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

1. Data Understanding & Cleaning

```
# Load the dataset

data=pd.read_csv("pythondata.csv")
data
```

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

705 rows × 13 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

JUST FOR CLARIFICATION THE ABOVE STEPS ARE DEFAULTLY SHOWN BY GOOGLE COLLAB HENCE ARE NOT GENERATED BY ANY AI TOOL

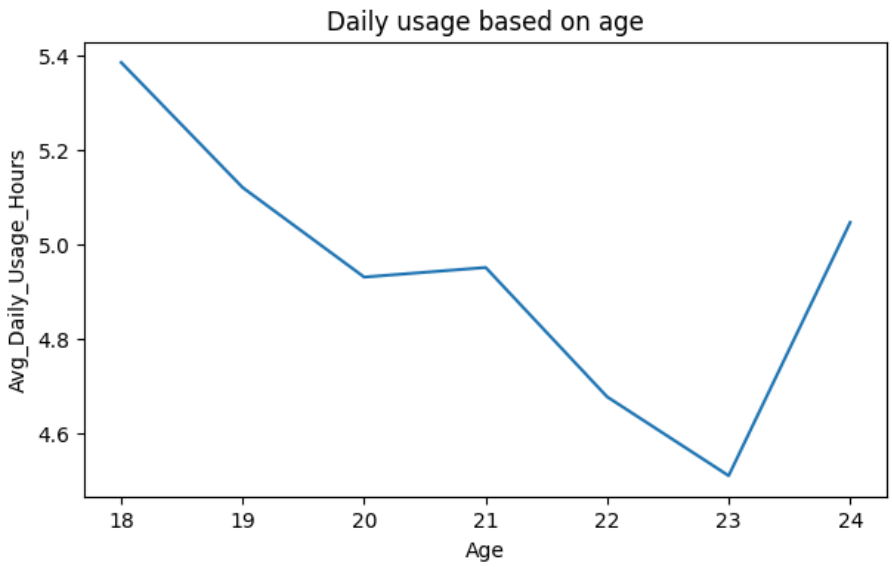
```
# Handle any missing values if any
data.info()
print("There are no missing values in the given data")
```

```
<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
There are no missing values in the given data
```

```
# Data type conversions werent required while working on this data
```

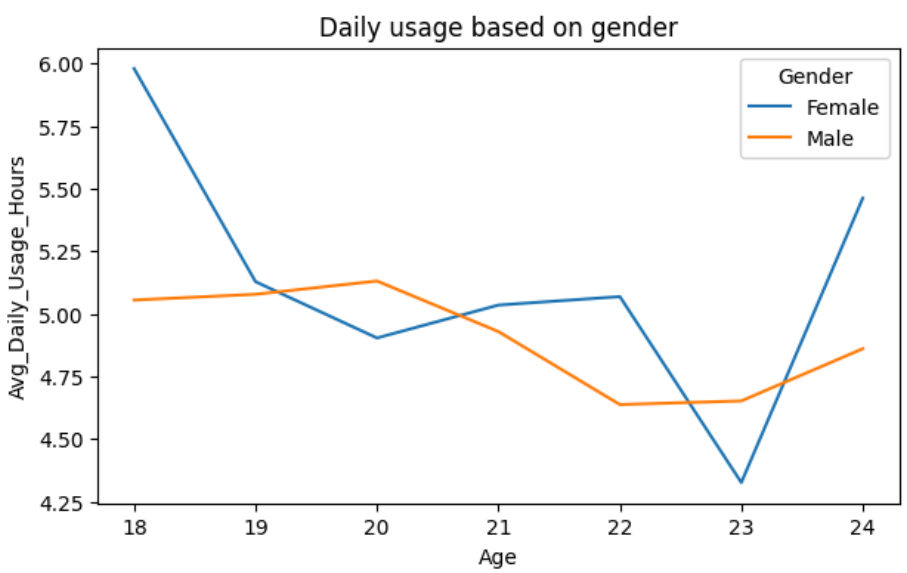
2. Exploratory Data Analysis (EDA)

```
# Understanding the relationship between various age groups, genders and their daily usage (in hours)
plt.figure(figsize=(7,4))
sns.lineplot(x="Age",y="Avg_Daily_Usage_Hours",data=data,errorbar=None)
plt.title("Daily usage based on age")
plt.show()
print("From the given line chart it is clearly shown that the 18 years old have the most average \
 daily usage hours")
print("23 years olds have the least average daily usage hours")
```




```
From the given line chart it is clearly shown that the 18 years old have the most average  daily usage hours
23 years olds have the least average daily usage hours
```

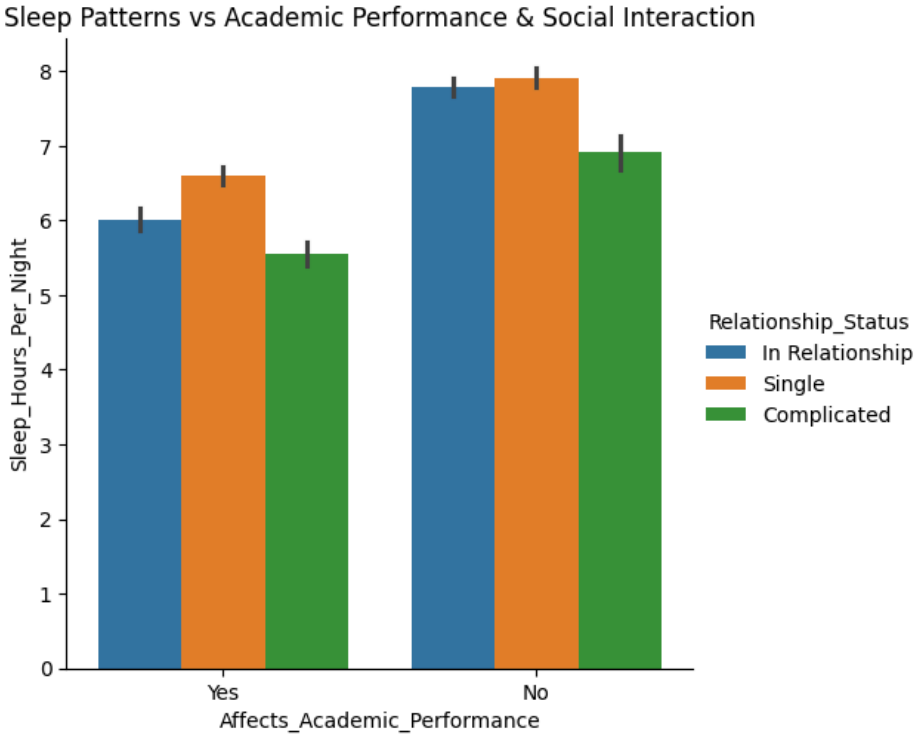
```
# Daily usage based on gender
plt.figure(figsize=(7,4))
sns.lineplot(x="Age",y="Avg_Daily_Usage_Hours",hue="Gender",data=data,errorbar=None)
plt.title("Daily usage based on gender")
plt.show()
print("From the above observations it is clear that BOYS have the most average daily usage than GIRLS")
```




```
From the above observations it is clear that BOYS have the most average daily usage than GIRLS
```

```
ing relationships between Sleep patterns, Academic performance, Social interaction
<="Affects_Academic_Performance",y="Sleep_Hours_Per_Night",hue="Relationship_Status",kind="bar",data=data)
leep Patterns vs Academic Performance & Social Interaction")
```

```
es whose sleep hours per night is above 7 shows their academic performance have no effect due to their normal schedule )
es who are getting lesser sleeps have have gotten effect on their academic performance ")
```




```
Peoples whose sleep hours per night is above 7 shows their academic performance have no effect due to their normal sle
Peoples who are getting lesser sleeps have have gotten effect on their academic performance
```

```
# Analyze how addiction varies across demographics
```

```
110
```

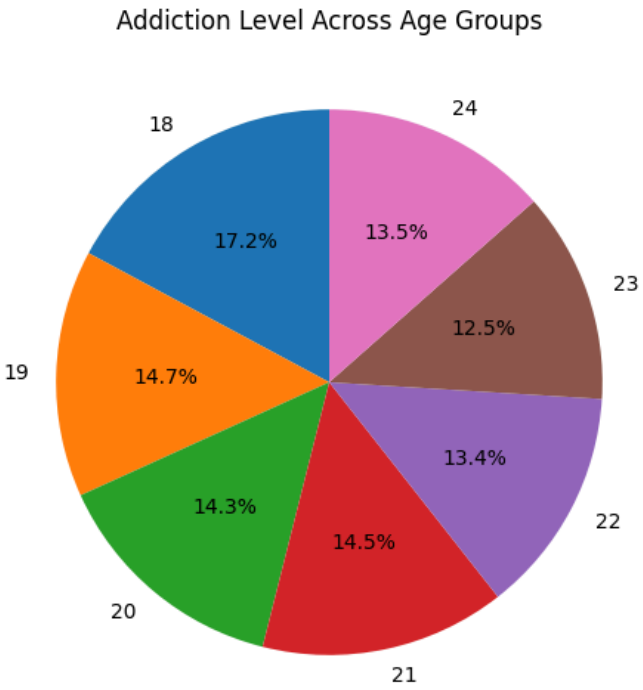
3. Aggregation & Insights

```
# Average additcion level acros diffrent gender
data.groupby("Gender")["Addicted_Score"].mean()
```

Gender	Addicted_Score
Female	6.515581
Male	6.357955

dtype: float64

```
# Average additcion level acros diffrent age groups
age_addiction = data.groupby('Age')['Addicted_Score'].mean()
plt.figure(figsize=(6,6))
plt.pie(age_addiction, labels=age_addiction.index, autopct='%1.1f%%', startangle=90)
plt.title("Addiction Level Across Age Groups")
plt.show()
print("18 years old have the highest average addiction score out of all age groups")
```




```
18 years old have the highest average addiction score out of all age groups
```

```
# Average addiction level acros diffrent education level (Academic level)
data.groupby("Academic_Level")["Addicted_Score"].mean()
```

Academic_Level	Addicted_Score
Graduate	6.243077
High School	8.037037
Undergraduate	6.492918

dtype: float64

4. Functions, Loops, and Conditionals

```
#Create custom functions to: Classify risk level (Low/Medium/High) based on usage hours
def classify_risk(usage_hours):
    if usage_hours <= 2:
        return "Low"
    elif usage_hours <= 5:
        return "Medium"
    else:
        return "High"
data["Risk_Level"] = data["Avg_Daily_Usage_Hours"].apply(classify_risk)
data[["Student_ID","Avg_Daily_Usage_Hours", "Risk_Level"]].head(5)
```

	Student_ID	Avg_Daily_Usage_Hours	Risk_Level
0	1	5.2	High
1	2	2.1	Medium
2	3	6.0	High
3	4	3.0	Medium
4	5	4.5	Medium

```
# Suggest digital detox strategies using if-else blocks
risk=input("Whats your risk level:").lower()
if risk=="low":
```

```
  print("Your daily usage is fine keep it up!")
elif risk=="medium":
  print("You have to lay off from your device somtimes!")
else:
  print("You better throw away that device and give your eyes some rest!!")
```

```
Whats your risk level:low
Your daily usage is fine keep it up!
```

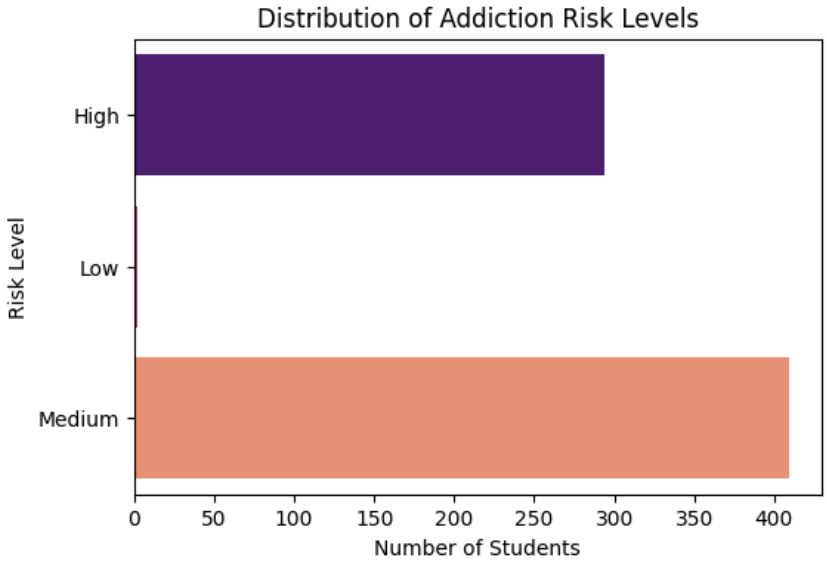
5. Data Visualization

```
# Here is one more visual chart becuase I have already used 3 above of diffrent kinds
risk_counts = data['Risk_Level'].value_counts().sort_index()
plt.figure(figsize=(6,4))
sns.barplot(x=risk_counts.values, y=risk_counts.index, palette="magma")
plt.xlabel("Number of Students")
plt.ylabel("Risk Level")
plt.title("Distribution of Addiction Risk Levels")
plt.show()
print("Highest medium level shows most people's average usage is between 2 hours to 5 hours")
```

```
/tmp/ipython-input-811417426.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hu

  sns.barplot(x=risk_counts.values, y=risk_counts.index, palette="magma")
```




```
Highest medium level shows most people's average usage is between 2 hours to 5 hours
```

6. Storytelling Deliverable

```
# 🔹 Key Patterns

#Higher daily usage hours strongly correlate with lower sleep and weaker academic performance.

#Addiction scores rise significantly when usage exceeds 5 hours/day.

#Relationship status and constant social interaction reduce average sleep hours.

#Country-wise differences highlight cultural variations in digital habits and screen engagement.

# 🔹 Root Causes

#Lack of self-regulation and poor time management fuel excessive screen use.

#Constant notifications and peer influence encourage prolonged engagement.

#Academic stress and emotional dependence on social media worsen addictive patterns.
```

```
# 🔹 Recommended Actions

#Implement structured digital detox routines (screen-free hours, app blockers).

#Promote awareness campaigns on healthy online behavior in schools and colleges.

#Encourage balanced routines with proper sleep, study schedules, and mindful social media use.


#Thank you for giving me this opportunity to work on this project.
#Although i am not a professional, I did my best.
#I am hoping for fair checking and evaluation from your end.


print("The End")
```

```
The End
```