year     Screen_Time    1.9
1496         497    Male        2nd Year     Screen_Time    3.5
1497         498    Male        3rd Year     Screen_Time    3.9
1498         499  Female        2nd Year     Screen_Time    3.5
```

```
1499         500    Male        3rd Year     Screen_Time    1.0

[1500 rows x 5 columns]
```

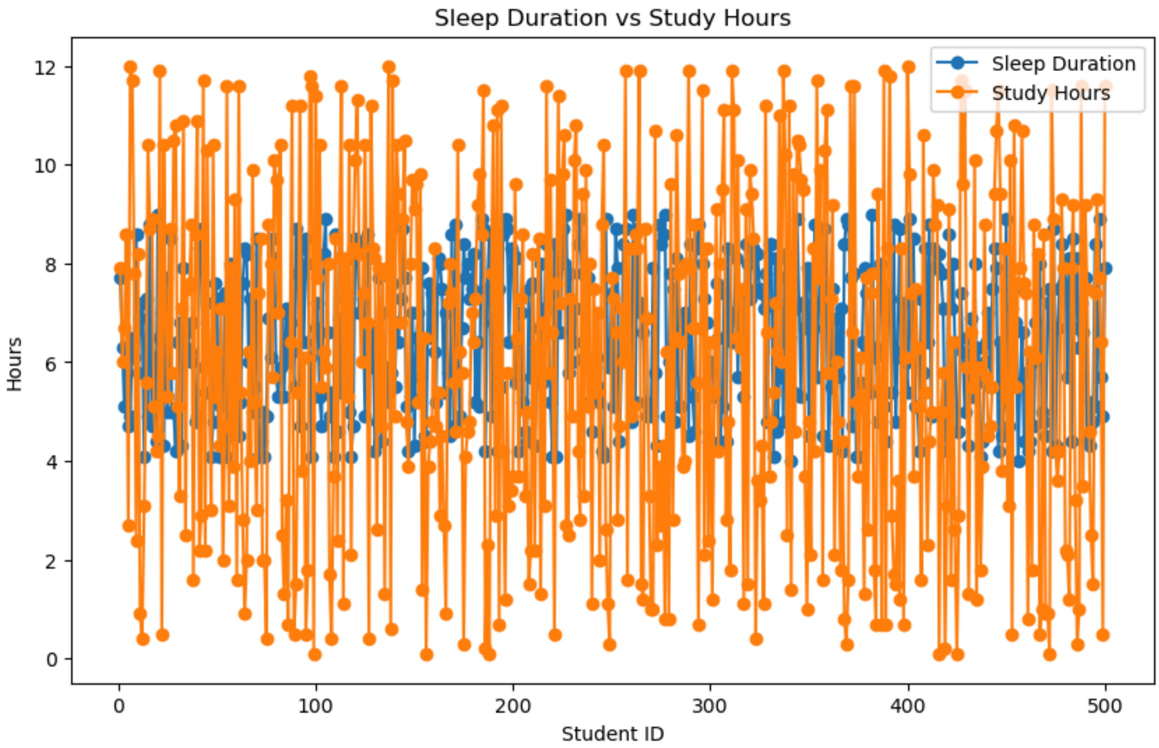
In [15]:

```python
#DATA GROUPING
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
print("Column names in the dataset:")
print(data.columns)
data.columns = data.columns.str.strip()

# Check if expected columns exist
if 'Sleep_Duration' not in data.columns or 'Study_Hours' not in data.columns:
    print("Error: 'Sleep_Duration' or 'Study_Hours' column is missing.")
    print("Available columns:", data.columns)
else:
    # Convert columns to numeric and handle errors
    data['Sleep_Duration'] = pd.to_numeric(data['Sleep_Duration'], errors='coerc
    data['Study_Hours'] = pd.to_numeric(data['Study_Hours'], errors='coerce')
    data.dropna(subset=['Sleep_Duration', 'Study_Hours'], inplace=True)

    # Grouping by University_Year and calculating averages
    grouped_data = data.groupby('University_Year').agg({
        'Sleep_Duration': 'mean',
        'Study_Hours': 'mean'
    }).reset_index()

    grouped_data.rename(columns={
        'Sleep_Duration': 'AvgSleepHours',
        'Study_Hours': 'AvgStudyHours'
    }, inplace=True)

    print("Grouped Data by University_Year:")
    print(grouped_data)
```

```
Column names in the dataset:
Index(['Student_ID', 'Age', 'Gender', 'University_Year', 'Sleep_Duration',
       'Study_Hours', 'Screen_Time', 'Caffeine_Intake', 'Physical_Activity',
       'Sleep_Quality', 'Weekday_Sleep_Start', 'Weekend_Sleep_Start',
       'Weekday_Sleep_End', 'Weekend_Sleep_End'],
      dtype='object')
Grouped Data by University_Year:
  University_Year  AvgSleepHours  AvgStudyHours
0        1st Year       6.493600       5.804000
1        2nd Year       6.561832       6.081679
2        3rd Year       6.489394       6.429545
3        4th Year       6.324107       5.534821
```

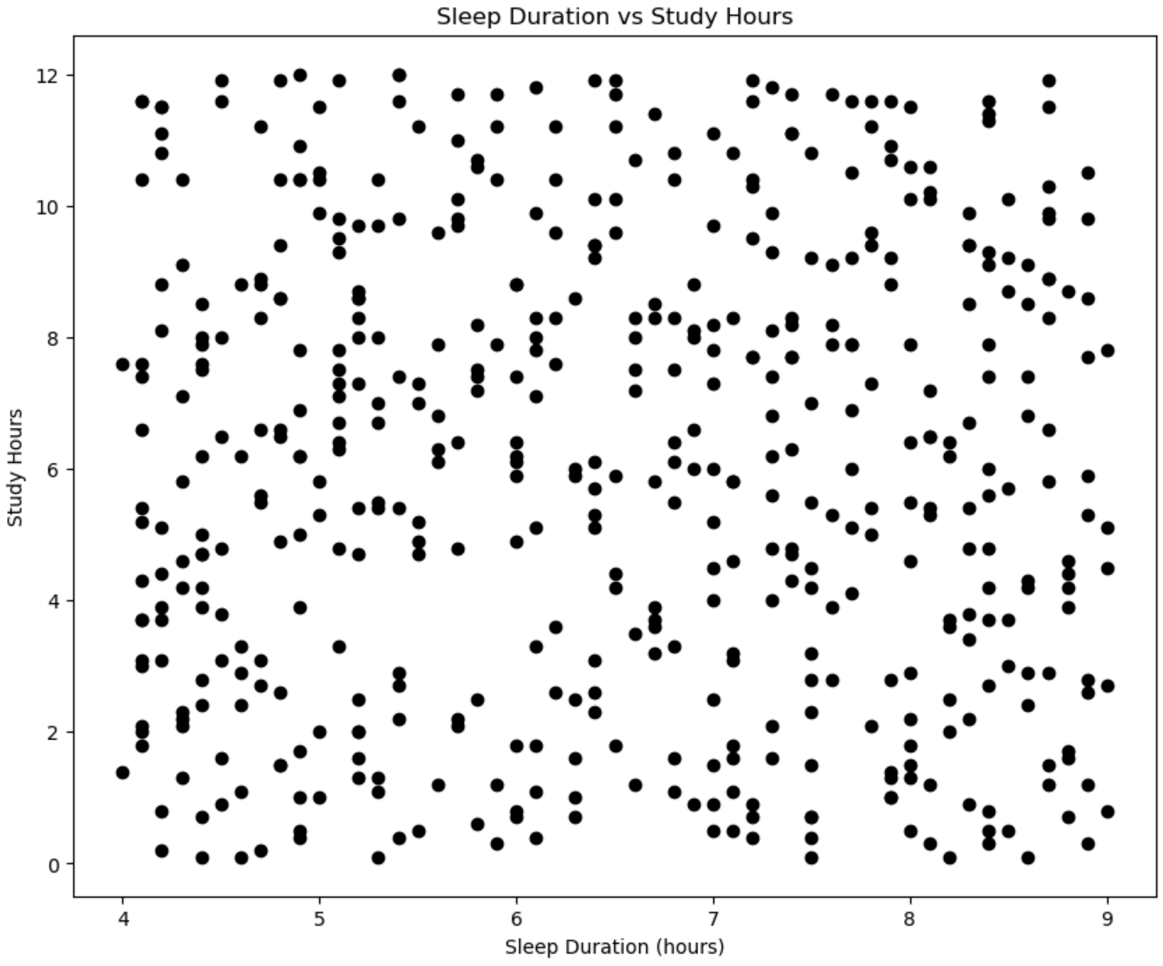
In [19]:

```python
import pandas as pd
data = pd.read_csv('47_student_sleep_patterns.csv')
data.head()

import matplotlib.pyplot as plt

# Line plot for Sleep Duration and Study Hours
plt.figure(figsize=(10, 6))
plt.plot(data['Student_ID'], data['Sleep_Duration'], label='Sleep Duration', mar
plt.plot(data['Student_ID'], data['Study_Hours'], label='Study Hours', marker='c
plt.title('Sleep Duration vs Study Hours')
plt.xlabel('Student ID')
plt.ylabel('Hours')
plt.legend()
plt.show()
```



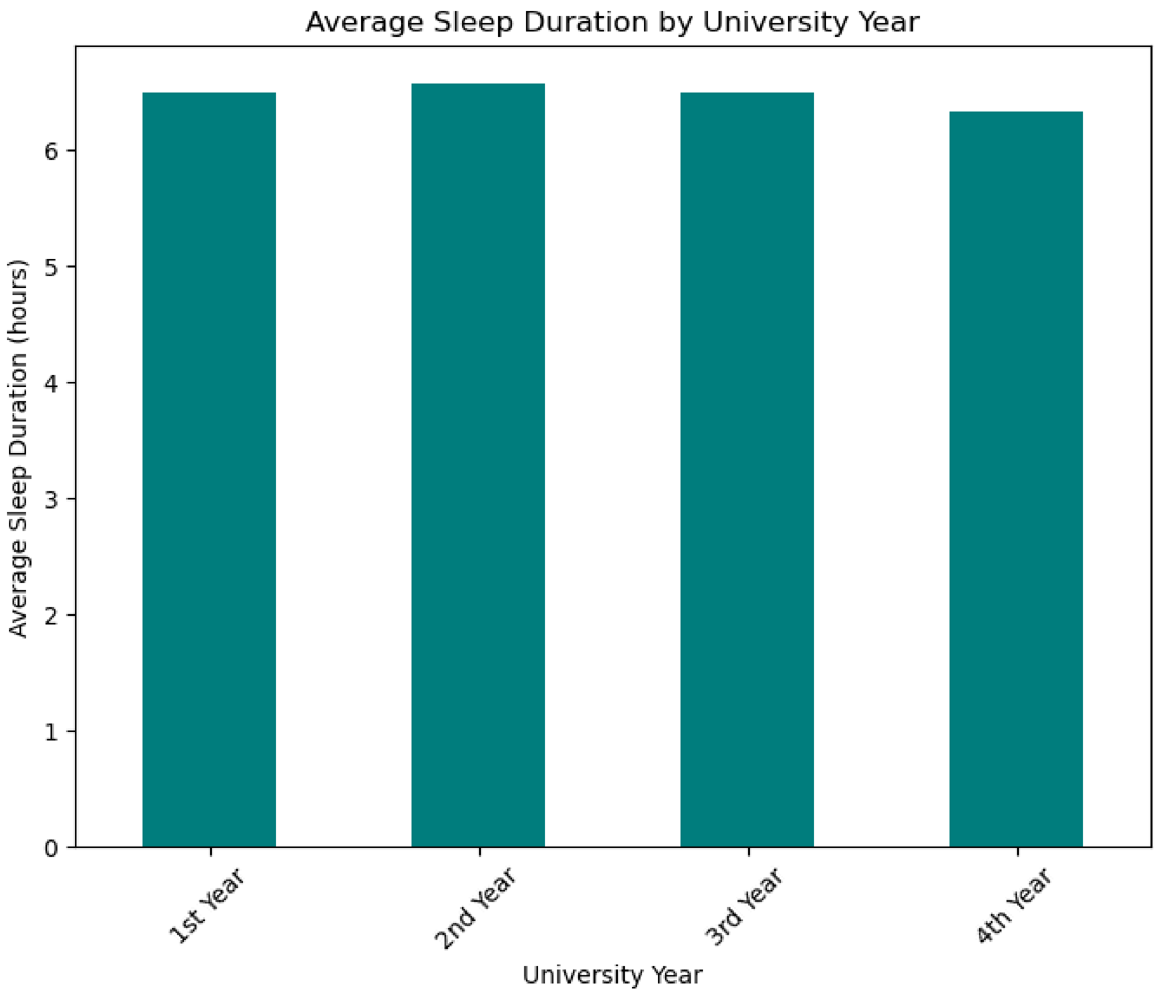
In [18]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 8))
plt.scatter(data['Sleep_Duration'], data['Study_Hours'], c='black')
plt.title('Sleep Duration vs Study Hours')
plt.xlabel('Sleep Duration (hours)')
plt.ylabel('Study Hours')
plt.show()
```

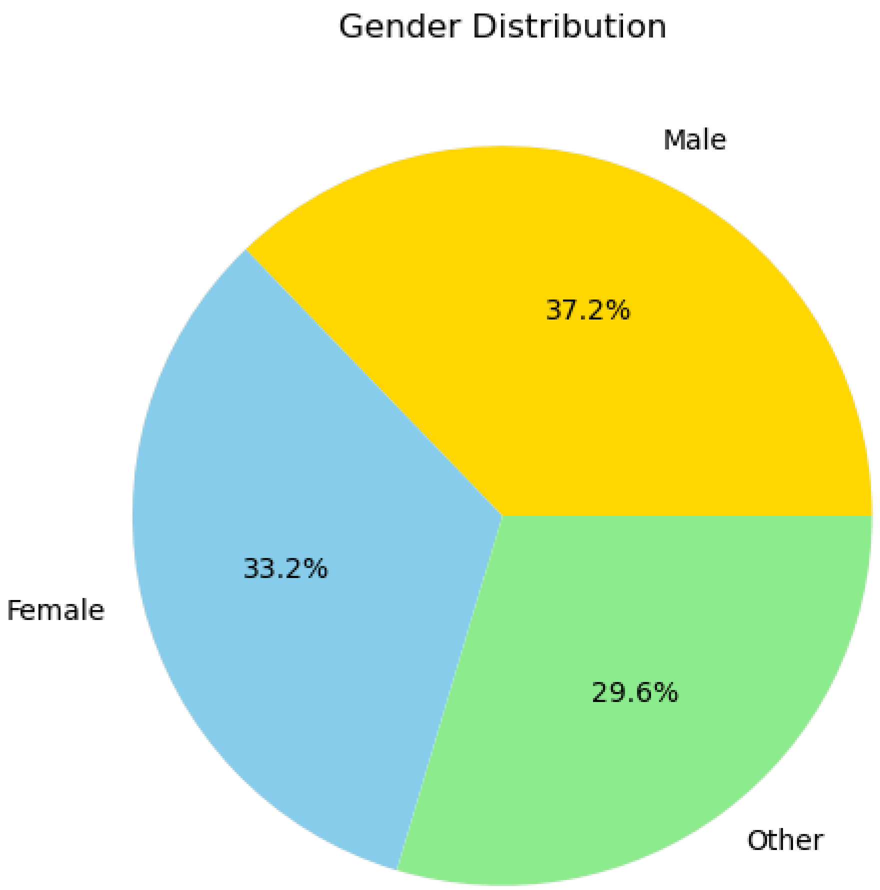


In [4]:

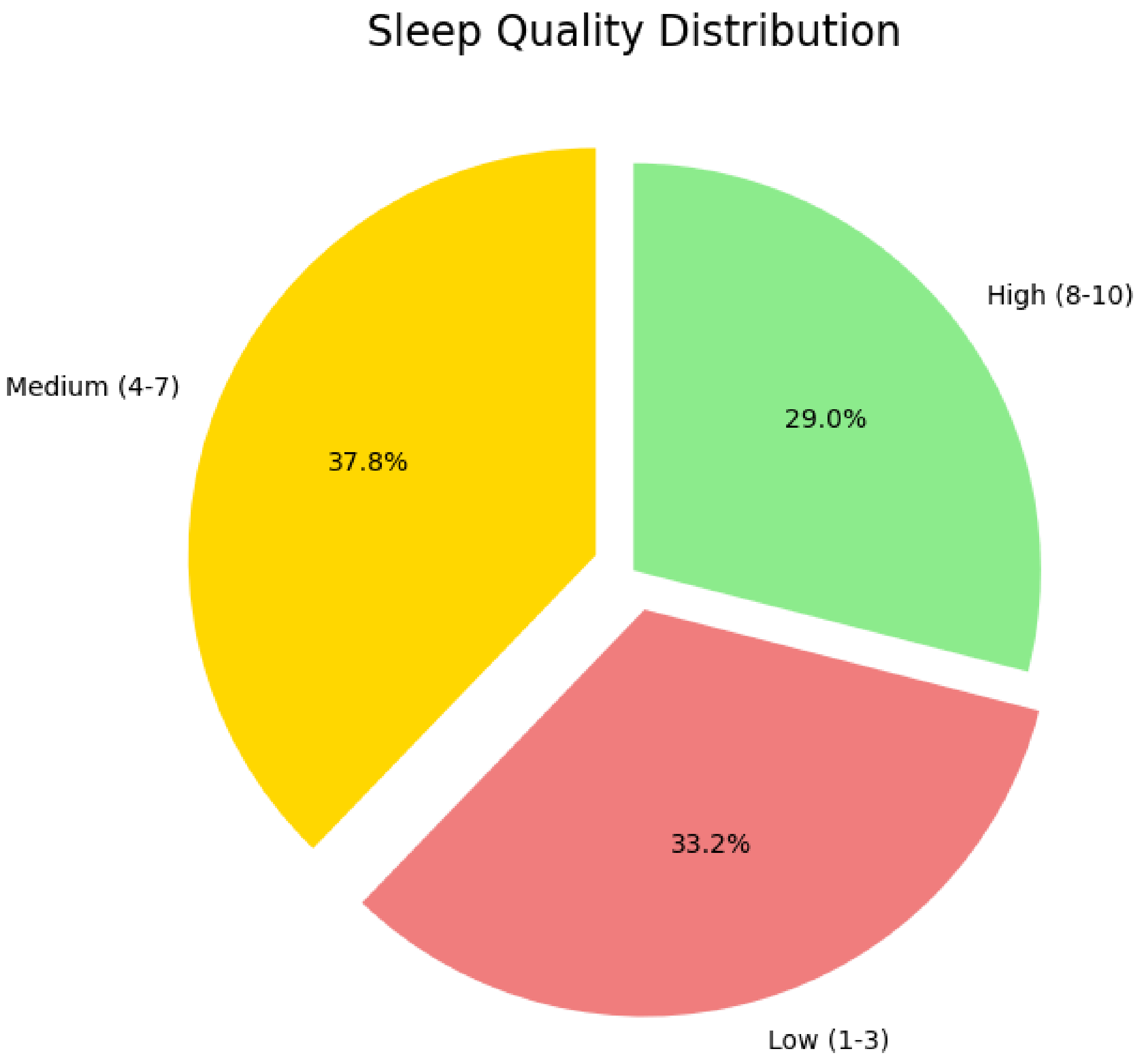
```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('47_student_sleep_patterns.csv')
avg_sleep = data.groupby('University_Year')['Sleep_Duration'].mean()

avg_sleep.plot(kind='bar', color='teal', figsize=(8, 6))
plt.title('Average Sleep Duration by University Year')
plt.xlabel('University Year')
plt.ylabel('Average Sleep Duration (hours)')
plt.xticks(rotation=45)
plt.show()
```



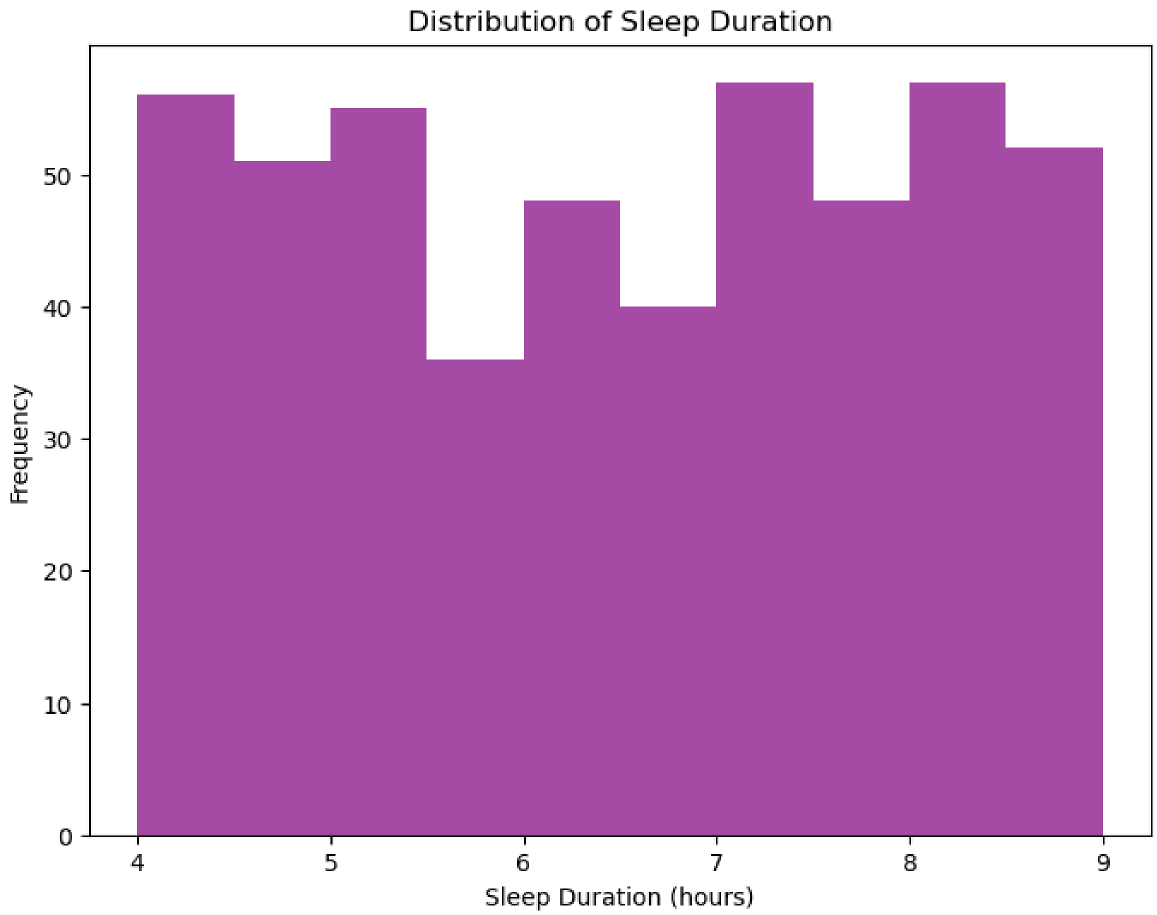
In [23]:
```
gender_counts = data['Gender'].value_counts()

gender_counts.plot(kind='pie', autopct='%1.1f%%', figsize=(6, 6), colors=['gold'
plt.title('Gender Distribution')
plt.ylabel('')
plt.show()
```



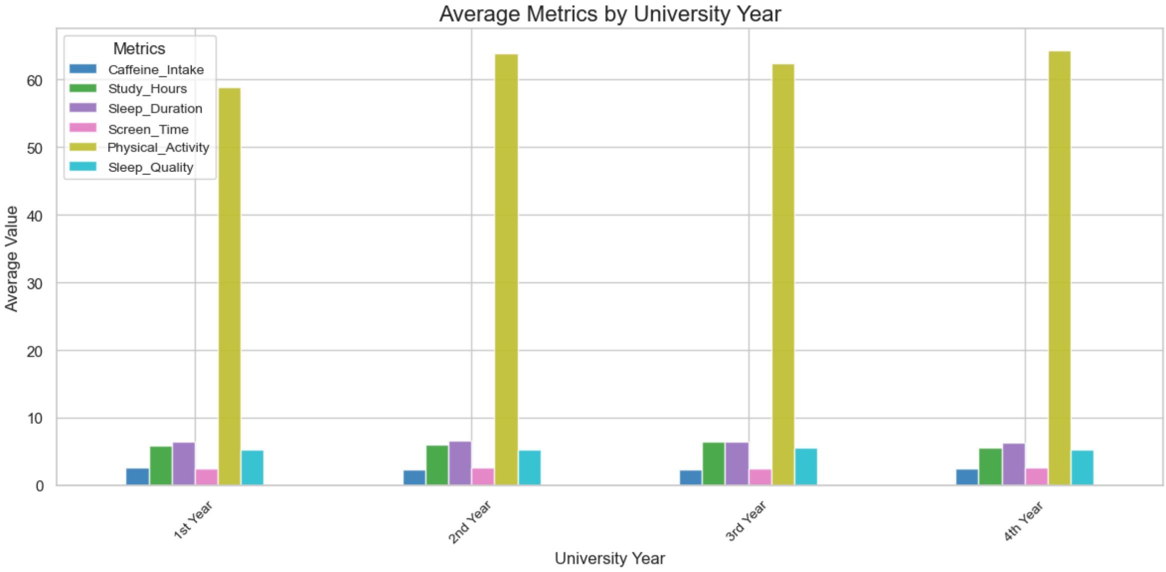
In [5]:
```
sleep_quality_categories = pd.cut(data['Sleep_Quality'], bins=[0, 3, 7, 10], lab
sleep_quality_counts = sleep_quality_categories.value_counts()
plt.figure(figsize=(8, 6))
sleep_quality_counts.plot(kind='pie', autopct='%1.1f%%', colors=['gold', 'lightc
plt.title('Sleep Quality Distribution', fontsize=16)
plt.ylabel('')
plt.tight_layout()
plt.show()
```



In [25]:
```
plt.figure(figsize=(8, 6))
plt.hist(data['Sleep_Duration'], bins=10, color='purple', alpha=0.7)
plt.title('Distribution of Sleep Duration')
plt.xlabel('Sleep Duration (hours)')
plt.ylabel('Frequency')
plt.show()
```

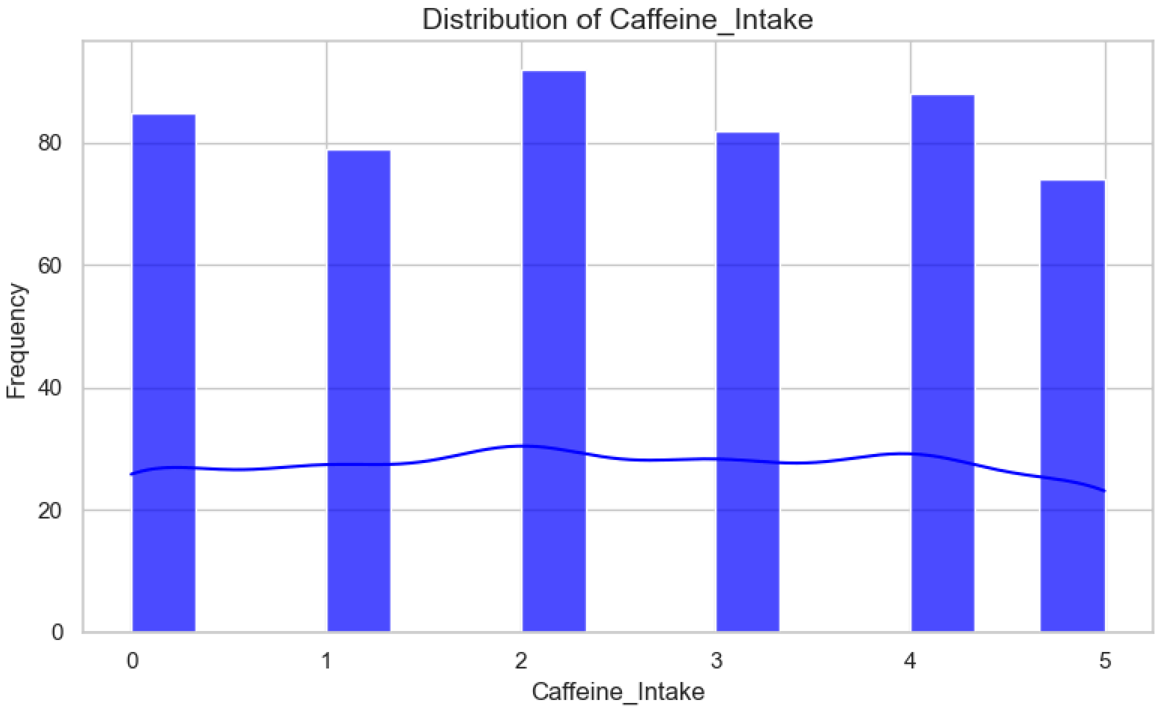


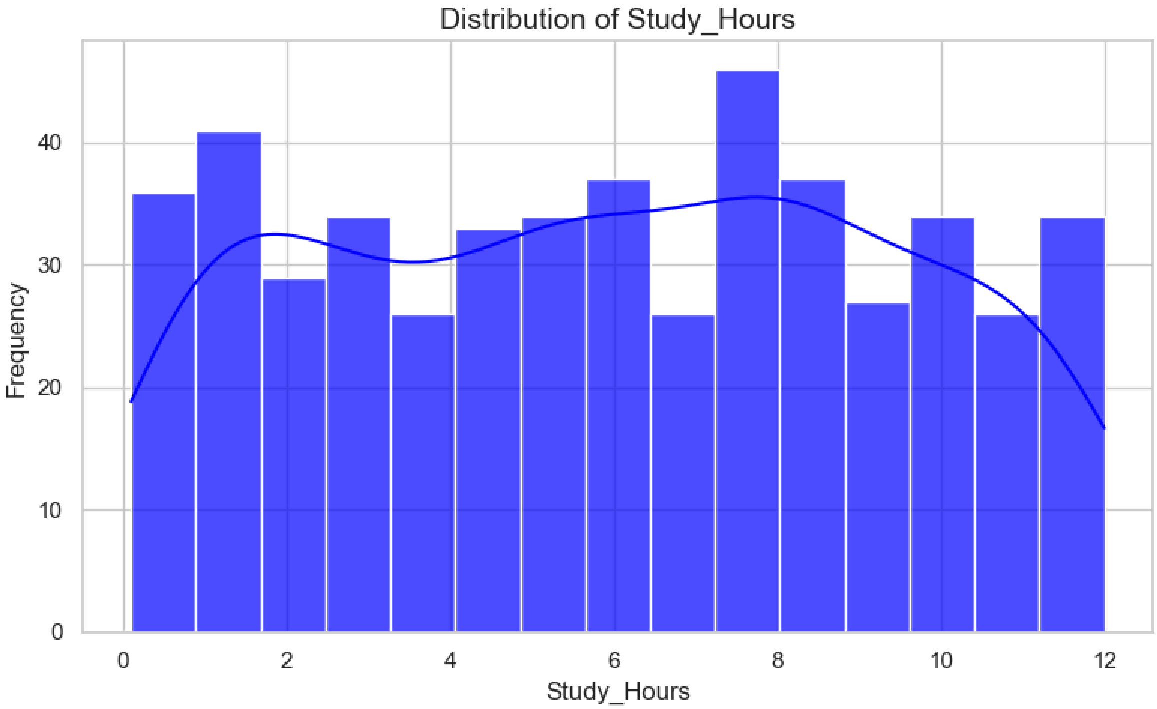
In [20]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv('47_student_sleep_patterns.csv')

# Strip spaces from column names
data.columns = data.columns.str.strip()

# Columns to analyze
columns_to_analyze = [
    'Caffeine_Intake', 'Study_Hours', 'Sleep_Duration',
    'Screen_Time', 'Physical_Activity', 'Sleep_Quality'
]

# Check for missing columns
missing_columns = [col for col in columns_to_analyze if col not in data.columns]

if missing_columns:
    print(f"Error: Missing columns: {missing_columns}")
else:
    # Convert relevant columns to numeric and handle non-numeric data
    data[columns_to_analyze] = data[columns_to_analyze].apply(pd.to_numeric, err
    data.dropna(subset=columns_to_analyze, inplace=True)

    # Grouped bar chart: Mean values grouped by University Year
    grouped_data = data.groupby('University_Year').agg({col: 'mean' for col in c

    sns.set(style="whitegrid")
    grouped_data.set_index('University_Year', inplace=True)
    grouped_data.plot(kind='bar', figsize=(12, 6), alpha=0.85, colormap='tab10')

    plt.title('Average Metrics by University Year', fontsize=16)
    plt.ylabel('Average Value', fontsize=12)
    plt.xlabel('University Year', fontsize=12)
    plt.xticks(rotation=45, fontsize=10)
    plt.legend(title="Metrics", fontsize=10)
    plt.tight_layout()
    plt.show()
```

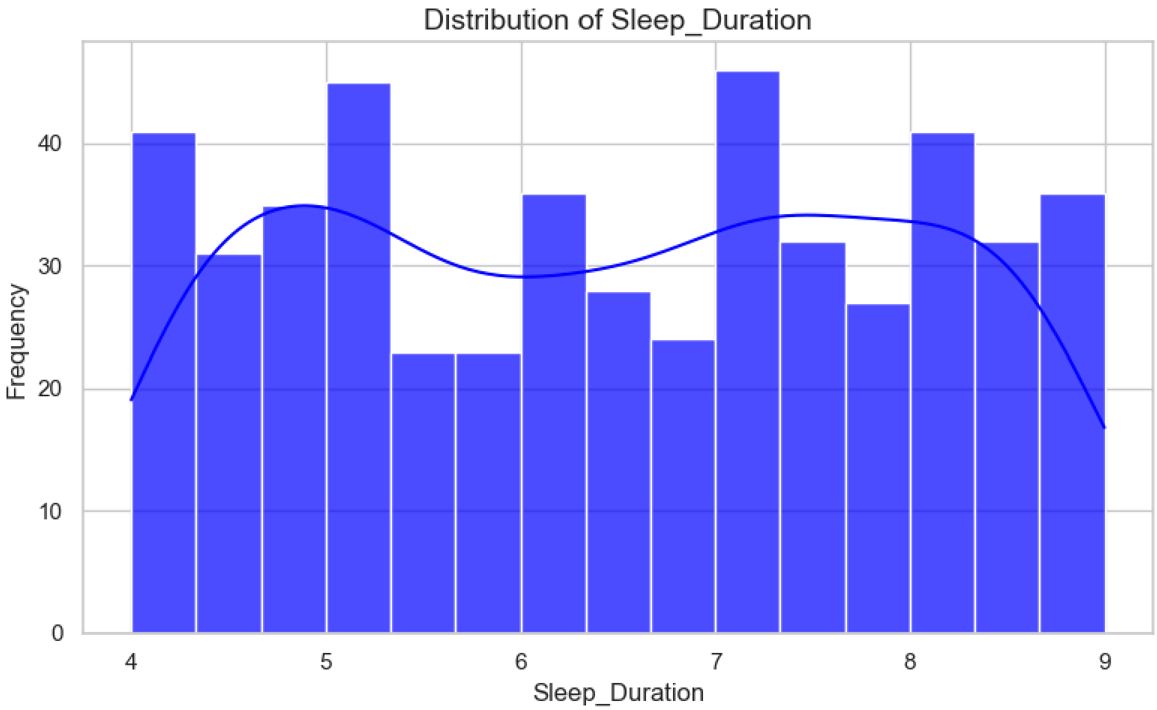


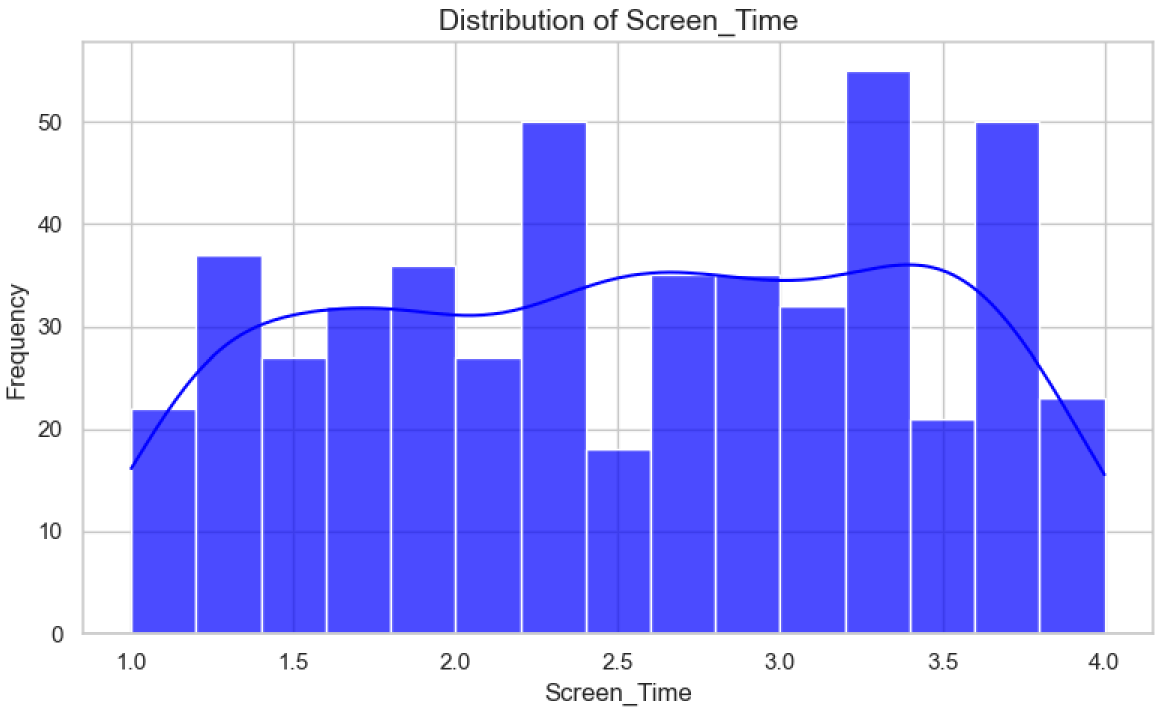
In [21]:

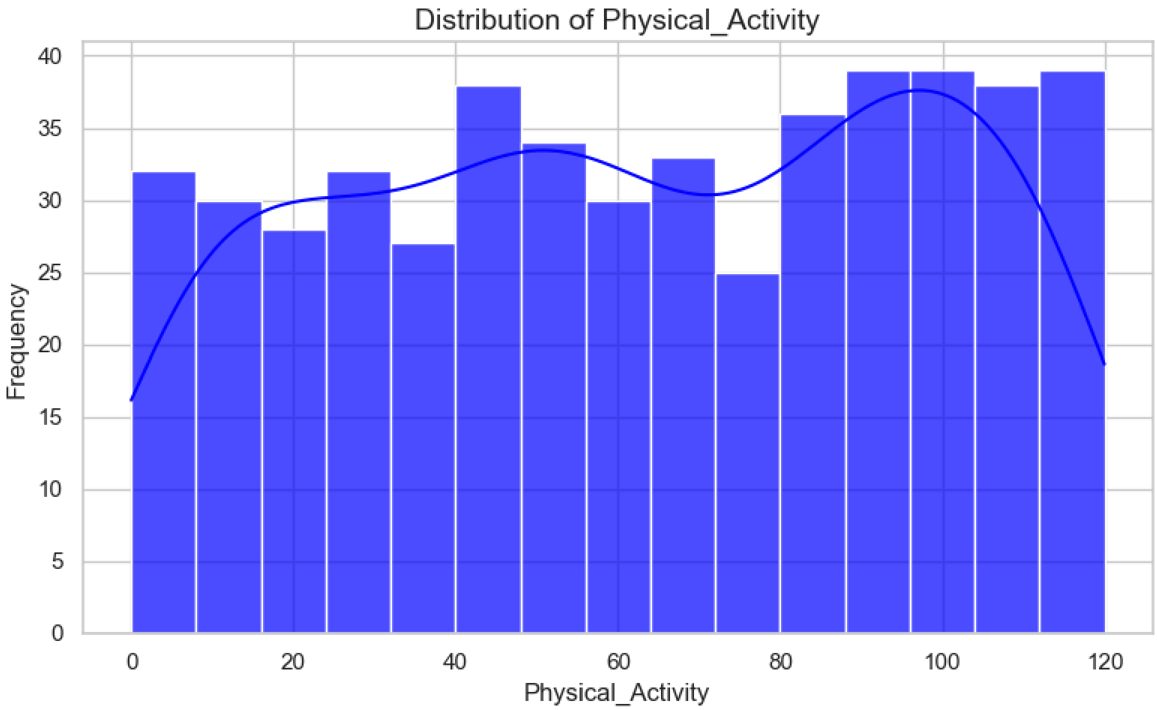
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv('47_student_sleep_patterns.csv')

# Strip spaces from column names
data.columns = data.columns.str.strip()

# Columns to analyze
columns_to_analyze = [
    'Caffeine_Intake', 'Study_Hours', 'Sleep_Duration',
    'Screen_Time', 'Physical_Activity', 'Sleep_Quality'
]

# Check for missing columns
missing_columns = [col for col in columns_to_analyze if col not in data.columns]

if missing_columns:
    print(f"Error: Missing columns: {missing_columns}")
else:
    # Convert relevant columns to numeric and handle non-numeric data
    data[columns_to_analyze] = data[columns_to_analyze].apply(pd.to_numeric, err
    data.dropna(subset=columns_to_analyze, inplace=True)

    # Histograms for each variable
    for col in columns_to_analyze:
        plt.figure(figsize=(8, 5))
        sns.histplot(data[col], kde=True, bins=15, color='blue', alpha=0.7)
        plt.title(f'Distribution of {col}', fontsize=14)
        plt.xlabel(col, fontsize=12)
        plt.ylabel('Frequency', fontsize=12)
        plt.tight_layout()
        plt.show()
```

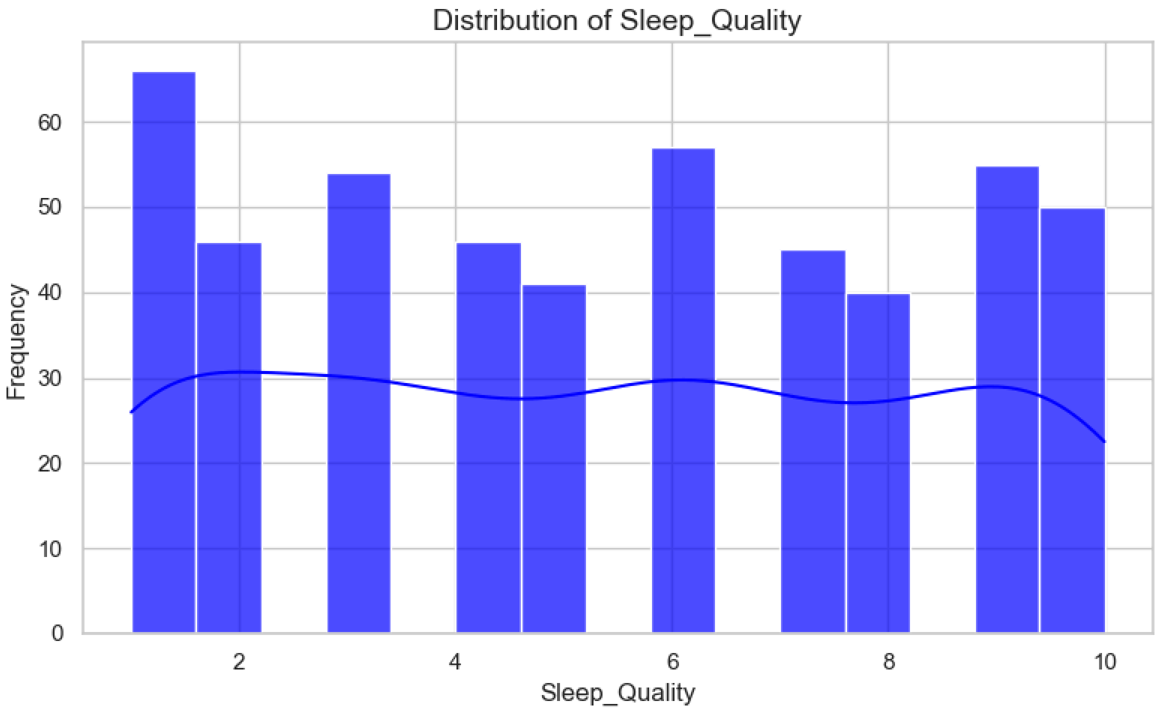





Distribution of Sleep_Duration
Frequency
0
10
20
30
40
4
5
6
7
8
9
Sleep_Duration

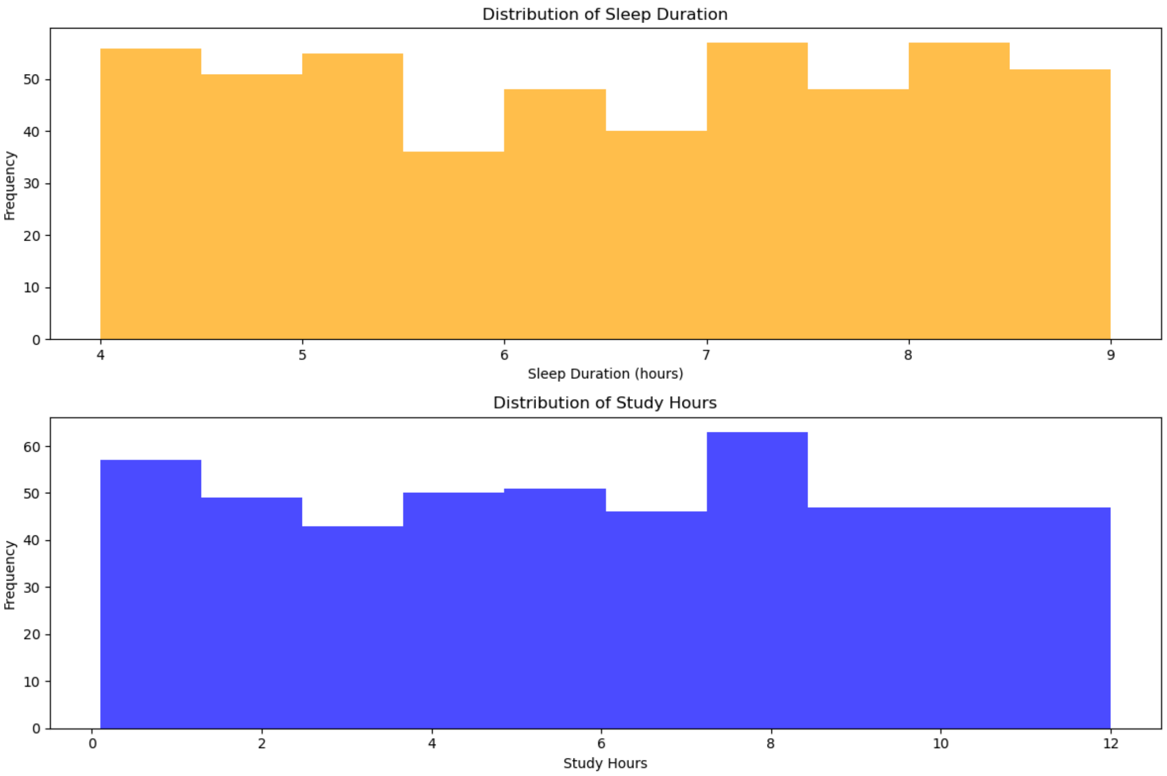

Distribution of Screen_Time
Frequency
0
10
20
30
40
50
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Screen_Time





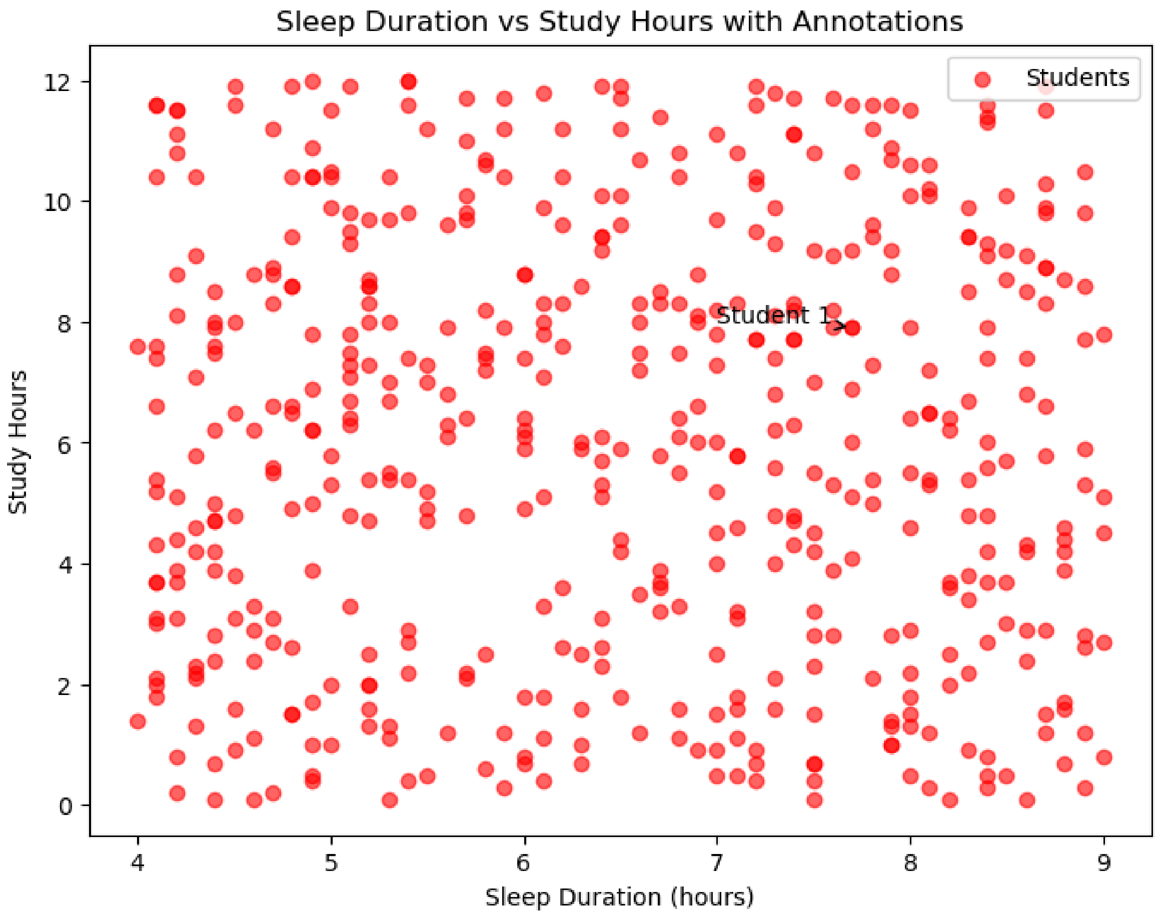
In [26]:

```
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.hist(data['Sleep_Duration'], bins=10, color='orange', alpha=0.7)
plt.title('Distribution of Sleep Duration')
plt.xlabel('Sleep Duration (hours)')
plt.ylabel('Frequency')

plt.subplot(2, 1, 2)
plt.hist(data['Study_Hours'], bins=10, color='blue', alpha=0.7)
plt.title('Distribution of Study Hours')
plt.xlabel('Study Hours')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



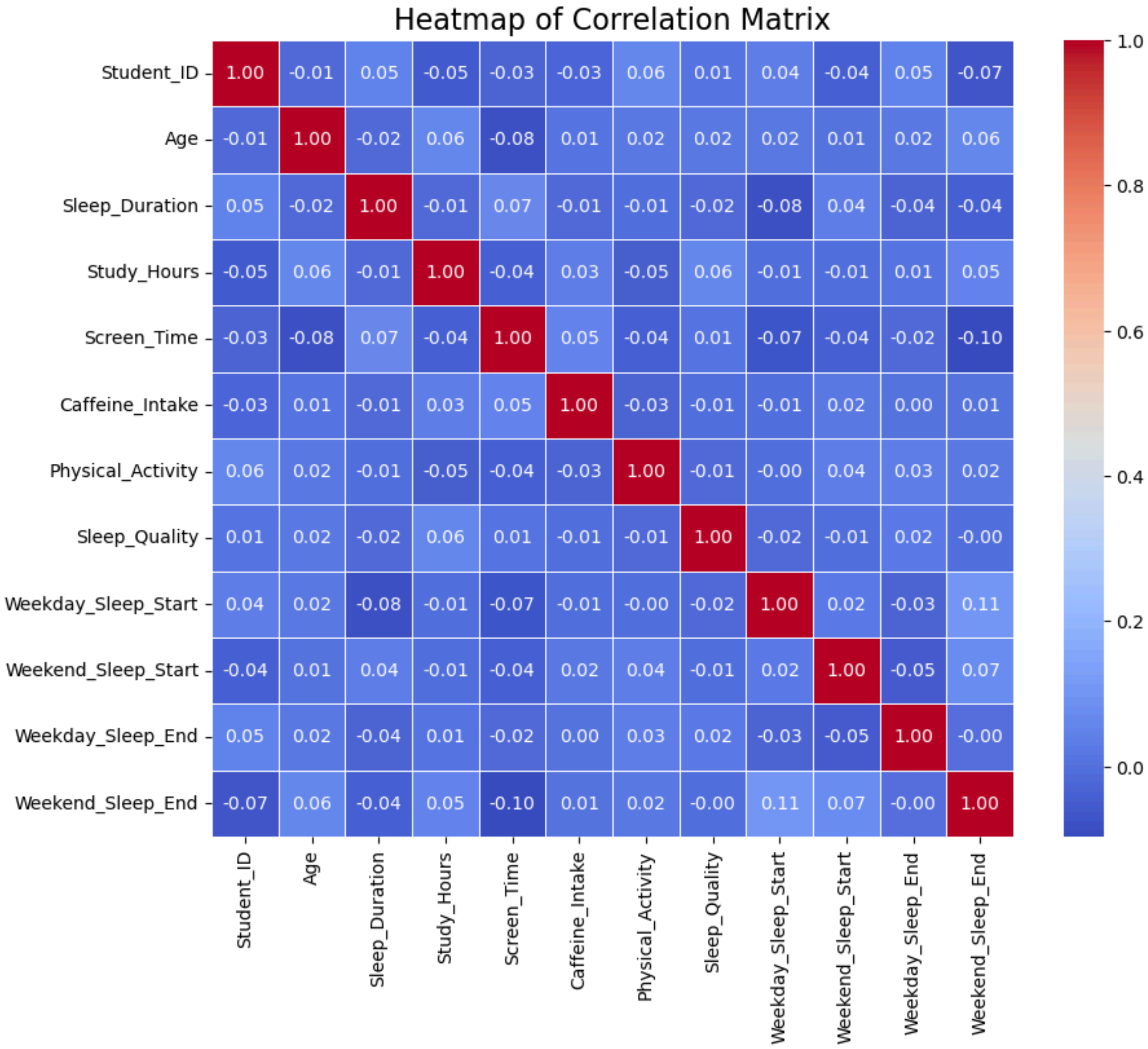
In [28]:

```
plt.figure(figsize=(8, 6))
plt.scatter(data['Sleep_Duration'], data['Study_Hours'], c='red', alpha=0.6, lab

plt.annotate('Student 1', xy=(data['Sleep_Duration'][0], data['Study_Hours'][0])
             xytext=(7, 8), arrowprops=dict(facecolor='black', arrowstyle='->'))

plt.title('Sleep Duration vs Study Hours with Annotations')
plt.xlabel('Sleep Duration (hours)')
plt.ylabel('Study Hours')
plt.legend()
plt.show()
```



In [9]:

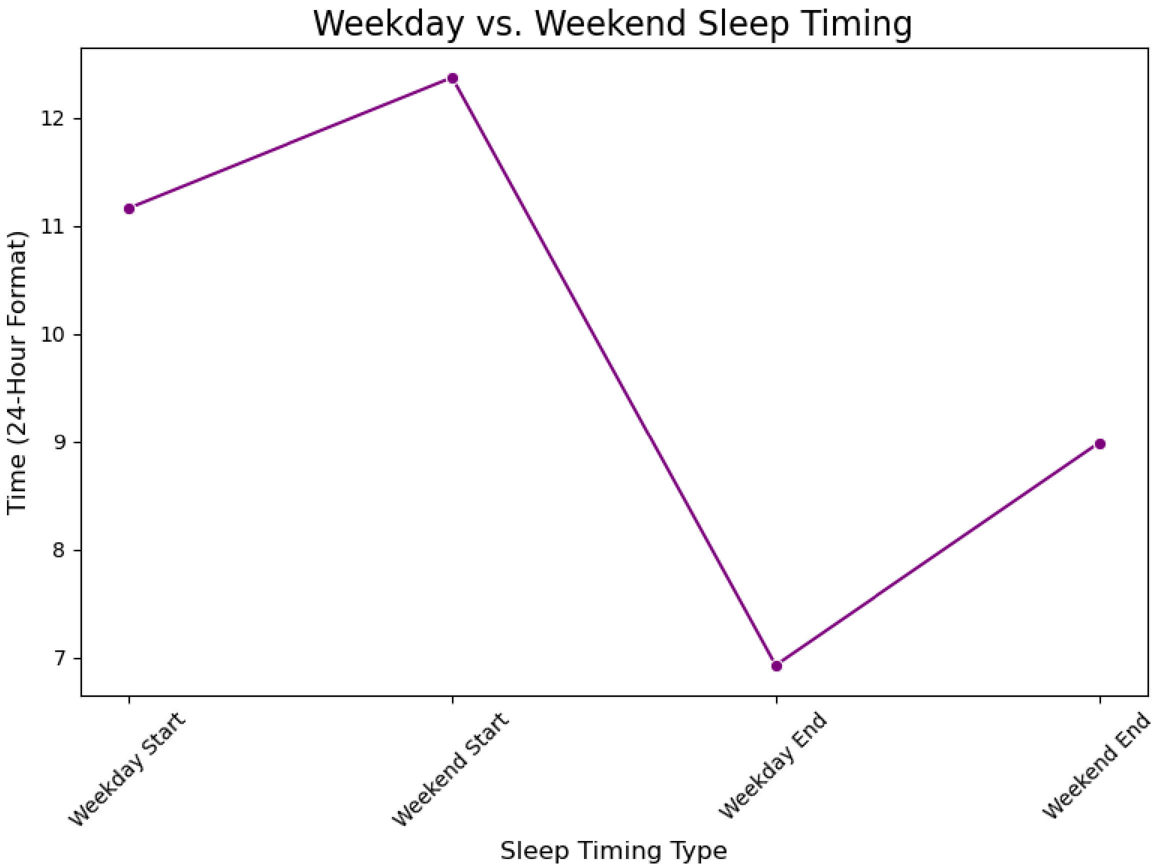
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('47_student_sleep_patterns.csv')

numeric_data = df.select_dtypes(include=['number'])

correlation_matrix = numeric_data.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidt
plt.title("Heatmap of Correlation Matrix", fontsize=16)
plt.show()
```



	Student_ID	Age	Sleep_Duration	Study_Hours	Screen_Time	Caffeine_Intake	Physical_Activity	Sleep_Quality	Weekday_Sleep_Start	Weekend_Sleep_Start	Weekday_Sleep_End	Weekend_Sleep_End
Student_ID	1.00	-0.01	0.05	-0.05	-0.03	-0.03	0.06	0.01	0.04	-0.04	0.05	-0.07
Age	-0.01	1.00	-0.02	0.06	-0.08	0.01	0.02	0.02	0.02	0.01	0.02	0.06
Sleep_Duration	0.05	-0.02	1.00	-0.01	0.07	-0.01	-0.01	-0.02	-0.08	0.04	-0.04	-0.04
Study_Hours	-0.05	0.06	-0.01	1.00	-0.04	0.03	-0.05	0.06	-0.01	-0.01	0.01	0.05
Screen_Time	-0.03	-0.08	0.07	-0.04	1.00	0.05	-0.04	0.01	-0.07	-0.04	-0.02	-0.10
Caffeine_Intake	-0.03	0.01	-0.01	0.03	0.05	1.00	-0.03	-0.01	-0.01	0.02	0.00	0.01
Physical_Activity	0.06	0.02	-0.01	-0.05	-0.04	-0.03	1.00	-0.01	-0.00	0.04	0.03	0.02
Sleep_Quality	0.01	0.02	-0.02	0.06	0.01	-0.01	-0.01	1.00	-0.02	-0.01	0.02	-0.00
Weekday_Sleep_Start	0.04	0.02	-0.08	-0.01	-0.07	-0.01	-0.00	-0.02	1.00	0.02	-0.03	0.11
Weekend_Sleep_Start	-0.04	0.01	0.04	-0.01	-0.04	0.02	0.04	-0.01	0.02	1.00	-0.05	0.07
Weekday_Sleep_End	0.05	0.02	-0.04	0.01	-0.02	0.00	0.03	0.02	-0.03	-0.05	1.00	-0.00
Weekend_Sleep_End	-0.07	0.06	-0.04	0.05	-0.10	0.01	0.02	-0.00	0.11	0.07	-0.00	1.00

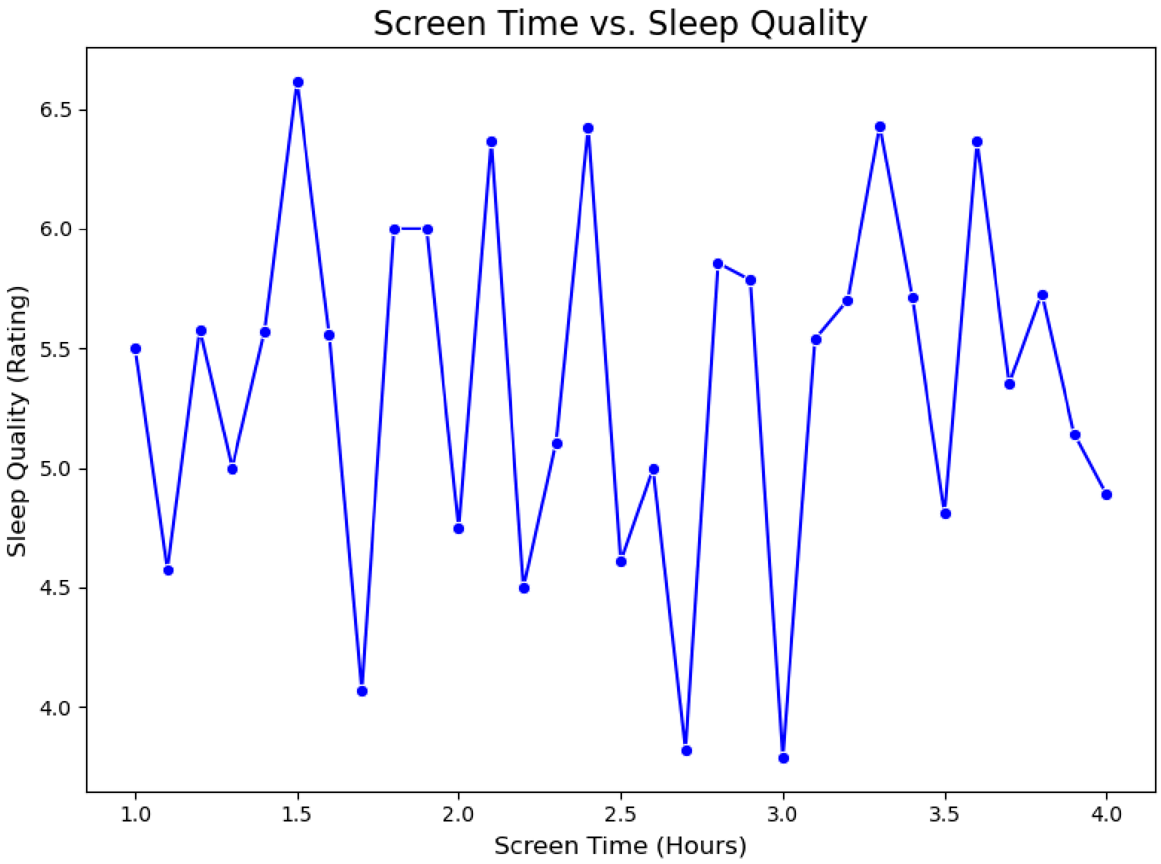
In [7]:

```
import seaborn as sns
weekday_start = data['Weekday_Sleep_Start'].mean()
weekend_start = data['Weekend_Sleep_Start'].mean()
weekday_end = data['Weekday_Sleep_End'].mean()
weekend_end = data['Weekend_Sleep_End'].mean()

sleep_timing = pd.DataFrame({
    'Type': ['Weekday Start', 'Weekend Start', 'Weekday End', 'Weekend End'],
    'Time': [weekday_start, weekend_start, weekday_end, weekend_end]
})

plt.figure(figsize=(8, 6))
sns.lineplot(x='Type', y='Time', data=sleep_timing, marker='o', color='purple')
plt.title('Weekday vs. Weekend Sleep Timing', fontsize=16)
plt.xlabel('Sleep Timing Type', fontsize=12)
plt.ylabel('Time (24-Hour Format)', fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.tight_layout()
plt.show()
```



In [9]:

```
screen_time_quality = data.groupby('Screen_Time')['Sleep_Quality'].mean().reset_
plt.figure(figsize=(8, 6))
sns.lineplot(x='Screen_Time', y='Sleep_Quality', data=screen_time_quality, marke
plt.title('Screen Time vs. Sleep Quality', fontsize=16)
plt.xlabel('Screen Time (Hours)', fontsize=12)
plt.ylabel('Sleep Quality (Rating)', fontsize=12)
plt.tight_layout()
plt.show()
```


Screen Time vs. Sleep Quality
Sleep Quality (Rating)
6.5
6.0
5.5
5.0
4.5
4.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Screen Time (Hours)

```
In [ ]:
```