

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import os
import scipy
```

```
In [2]: os.getcwd()
```

```
Out[2]: 'C:\\\\Users\\HP PC'
```

```
In [3]: os.chdir("C:/Users/HP PC/Data Analyst")
```

```
In [4]: df = pd.read_csv("Students Social Media Addiction.csv")
```

```
In [40]: df
```

```
Out[40]:
```

	Student_ID	Age	Gender	Academic_Level	Country	Avg_Daily_Usage_Hours	M
0	1	19	Female	Undergraduate	Bangladesh	5.2	
1	2	22	Male	Graduate	India	2.1	
2	3	20	Female	Undergraduate	USA	6.0	
3	4	18	Male	High School	UK	3.0	
4	5	21	Male	Graduate	Canada	4.5	
...	
700	701	20	Female	Undergraduate	Italy	4.7	
701	702	23	Male	Graduate	Russia	6.8	
702	703	21	Female	Undergraduate	China	5.6	
703	704	24	Male	Graduate	Japan	4.3	
704	705	19	Female	Undergraduate	Poland	6.2	

705 rows × 13 columns



```
In [41]: df.isnull().sum()
```

```
Out[41]: 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
```

```
In [42]: df.info()
```

```
<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
```

```
In [43]: df.shape
```

```
Out[43]: (705, 13)
```

Exploratory Data Analysis (EDA)

```
In [44]: df.describe()
```

Out[44]:

	Student_ID	Age	Avg_Daily_Usage_Hours	Sleep_Hours_Per_Night	Mental_I
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	

Average_usage_gender hourly basis

```
In [45]: average_usage_gender = df.groupby("Gender")["Avg_Daily_Usage_Hours"].mean()
print(average_usage_gender)
```

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

Average time spent according to age

```
In [46]: average_usage_age_wise = df.groupby("Age")["Avg_Daily_Usage_Hours"].mean()
```

```
In [47]: average_usage_age_wise
```

```
Out[47]: Age
18    5.385714
19    5.120245
20    4.930303
21    4.950641
22    4.676190
23    4.508824
24    5.046154
Name: Avg_Daily_Usage_Hours, dtype: float64
```

Average time spent on each platform

```
In [48]: average_time_spent = df.groupby("Most_Used_Platform")["Avg_Daily_Usage_Hours"].m
print(average_time_spent)
```

```
Most_Used_Platform
WhatsApp      6.475926
TikTok        5.346104
Snapchat      5.092308
WeChat        4.960000
Instagram     4.872289
Twitter       4.870000
KakaoTalk     4.725000
Facebook      4.507317
VKontakte     4.250000
YouTube       4.080000
LINE          3.250000
LinkedIn      2.519048
Name: Avg_Daily_Usage_Hours, dtype: float64
```

Average Sleep_Hours_Per_Night according to age

```
In [49]: Average_Sleep = df.groupby("Age")["Sleep_Hours_Per_Night"].mean()
print(Average_Sleep)
```

```
Age
18    5.535714
19    6.742945
20    6.864848
21    6.868590
22    7.127891
23    6.938235
24    6.850000
Name: Sleep_Hours_Per_Night, dtype: float64
```

Addiction varies across demographics

```
In [50]: Demographic_Addiction = df.groupby("Country")["Addicted_Score"].mean().sort_values
print(Demographic_Addiction)
```

```
Country
Armenia      9.000
Ecuador      9.000
Czech Republic  9.000
Liechtenstein  9.000
Lebanon      9.000
USA          8.600
UAE          8.125
Costa Rica   8.000
Bhutan       8.000
Kuwait       8.000
Name: Addicted_Score, dtype: float64
```

Aggregation & Insights

Average addiction level across gender

```
In [51]: Addition_level_Gender = df.groupby("Gender")["Addicted_Score"].mean()  
print(Addiction_level_Gender)
```

```
Gender  
Female    6.515581  
Male      6.357955  
Name: Addicted_Score, dtype: float64
```

Average addiction level across Age group

```
In [52]: Addition_level_Age = df.groupby("Age")["Addicted_Score"].mean()  
print(Addiction_level_Age)
```

```
Age  
18    7.785714  
19    6.650307  
20    6.478788  
21    6.589744  
22    6.095238  
23    5.676471  
24    6.115385  
Name: Addicted_Score, dtype: float64
```

Average addiction level across Education level

```
In [53]: Addition_level_Education = df.groupby("Academic_Level")["Addicted_Score"].mean()  
print(Addiction_level_Education)
```

```
Academic_Level  
Graduate    6.243077  
High School  8.037037  
Undergraduate  6.492918  
Name: Addicted_Score, dtype: float64
```

Functions, Loops, and Conditionals

Custom function: Classify risk level (Low/Medium/High) based on usage hours

```
In [54]: def usage_hours(n):  
    if n >= 6:  
        return "High Risk"  
    if n >= 3 and n <= 5.9:  
        return "Medium Risk"  
    else:  
        return "Low Risk"
```

In [55]: usage_hours(7)

Out[55]: 'High Risk'

Suggest digital detox strategies using if-else blocks

```
In [57]: screen_time = int(input("Enter your daily screen time in hours: "))

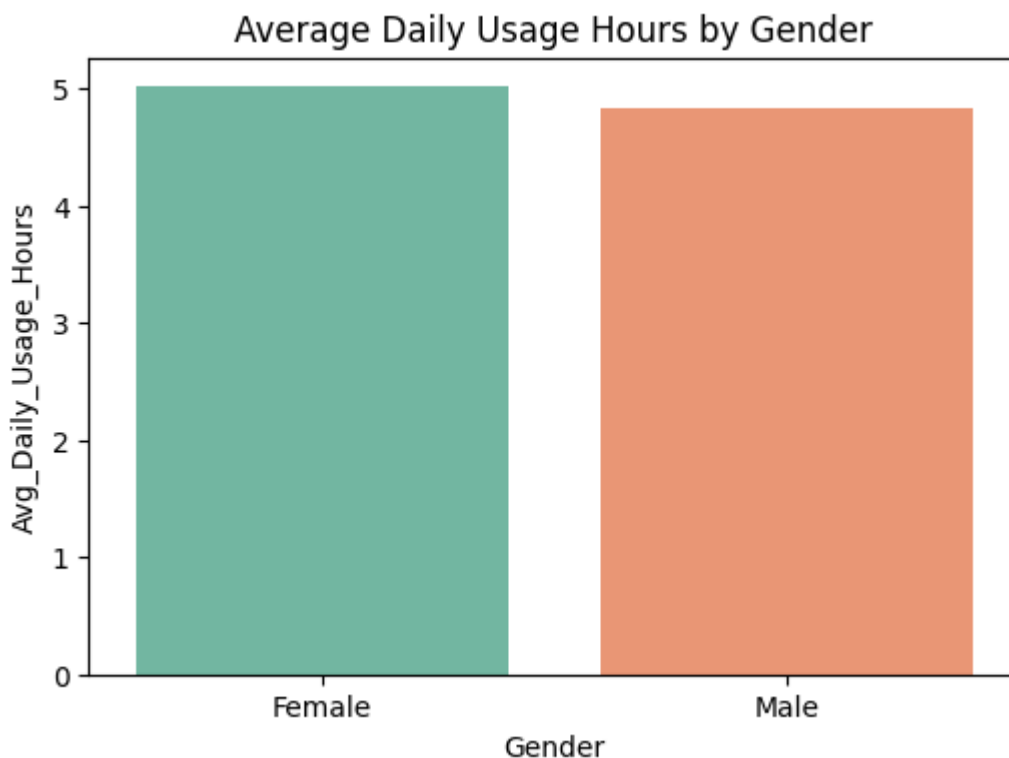
if screen_time > 8:
    print("⚠️ You are overusing screens. Try a full-day digital detox once a week.")
elif screen_time > 5:
    print("😊 Consider setting app usage limits and taking regular breaks.")
elif screen_time > 3:
    print("👍 Your usage is moderate. Try no-phone time before bed.")
else:
    print("✅ Great! Keep balancing your screen time with outdoor activities.")
```

😊 Consider setting app usage limits and taking regular breaks.

Data Visualization

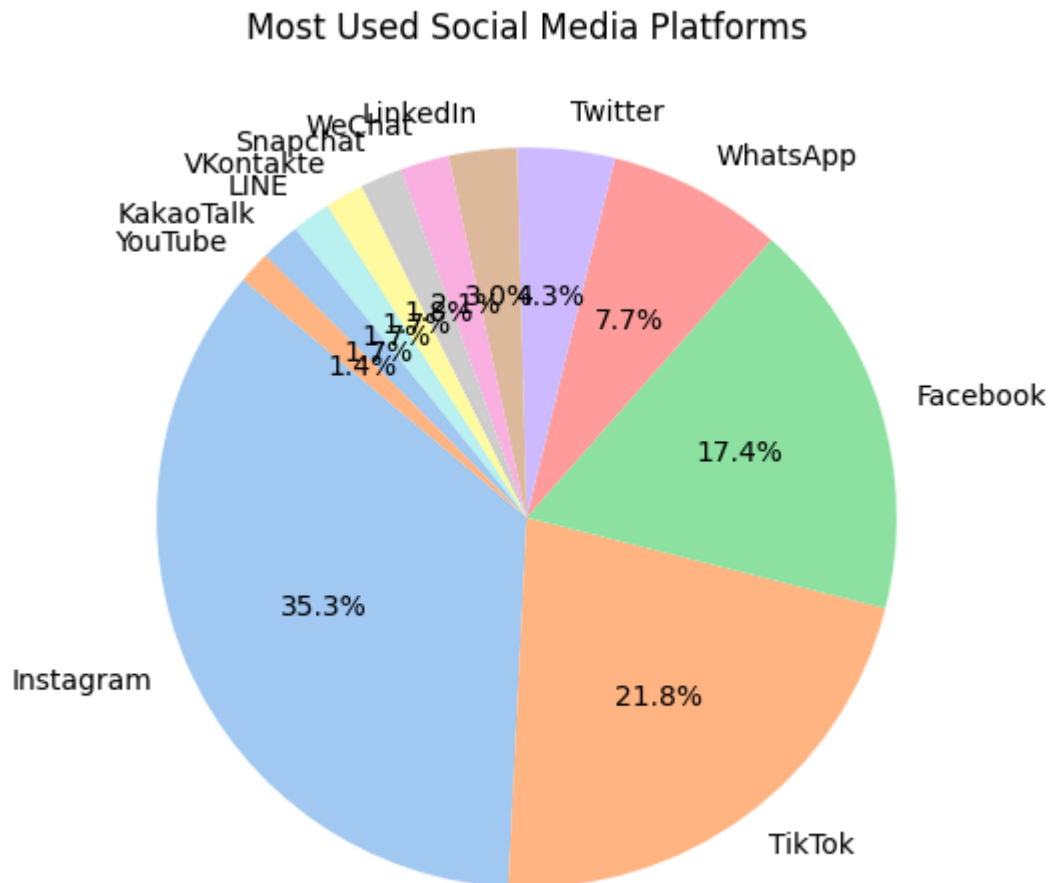
Bar chart

```
In [62]: plt.figure(figsize=(6,4))
sns.barplot(x="Gender", y="Avg_Daily_Usage_Hours", data=df, ci=None, palette="Set2")
plt.title("Average Daily Usage Hours by Gender")
plt.show()
```



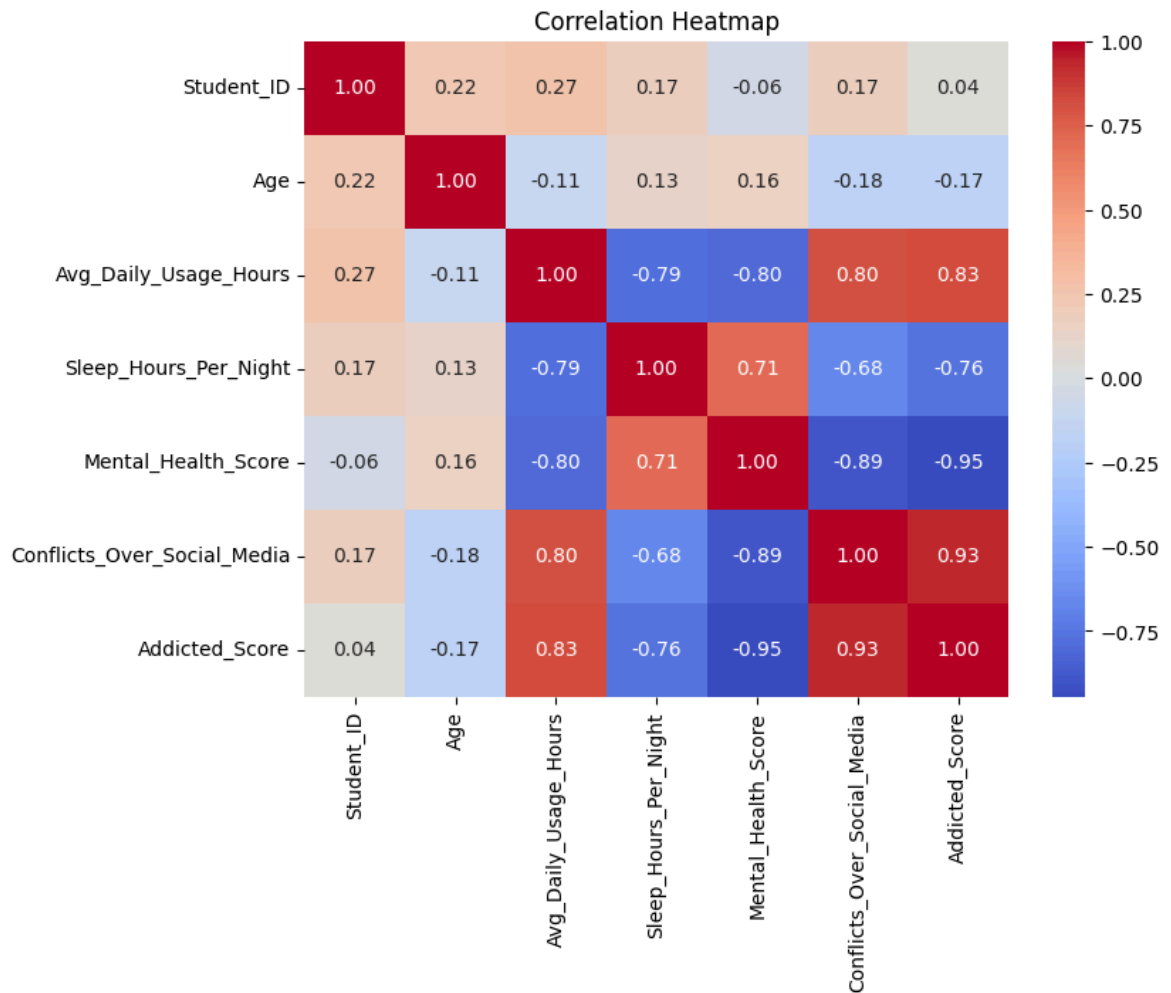
Pie Chart

```
In [63]: platform_counts = df["Most_Used_Platform"].value_counts()
plt.figure(figsize=(6,6))
plt.pie(platform_counts, labels=platform_counts.index, autopct="%1.1f%%",
        startangle=140, colors=sns.color_palette("pastel"))
plt.title("Most Used Social Media Platforms")
plt.show()
```



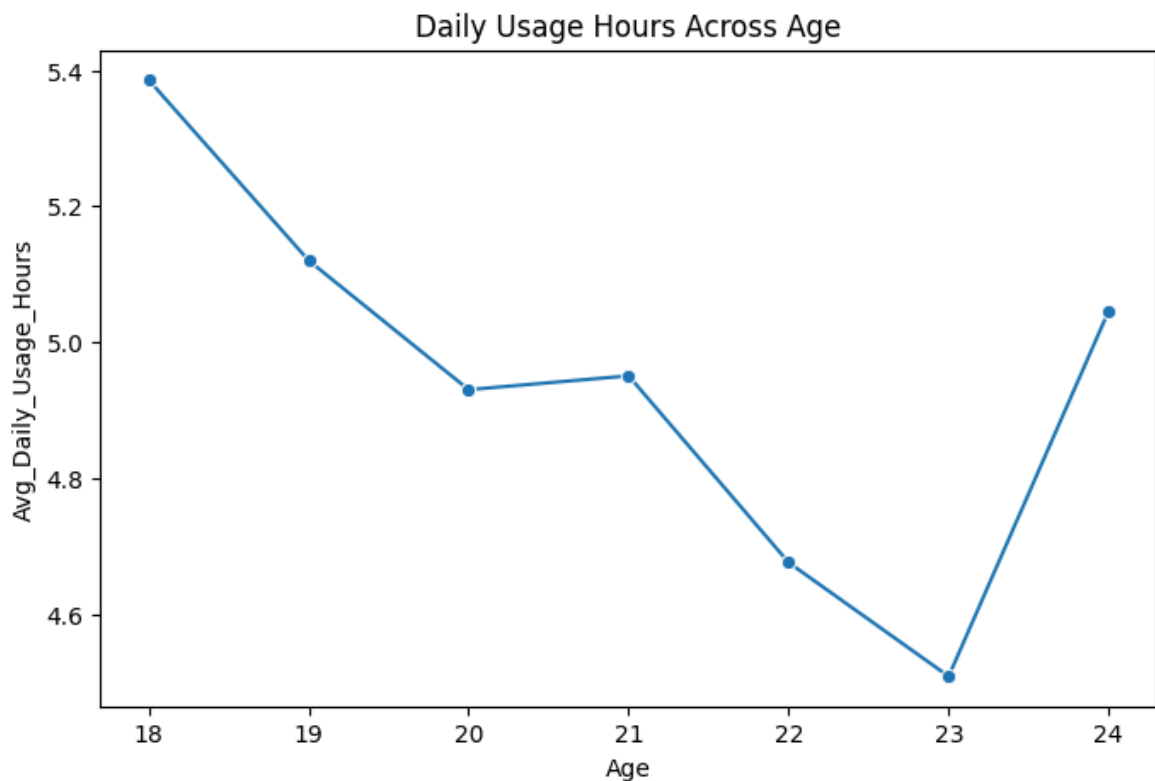
Heatmap

```
In [65]: plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



Line Chart

```
In [66]: plt.figure(figsize=(8,5))
sns.lineplot(x="Age", y="Avg_Daily_Usage_Hours", data=df, ci=None, marker="o")
plt.title("Daily Usage Hours Across Age")
plt.show()
```

**** Summary ****

1. Students today spend a significant part of their day on social media, with daily usage often exceeding healthy limits.
2. Analysis shows that younger students, especially in the 18–22 age group, are the heaviest users.
3. Gender trends reveal that both males and females are equally engaged, though platform preferences differ.
4. Popular platforms like Instagram and TikTok dominate, accounting for the majority of student screen time.
5. High daily usage directly correlates with reduced sleep hours, impacting mental health and focus.
6. Academic performance tends to decline among students reporting high addiction scores.
7. Social interactions outside the digital world also shrink with increasing online engagement.
8. Conflicts with family and friends rise, showing the broader impact of excessive usage.
9. Clear risk levels (Low, Medium, High) can be identified, enabling targeted digital detox strategies.
10. A balanced routine, awareness campaigns, and guided digital breaks can significantly reduce addiction risks.

In []:

In []:

In []:

In []:

In []:

In []: