Combating Social Media Addiction With Data-Driven Insights

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

```
df = pd.read_csv("/content/Students Social Media Addiction (1).csv")
```

Data Understanding & Cleaning

```
print(df.head(5))   # Show first 5 rows
print(df.info())   #  Checking Column types & null
print(df.describe()) # Show Statistical summary
```

```
1           Twitter                          No                    7.5
2            TikTok                         Yes                    5.0
3           YouTube                          No                    7.0
4          Facebook                         Yes                    6.0

   Mental_Health_Score Relationship_Status  Conflicts_Over_Social_Media  \
0                    6     In Relationship                            3
1                    8              Single                            0
```

```
4                7
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 705 entries, 0 to 704
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Student_ID                    705 non-null    int64
 1   Age                           705 non-null    int64
 2   Gender                        705 non-null    object
 3   Academic_Level                705 non-null    object
 4   Country                       705 non-null    object
 5   Avg_Daily_Usage_Hours         705 non-null    float64
 6   Most_Used_Platform            705 non-null    object
 7   Affects_Academic_Performance  705 non-null    object
 8   Sleep_Hours_Per_Night         705 non-null    float64
 9   Mental_Health_Score           705 non-null    int64
 10  Relationship_Status           705 non-null    object
 11  Conflicts_Over_Social_Media   705 non-null    int64
 12  Addicted_Score                705 non-null    int64
dtypes: float64(2), int64(5), object(6)
memory usage: 71.7+ KB
None
       Student_ID         Age  Avg_Daily_Usage_Hours  Sleep_Hours_Per_Night  \
count  705.000000  705.000000             705.000000             705.000000
mean   353.000000   20.659574               4.918723               6.868936
std    203.660256    1.399217               1.257395               1.126848
min      1.000000   18.000000               1.500000               3.800000
25%    177.000000   19.000000               4.100000               6.000000
50%    353.000000   21.000000               4.800000               6.900000
75%    529.000000   22.000000               5.800000               7.700000
max    705.000000   24.000000               8.500000               9.600000

       Mental_Health_Score  Conflicts_Over_Social_Media  Addicted_Score
count           705.000000                   705.000000      705.000000
mean              6.226950                     2.849645        6.436879
std               1.105055                     0.957968        1.587165
min               4.000000                     0.000000        2.000000
25%               5.000000                     2.000000        5.000000
50%               6.000000                     3.000000        7.000000
75%               7.000000                     4.000000        8.000000
max               9.000000                     5.000000        9.000000
```

```
print(df.isnull().sum()) # yes it show no null values in dataset
```

```
Student_ID                      0
Age                             0
Gender                          0
Academic_Level                  0
Country                         0
Avg_Daily_Usage_Hours           0
Most_Used_Platform              0
Affects_Academic_Performance    0
Sleep_Hours_Per_Night           0
Mental_Health_Score             0
Relationship_Status             0
Conflicts_Over_Social_Media     0
Addicted_Score                  0
dtype: int64
```

```
# Fill missing values if any
df = df.fillna(df.mean(numeric_only=True))
```

```
# Check data types
print("\nData types:")
print(df.dtypes)
```

```

Data types:
Student_ID                      int64
Age                             int64
Gender                         object
Academic_Level                 object
Country                        object
Avg_Daily_Usage_Hours         float64
Most_Used_Platform             object
Affects_Academic_Performance   object
Sleep_Hours_Per_Night         float64
Mental_Health_Score             int64
Relationship_Status            object
```

```
Conflicts_Over_Social_Media       int64
Addicted_Score                    int64
dtype: object
```

```
# Check for duplicates
duplicates = df.duplicated().sum()
print(f"Duplicate rows: {duplicates}")
if duplicates > 0:
    df = df.drop_duplicates()
    print("Duplicates removed")
```

```
Duplicate rows: 0
```

Exploratory Data Analysis (EDA)

```
#Understand distribution
print(df['Gender'].value_counts())
print(df['Age'].describe())
```

```
Gender
Female    353
Male      352
Name: count, dtype: int64
count    705.000000
mean      20.659574
std        1.399217
min       18.000000
25%       19.000000
50%       21.000000
75%       22.000000
max       24.000000
Name: Age, dtype: float64
```

Understand Relationships Between:

```
# Age vs Daily Usage
df.groupby('Age')['Avg_Daily_Usage_Hours'].mean()
```

Age	Avg_Daily_Usage_Hours
18	5.385714
19	5.120245
20	4.930303
21	4.950641
22	4.676190
23	4.508824
24	5.046154

dtype: float64

```
# Gender Vs Daily Usage
df.groupby('Gender')['Avg_Daily_Usage_Hours'].mean()
```

Gender	Avg_Daily_Usage_Hours
Female	5.011048
Male	4.826136

dtype: float64

```
# Sleep vs Usage (correlation check)
df[['Avg_Daily_Usage_Hours','Sleep_Hours_Per_Night']].corr()
```

	Avg_Daily_Usage_Hours	Sleep_Hours_Per_Night
Avg_Daily_Usage_Hours	1.000000	-0.790582
Sleep_Hours_Per_Night	-0.790582	1.000000

```
# Academic Level Vs Daily Usage
df.groupby('Academic_Level')['Avg_Daily_Usage_Hours'].mean()
```

Academic_Level	Avg_Daily_Usage_Hours
Graduate	4.776923
High School	5.544444
Undergraduate	5.001416

dtype: float64

```
# Country Vs Daily Usage
df.groupby('Country')['Avg_Daily_Usage_Hours'].mean()
```

Country	Avg_Daily_Usage_Hours
Afghanistan	2.9
Albania	4.7
Andorra	5.3
Argentina	5.5
Armenia	5.9
...	...
Uzbekistan	5.5
Vatican City	4.4
Venezuela	3.3
Vietnam	3.6
Yemen	4.7

110 rows × 1 columns

dtype: float64

```
# Correlation between Addicted Score to ( Daily Usage Hours, Conflicted Over Social Media, Sleep Hours And Mental H
correlation_usage_addiction = df['Addicted_Score'].corr(df['Avg_Daily_Usage_Hours'])
display(f"Correlation between Addicted_Score and Avg_Daily_Usage_Hours: {correlation_usage_addiction}")

correlation_conflicts_addiction = df['Addicted_Score'].corr(df['Conflicts_Over_Social_Media'])
display(f"Correlation between Addicted_Score and Conflicts_Over_Social_Media: {correlation_conflicts_addiction}")

correlation_sleep_addiction = df['Addicted_Score'].corr(df['Sleep_Hours_Per_Night'])
display(f"Correlation between Addicted_Score and Sleep_Hours_Per_Night: {correlation_sleep_addiction}")
```

```
correlation_mental_health_addiction = df['Addicted_Score'].corr(df['Mental_Health_Score'])
display(f"Correlation between Addicted_Score and Mental_Health_Score: {correlation_mental_health_addiction}")
```

```
'Correlation between Addicted_Score and Avg_Daily_Usage_Hours: 0.8320001573523091'
'Correlation between Addicted_Score and Conflicts_Over_Social_Media: 0.9335858668503304'
'Correlation between Addicted_Score and Sleep_Hours_Per_Night: -0.7648579747036489'
'Correlation between Addicted_Score and Mental_Health_Score: -0.9450506757277399'
```

Aggregation And Insights

```
# Groupby Genders
print(df.groupby("Gender")['Avg_Daily_Usage_Hours'].mean())
```

```
Gender
Female    5.011048
Male      4.826136
Name: Avg_Daily_Usage_Hours, dtype: float64
```

```
# Groupby Age Groups
df['Age_Group'] = pd.cut(df['Age'], bins=[10,15,20,25], labels=["11-15","16-20","21-25"])
print(df.groupby("Age_Group")['Avg_Daily_Usage_Hours'].mean())
```

```
Age_Group
11-15         NaN
16-20    5.039474
21-25    4.804959
Name: Avg_Daily_Usage_Hours, dtype: float64
/tmp/ipython-input-111501381.py:3: FutureWarning: The default of observed=False is deprecated and will be changed to True in a f
  print(df.groupby("Age_Group")['Avg_Daily_Usage_Hours'].mean())
```

```
# Groupby Educational Level
print(df.groupby("Academic_Level")['Avg_Daily_Usage_Hours'].mean())
```

```
Academic_Level
Graduate         4.776923
High School      5.544444
Undergraduate    5.001416
Name: Avg_Daily_Usage_Hours, dtype: float64
```

Functions, Loops, and Conditionals

```
# Create Risk Level Classification Function based on usage hours
def classify_risk_level(usage_hours):
    """
    Classify students into addiction risk categories
    Low Risk: Less than 3 hours daily
    Medium Risk: 3-5 hours daily
    High Risk: More than 5 hours daily
    """
    if usage_hours < 3:
        return 'Low'
    elif usage_hours <= 5:
        return 'Medium'
    else:
        return 'High'

df['Risk_Level'] = df['Avg_Daily_Usage_Hours'].apply(classify_risk_level)
```

```
# Suggest digital detox strategies using if-else blocks
def suggest_strategy(risk):
    if risk == "Low":
```

```
        return "Maintain balance"
    elif risk == "Medium":
        return "Digital detox weekends"
    else:
        return "Seek counseling & reduce gradually"


df['Strategy'] = df['Risk_Level'].apply(suggest_strategy)
```

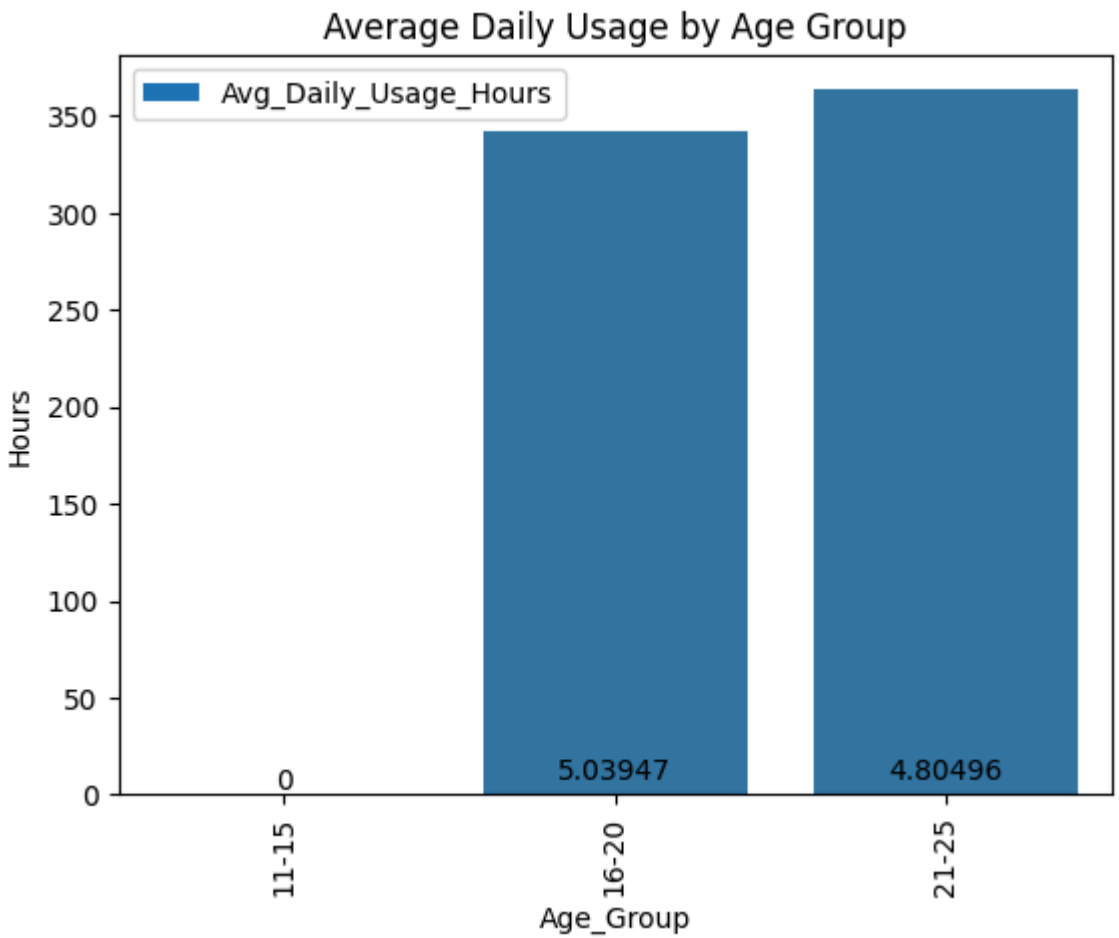
```
# Sleep Quality Category
def sleep_quality(hours):
    if hours >= 8:
        return "Good"
    elif hours >= 6:
        return "Average"
    else:
        return "Poor"

df['Sleep_Quality'] = df['Sleep_Hours_Per_Night'].apply(sleep_quality)
```

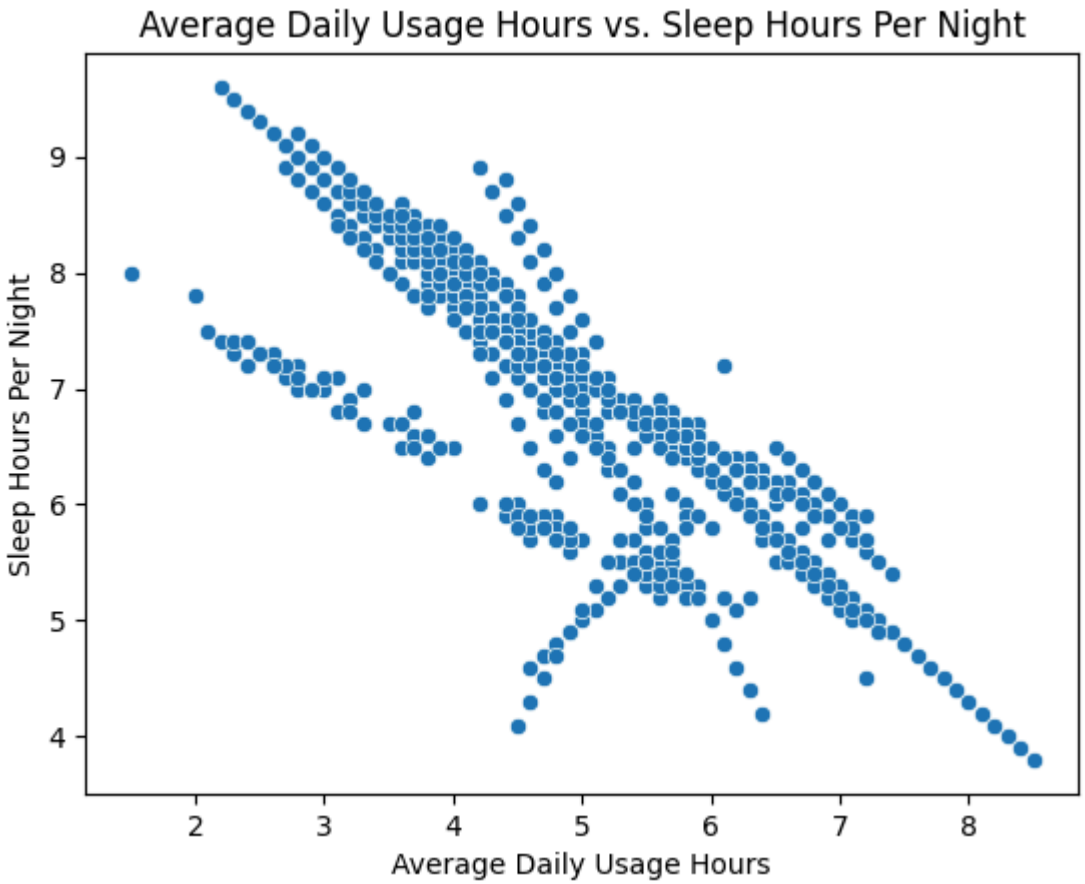
Data Visualization

```
# Bar Chart → Daily Usage by Age Group
df.groupby("Age_Group")['Avg_Daily_Usage_Hours'].mean().plot(kind='bar')
ax = sns.countplot(x='Age_Group', data=df)
ax.bar_label(ax.containers[0])
plt.title("Average Daily Usage by Age Group")
plt.ylabel("Hours")
plt.show()
```

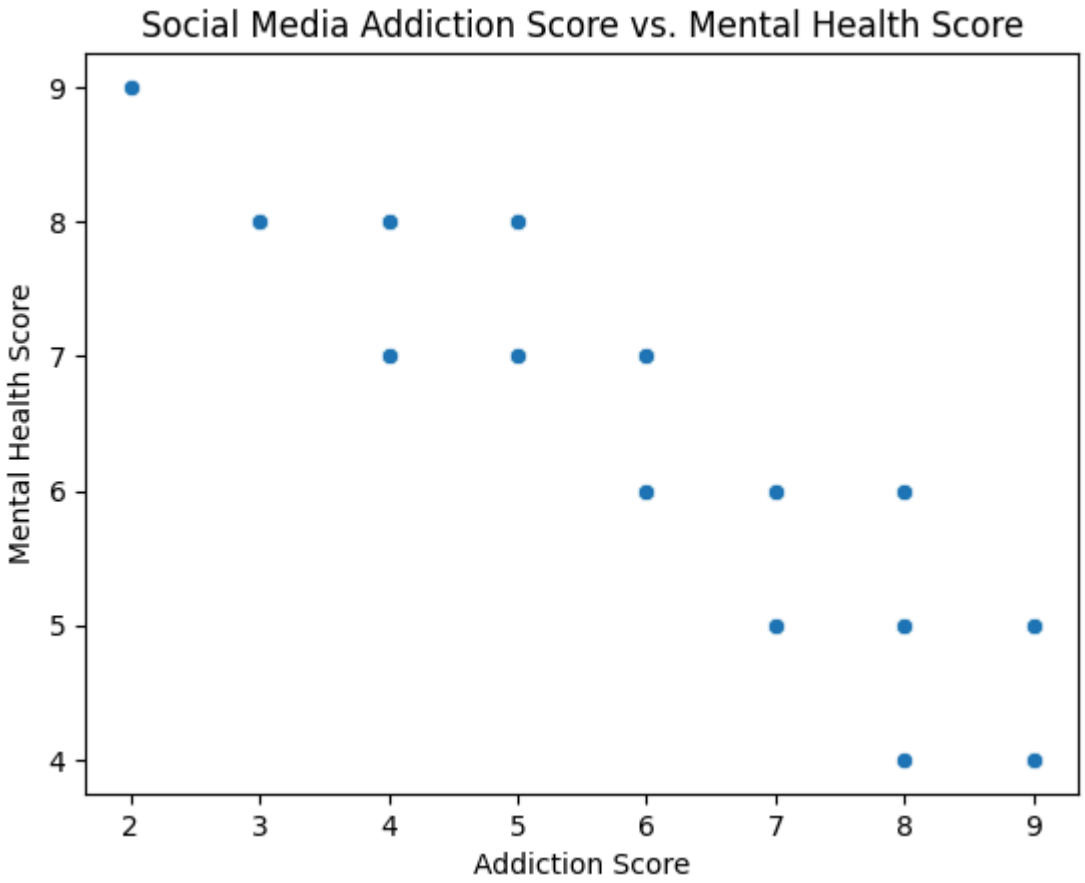
```
/tmp/ipython-input-2523780185.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a
  df.groupby("Age_Group")['Avg_Daily_Usage_Hours'].mean().plot(kind='bar')
```



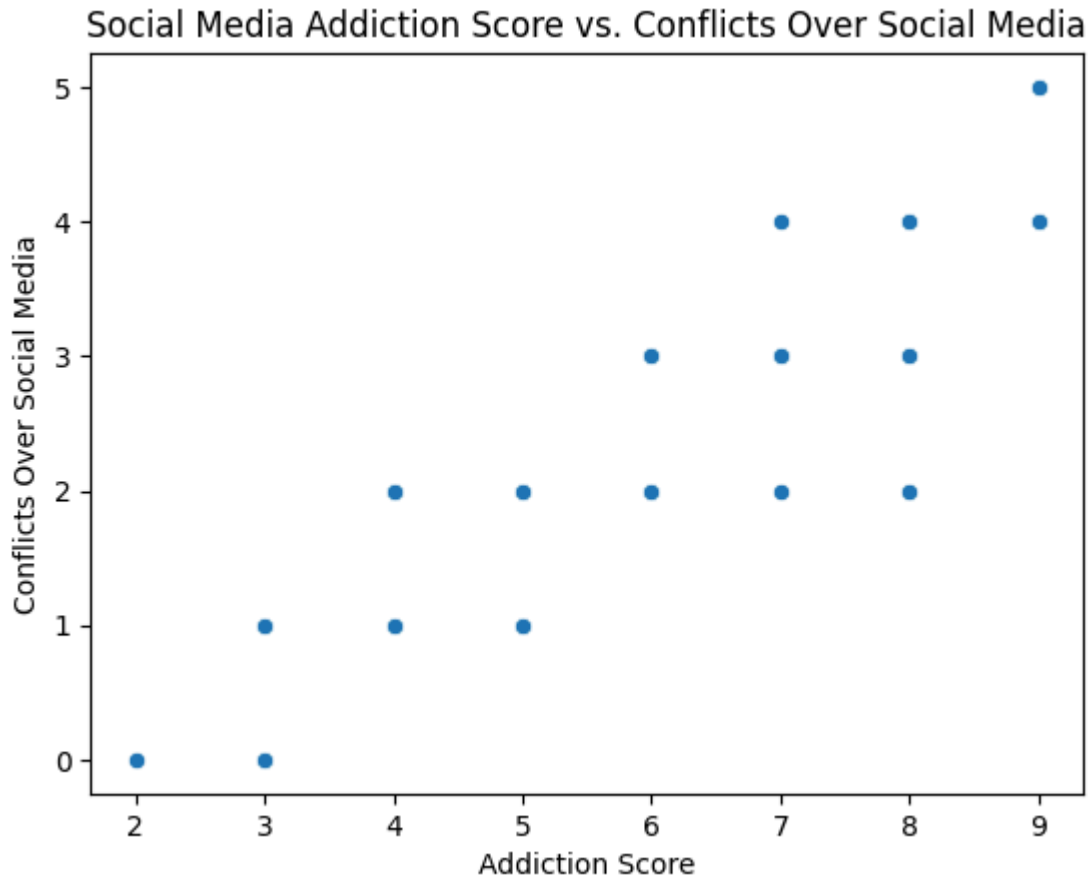

```
sns.scatterplot(x='Avg_Daily_Usage_Hours', y='Sleep_Hours_Per_Night', data=df)
plt.title('Average Daily Usage Hours vs. Sleep Hours Per Night')
plt.xlabel('Average Daily Usage Hours')
plt.ylabel('Sleep Hours Per Night')
plt.show()
```



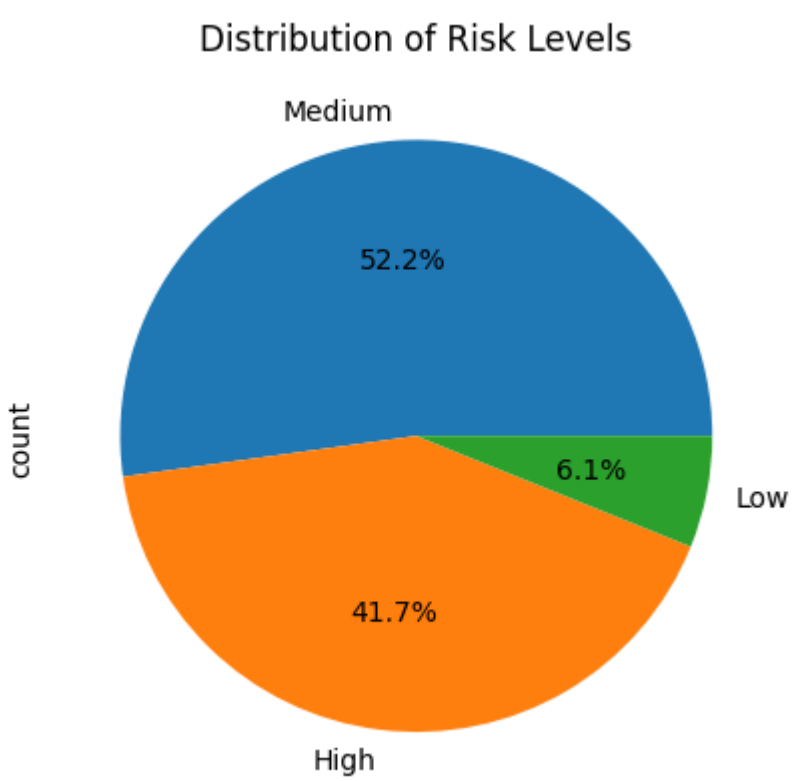

```
sns.scatterplot(x='Addicted_Score', y='Mental_Health_Score', data=df)
plt.title('Social Media Addiction Score vs. Mental Health Score')
plt.xlabel('Addiction Score')
plt.ylabel('Mental Health Score')
plt.show()
```



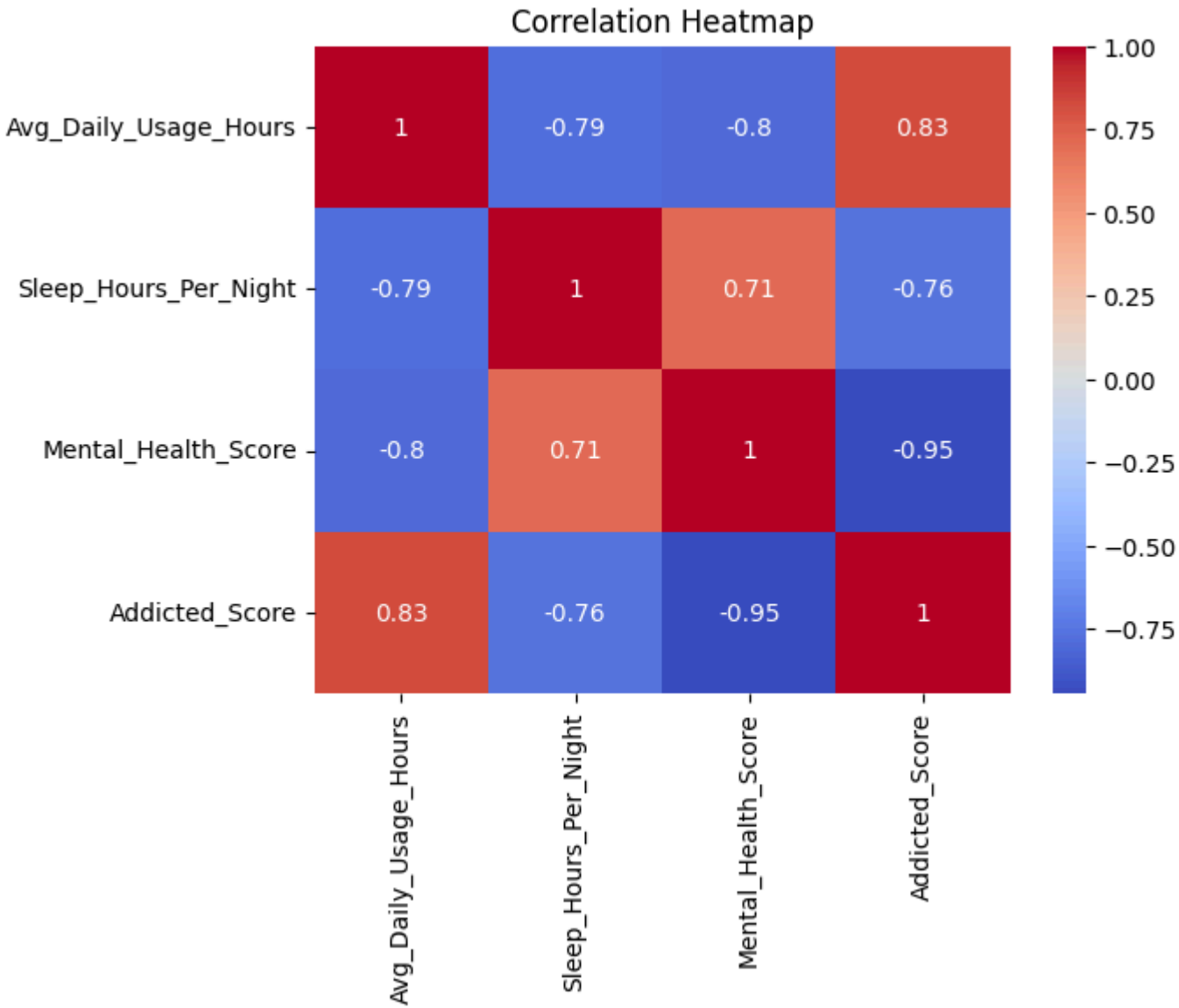

```
sns.scatterplot(x='Addicted_Score', y='Conflicts_Over_Social_Media', data=df)
plt.title('Social Media Addiction Score vs. Conflicts Over Social Media')
plt.xlabel('Addiction Score')
plt.ylabel('Conflicts Over Social Media')
plt.show()
```




```
# Pie Chart → Risk Level Distribution
df['Risk_Level'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title("Distribution of Risk Levels")
plt.show()
```

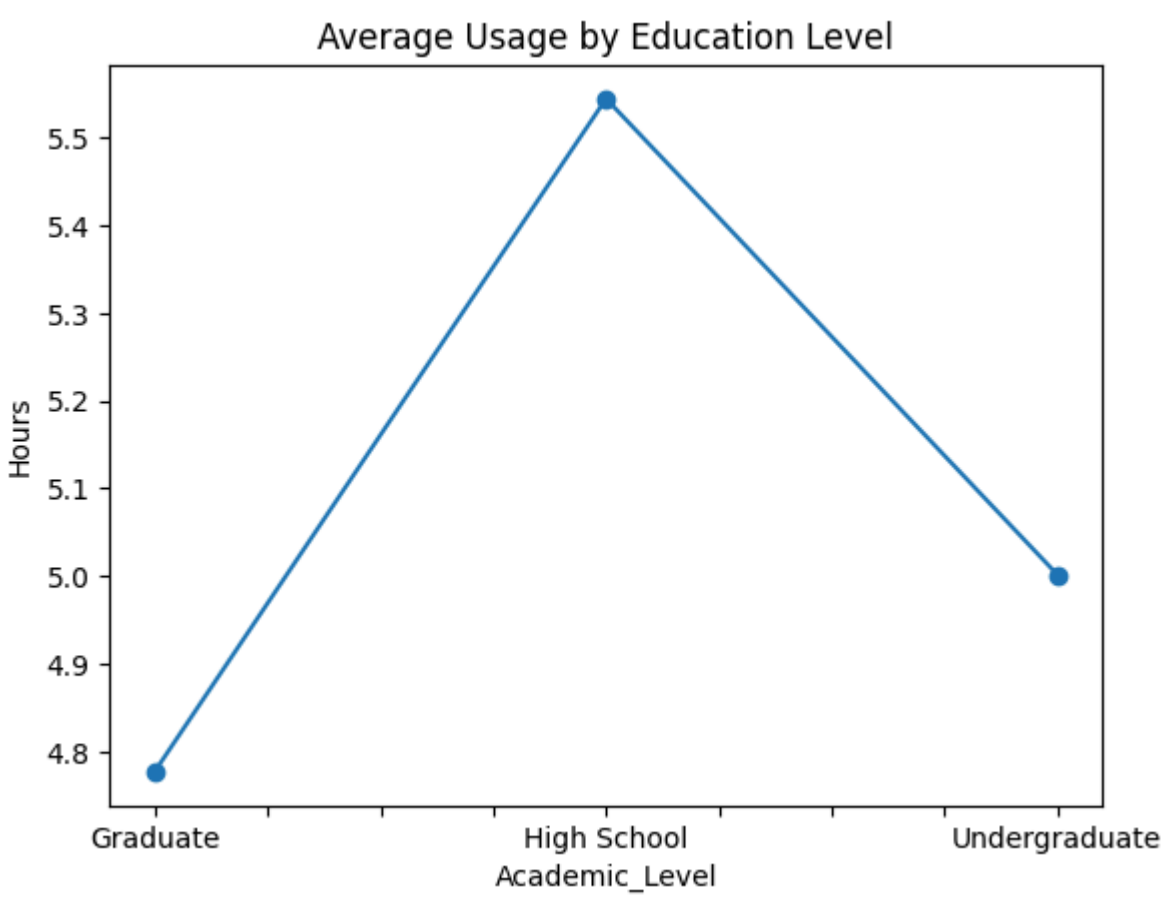



```
# Heatmap → Correlation Between Usage, Sleep, Mental Health and Addiction
sns.heatmap(df[['Avg_Daily_Usage_Hours','Sleep_Hours_Per_Night','Mental_Health_Score', 'Addicted_Score']].corr(), a
plt.title("Correlation Heatmap")
plt.show()
```




```
df.groupby("Academic_Level")['Avg_Daily_Usage_Hours'].mean().plot(kind='line', marker='o')
plt.title("Average Usage by Education Level")
plt.ylabel("Hours")
```

```
plt.show()
```



Final Summary

The project analyzed student social media usage to combat addiction with data-driven insights.

Data cleaning involved checking for null values and duplicates, ensuring the dataset was ready for analysis.

Exploratory data analysis (EDA) revealed a negative correlation between average daily social media usage and hours of sleep per night.

Students were classified into low, medium, and high-risk categories for social media addiction based on their daily usage hours.

The project developed and applied a function to suggest personalized digital detox strategies for each risk level.

Visualizations, such as a pie chart, demonstrated the distribution of students across the different addiction risk levels.

A heatmap showed strong correlations between key variables like daily usage, sleep, and mental health scores.

The analysis indicated that younger students (ages 11-15) tend to have higher average daily usage hours compared to older age groups.

The strong negative correlations between social media engagement (usage and addiction) and both sleep and mental health suggest a significant public health concern for students.

The strong positive correlation between conflicts over social media and addiction scores indicates that addressing conflict resolution related to social media use could be a strategy in managing addiction.

Convert ipynb to HTML in Colab

```
#@title Convert ipynb to HTML in Colab
```