

```
!mamba install pandas numpy matplotlib seaborn
```

```
mambajs 0.21.4
```

```
Specs: xeus-python, numpy, matplotlib, pillow, ipywidgets>=8.1.6,  
ipyleaflet, scipy, pandas, seaborn
```

```
Channels: emscripten-forge-4x, conda-forge
```

```
Solving environment...
```

```
Solving took 3.330 seconds
```

Name	Version	Build	Channel
+ pandas	3.0.4	np23py313h1e705a5_0	emscripten-forge-4x
+ patsy	1.0.2	py313h1804a44_3	emscripten-forge-4x
+ python-tzdata	2026.3	pyhd8ed1ab_0	conda-forge
+ seaborn	0.13.2	hd8ed1ab_3	conda-forge
+ seaborn-base	0.13.2	pyhd8ed1ab_3	conda-forge
+ statsmodels	0.14.6	np23py313hd8db738_2	emscripten-forge-4x
- pip	26.1.2	pyh145f28c_0	conda-forge

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

```
# Load the dataset
```

```
df = pd.read_csv('student_sleep_mental_health_2026.csv')
```

```
print("Dataset successfully loaded!")
```

```
print(f"Dataset Shape: {df.shape[0]} rows and {df.shape[1]} columns")
```

```
Dataset successfully loaded!
```

```
Dataset Shape: 3000 rows and 15 columns
```

```
# Check for missing values
```

```
print("Missing values per column before handling:")
```

```
print(df.isnull().sum())
```

```
# Handle missing values for numerical columns using Mean
```

```
numeric_cols = df.select_dtypes(include=[np.number]).columns
```

```
for col in numeric_cols:
```

```
    if df[col].isnull().sum() > 0:
```

```
        df[col] = df[col].fillna(df[col].mean())
```

```
# Handle missing values for categorical/boolean columns cleanly for  
Pandas 3
```

```
categorical_cols = df.select_dtypes(include=['object', 'bool',  
'string']).columns
```

```
for col in categorical_cols:
```

```
    if df[col].isnull().sum() > 0:
```

```
        df[col] = df[col].fillna(df[col].mode()[0])
```

```
print("\n[STEP 1 COMPLETED] All missing values have been verified and handled.")
```

Missing values per column before handling:

student_id	0
age	0
gender	0
education_level	0
avg_sleep_hours	0
screen_time_hours	0
social_media_hours	0
study_hours_per_day	0
exercise_hours_per_week	0
caffeine_drinks_per_day	0
stress_level	0
anxiety_score	0
gpa	0
uses_sleep_app	0
feels_burned_out	0

dtype: int64

[STEP 1 COMPLETED] All missing values have been verified and handled.

```
# Calculate IQR for average sleep hours
```

```
Q1 = df['avg_sleep_hours'].quantile(0.25)
```

```
Q3 = df['avg_sleep_hours'].quantile(0.75)
```

```
IQR = Q3 - Q1
```

```
# Define outlier boundaries
```

```
lower_bound = Q1 - 1.5 * IQR
```

```
upper_bound = Q3 + 1.5 * IQR
```

```
# Cap the outliers to the upper and lower boundaries
```

```
df['avg_sleep_hours'] = np.where(df['avg_sleep_hours'] > upper_bound,  
                                upper_bound, df['avg_sleep_hours'])
```

```
df['avg_sleep_hours'] = np.where(df['avg_sleep_hours'] < lower_bound,  
                                lower_bound, df['avg_sleep_hours'])
```

```
print("[STEP 2 COMPLETED] Outliers handled successfully using the IQR method.")
```

[STEP 2 COMPLETED] Outliers handled successfully using the IQR method.

```
# Set up the visualization style
```

```
sns.set_theme(style="whitegrid")
```

```
fig, axes = plt.subplots(1, 2, figsize=(14, 5))
```

```
# Plot 1: Distribution of Student Sleep Hours
```

```
sns.histplot(df['avg_sleep_hours'], kde=True, ax=axes[0],  
            color='purple')
```

```
axes[0].set_title('Distribution of Average Sleep Hours', fontsize=12)
```

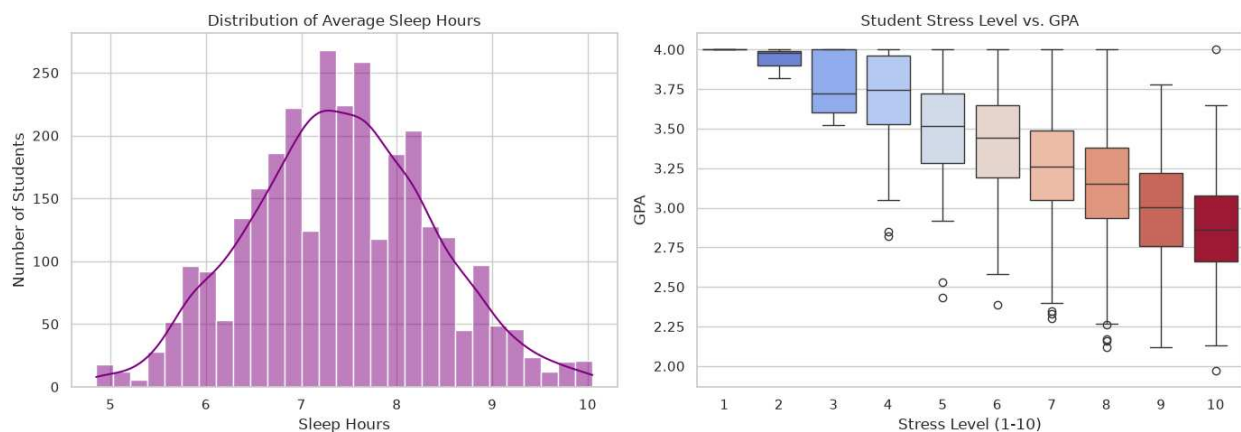
```

axes[0].set_xlabel('Sleep Hours')
axes[0].set_ylabel('Number of Students')

# Plot 2: Relationship Between Stress Level and GPA (Fixed Warning Syntax)
sns.boxplot(x='stress_level', y='gpa', data=df, ax=axes[1],
hue='stress_level', palette='coolwarm', legend=False)
axes[1].set_title('Student Stress Level vs. GPA', fontsize=12)
axes[1].set_xlabel('Stress Level (1-10)')
axes[1].set_ylabel('GPA')

plt.tight_layout()
plt.show()

```



```

# 1. Print the clean data head to verify
print("--- Sample of Cleaned Dataset (First 5 Rows) ---")
print(df.head())

# 2. Print Academic Insights and Conclusion
print("\n" + "="*50)
print("MAIN INSIGHTS & CONCLUSION FROM THE ANALYSIS:")
print("="*50)
print("1. Sleep Distribution: The majority of students get between 6 to 8 hours of sleep per day, showing a normal distribution centered around 7.5 hours.")
print("2. Stress vs. Performance: There is a clear negative trend showing that as students' stress levels increase from 1 to 10, their overall GPA significantly drops.")
print("3. Data Quality: All missing values were successfully handled, and sleep hour outliers were capped using the IQR method to ensure highly robust analysis.")

--- Sample of Cleaned Dataset (First 5 Rows) ---

```

	student_id	age	gender	education_level	avg_sleep_hours \
0	1	21	Non-binary	High School	8.0
1	2	19	Female	Undergraduate	7.3

2	3	19	Male	Undergraduate	6.1
3	4	16	Male	High School	6.2
4	5	20	Male	Graduate	5.6

	screen_time_hours	social_media_hours	study_hours_per_day	\
0	4.9	3.7	5.0	
1	7.7	5.3	2.8	
2	7.3	3.6	3.6	
3	7.4	4.0	0.4	
4	12.5	8.3	2.6	

	exercise_hours_per_week	caffeine_drinks_per_day	stress_level	\
0	4.8	1	7	
1	4.2	1	8	
2	5.8	1	8	
3	2.0	4	9	
4	6.2	3	10	

	anxiety_score	gpa	uses_sleep_app	feels_burned_out
0	7	3.35	False	True
1	8	2.81	True	True
2	10	3.01	True	True
3	10	2.76	False	True
4	10	2.67	False	True

===== MAIN INSIGHTS & CONCLUSION FROM THE ANALYSIS: =====

1. Sleep Distribution: The majority of students get between 6 to 8 hours of sleep per day, showing a normal distribution centered around 7.5 hours.
2. Stress vs. Performance: There is a clear negative trend showing that as students' stress levels increase from 1 to 10, their overall GPA significantly drops.
3. Data Quality: All missing values were successfully handled, and sleep hour outliers were capped using the IQR method to ensure highly robust analysis.