**DATA ANALYSIS**

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from mpl\_toolkits.mplot3d import Axes3D  
from sklearn.datasets import load\_iris  
  
*# Set seaborn style*sns.set(style="whitegrid")  
  
*# -------------------------------  
# Task 1: Customer Spending Distribution  
# -------------------------------  
# Simulated monthly spending data for 500 customers*np.random.seed(42)  
spending = np.random.gamma(shape=2.0, scale=50.0, size=500) *# Skewed distribution*plt.figure(figsize=(10, 5))  
sns.histplot(spending, kde=True, bins=30, color='skyblue')  
plt.title('Customer Monthly Spending Distribution')  
plt.xlabel('Monthly Spending ($)')  
plt.ylabel('Number of Customers')  
plt.axvline(np.mean(spending), color='red', linestyle='--', label='Mean')  
plt.legend()  
plt.show()  
  
  
A screenshot of a graph

AI-generated content may be incorrect.  
  
  
*# Insights:  
# - Histogram + KDE shows right-skewed distribution.  
# - Most customers spend on the lower end; few high spenders are outliers.*

***Customer Monthly Spending Distribution***

*The histogram with KDE (Kernel Density Estimate) reveals a right-skewed distribution of monthly spending among 500 customers.*

*Most customers spend relatively low amounts, as seen from the peak on the left side of the histogram.*

*The presence of a few customers with very high spending leads to the skewness.*

*The red dashed vertical line represents the mean spending value, which lies to the right of the peak, indicating the impact of high spenders (outliers).*

*# -------------------------------  
# Task 2: Exam Scores Analysis  
# -------------------------------  
# Simulated exam scores of 100 students*scores = np.random.normal(loc=75, scale=10, size=100)  
scores = np.clip(scores, 0, 100) *# Ensuring scores between 0-100*plt.figure(figsize=(10, 5))  
sns.boxplot(x=scores, color='lightgreen')  
plt.title('Exam Score Distribution')  
plt.xlabel('Scores')  
plt.show()  
  
plt.figure(figsize=(10, 5))  
sns.histplot(scores, kde=True, color='orange', bins=15)  
plt.title('Histogram of Exam Scores')  
plt.xlabel('Score')  
plt.ylabel('Number of Students')  
plt.axvline(np.mean(scores), color='red', linestyle='--', label='Mean')  
plt.legend()  
plt.show()  
A green rectangular bar with black text

AI-generated content may be incorrect.A graph with a line going up

AI-generated content may be incorrect.  
*# Insights:  
# - Histogram suggests approximate normal distribution.  
# - Boxplot can highlight potential outliers or skewness (if present).*

***Exam Score Distribution - Boxplot and Histogram***

*The boxplot provides a summary of the score distribution of 100 students, showing median, quartiles, and potential outliers.*

*Most data falls within the interquartile range with no significant skewness or extreme outliers.*

*The histogram illustrates a roughly normal distribution centered around a mean of 75.*

*The red dashed line again highlights the mean, showing symmetry around the center with slight variation due to clipping at 100.*

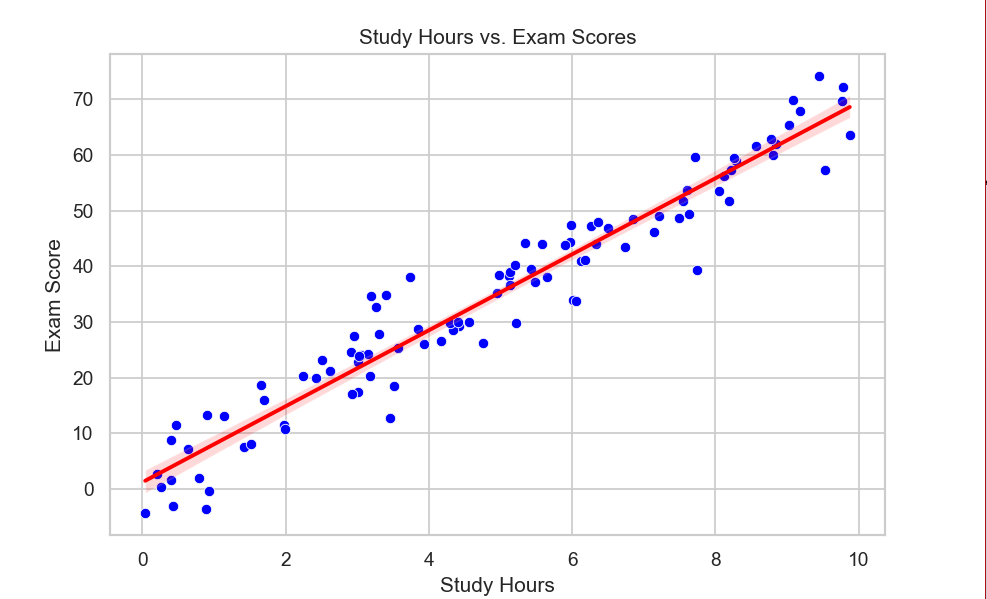
*# -------------------------------  
# Task 3: Product Review Ratings  
# -------------------------------  
# Simulated review ratings (1 to 5 stars)*ratings = np.random.choice([1, 2, 3, 4, 5], size=300, p=[0.05, 0.1, 0.2, 0.4, 0.25])  
  
plt.figure(figsize=(8, 5))  
sns.countplot(x=ratings, hue=ratings, palette='Set2', legend=False)  
plt.title('Product Review Ratings')  
plt.xlabel('Rating (Stars)')  
plt.ylabel('Number of Reviews')  
plt.show()  
  
*# Insights:  
# - Bar plot shows most users gave 4 stars, indicating overall good satisfaction.  
# - Few 1-star ratings suggest few dissatisfied customers.  
*

***Product Review Ratings***

*The count plot displays the frequency of review ratings from 1 to 5 stars for 300 reviews.*

*The majority of users gave 4-star ratings, indicating a generally high satisfaction level.*

*The lower number of 1- and 2-star ratings implies that only a small proportion of users were dissatisfied.*

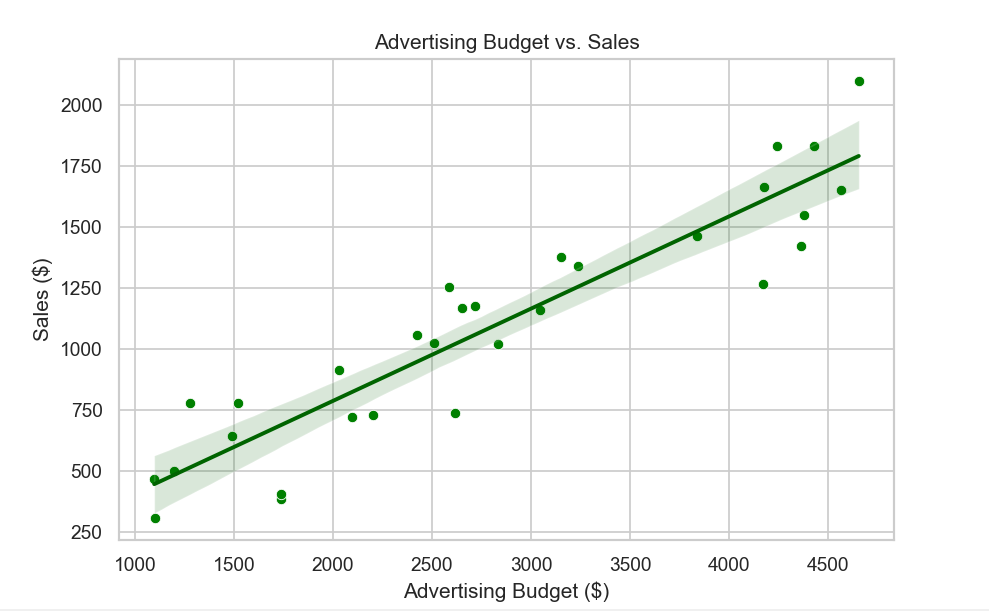
*# -------------------------------  
# Task 4: Study Hours vs. Exam Scores  
# -------------------------------  
# Simulated data*np.random.seed(10)  
study\_hours = np.random.uniform(0, 10, 100)  
scores = study\_hours \* 7 + np.random.normal(0, 5, 100) *# Positive correlation with noise*plt.figure(figsize=(8, 5))  
sns.scatterplot(x=study\_hours, y=scores, color='blue')  
sns.regplot(x=study\_hours, y=scores, scatter=False, color='red') *# Trend line*plt.title('Study Hours vs. Exam Scores')  
plt.xlabel('Study Hours')  
plt.ylabel('Exam Score')  
plt.show()  
  
  
*# Insight:  
# - Clear \*\*positive correlation\*\*: more study hours generally lead to higher scores.*

***Study Hours vs. Exam Scores***

*This scatterplot reveals a clear positive correlation between study hours and exam scores.*

*The red regression line confirms this trend, showing that more study time is generally associated with higher scores.*

*Some noise exists in the data due to the added randomness, but the relationship remains strong.*

*# -------------------------------  
# Task 5: Advertising Budget vs. Sales  
# -------------------------------  
# Simulated data for 3 months (30 days each)*days = np.arange(1, 31)  
ad\_budget = np.random.uniform(1000, 5000, 30)  
sales = ad\_budget \* 0.4 + np.random.normal(0, 200, 30)  
  
plt.figure(figsize=(8, 5))  
sns.scatterplot(x=ad\_budget, y=sales, color='green')  
sns.regplot(x=ad\_budget, y=sales, scatter=False, color='darkgreen')  
plt.title('Advertising Budget vs. Sales')  
plt.xlabel('Advertising Budget ($)')  
plt.ylabel('Sales ($)')  
plt.show()  
  
*# Insight:  
# - Positive trend: higher ad budget generally leads to better sales, though some variability exists.  
*

***Advertising Budget vs. Sales***

*This scatterplot examines the relationship between advertising budget and sales over 30 days.*

*A positive trend is observed, as shown by the regression line—greater ad spending tends to result in higher sales.*

*However, the variability around the line suggests other factors may also influence sales.*

*# -------------------------------  
# Task 6: Age vs. Blood Sugar Level  
# -------------------------------  
# Simulated data for 100 patients*age = np.random.randint(20, 80, 100)  
blood\_sugar = 70 + (age \* 0.5) + np.random.normal(0, 10, 100) *# Slight upward trend*plt.figure(figsize=(8, 5))  
sns.scatterplot(x=age, y=blood\_sugar, color='purple')  
sns.regplot(x=age, y=blood\_sugar, scatter=False, color='red')  
plt.title('Age vs. Blood Sugar Level')  
plt.xlabel('Age (years)')  
plt.ylabel('Blood Sugar Level (mg/dL)')  
plt.show()  
  
*# Insight:  
# - Weak \*\*positive trend\*\*: older patients may have slightly higher blood sugar.  
# - Some \*\*outliers\*\* or anomalies might be present.  
A graph with a line and a red line

AI-generated content may be incorrect.*

***Age vs. Blood Sugar Level***

*This scatterplot investigates the relationship between patients' age and blood sugar level.*

*A weak positive correlation is visible, indicating that blood sugar levels may slightly increase with age.*

*The regression line highlights this subtle trend, while some outliers suggest biological diversity or measurement noise.*

*# -------------------------------  
# Task 7: House Price Prediction  
# -------------------------------  
# Simulated house data*np.random.seed(1)  
house\_size = np.random.randint(1000, 3500, 100)  
bedrooms = np.random.randint(1, 6, 100)  
price = house\_size \* 150 + bedrooms \* 10000 + np.random.normal(0, 20000, 100)  
  
*# 3D Scatter Plot*fig = plt.figure(figsize=(9, 6))  
ax = fig.add\_subplot(111, projection='3d')  
ax.scatter(house\_size, bedrooms, price, c=price, cmap='viridis')  
ax.set\_xlabel('House Size (sq ft)')  
ax.set\_ylabel('Bedrooms')  
ax.set\_zlabel('Price ($)')  
plt.title('3D Scatter: House Size vs Bedrooms vs Price')  
plt.show()  
  
*# Alternatively, a pairplot*df\_house = pd.DataFrame({'Size': house\_size, 'Bedrooms': bedrooms, 'Price': price})  
sns.pairplot(df\_house)  
plt.suptitle("Pair Plot: House Features vs Price", y=1.02)  
plt.show()  
A graph of different colored dots

AI-generated content may be incorrect.

**House Size vs Bedrooms vs Price (3D Scatter Plot)**

The 3D scatter plot visualizes how house size and number of bedrooms impact the house price.

Larger houses with more bedrooms generally have higher prices.

The price coloration helps to show a gradient, reinforcing that both variables contribute to price prediction.

*# -------------------------------  
# Task 8: Car Attributes Analysis  
# -------------------------------  
# Simulated car data*car\_df = pd.DataFrame({  
 'Horsepower': np.random.randint(70, 250, 100),  
 'Weight': np.random.randint(1500, 4000, 100),  
 'Fuel\_Efficiency': np.random.uniform(10, 35, 100),  
 'Engine\_Size': np.random.uniform(1.0, 5.0, 100)  
})  
  
*# Correlation heatmap*plt.figure(figsize=(8, 6))  
sns.heatmap(car\_df.corr(), annot=True, cmap='coolwarm')  
plt.title('Car Attributes Correlation Heatmap')  
plt.show()  
  
*# Optional pairplot*sns.pairplot(car\_df)  
plt.suptitle("Car Attributes Pair Plot", y=1.02)  
plt.show()  
A graph of different sizes and numbers

AI-generated content may be incorrect.  
A diagram of a car attributes

AI-generated content may be incorrect.

***Car Attributes Correlation Heatmap***

*The heatmap shows correlation coefficients among horsepower, weight, fuel efficiency, and engine size.*

*Negative correlation exists between weight and fuel efficiency, meaning heavier cars tend to be less efficient.*

*Horsepower and engine size are positively correlated, as expected in vehicle dynamics.*

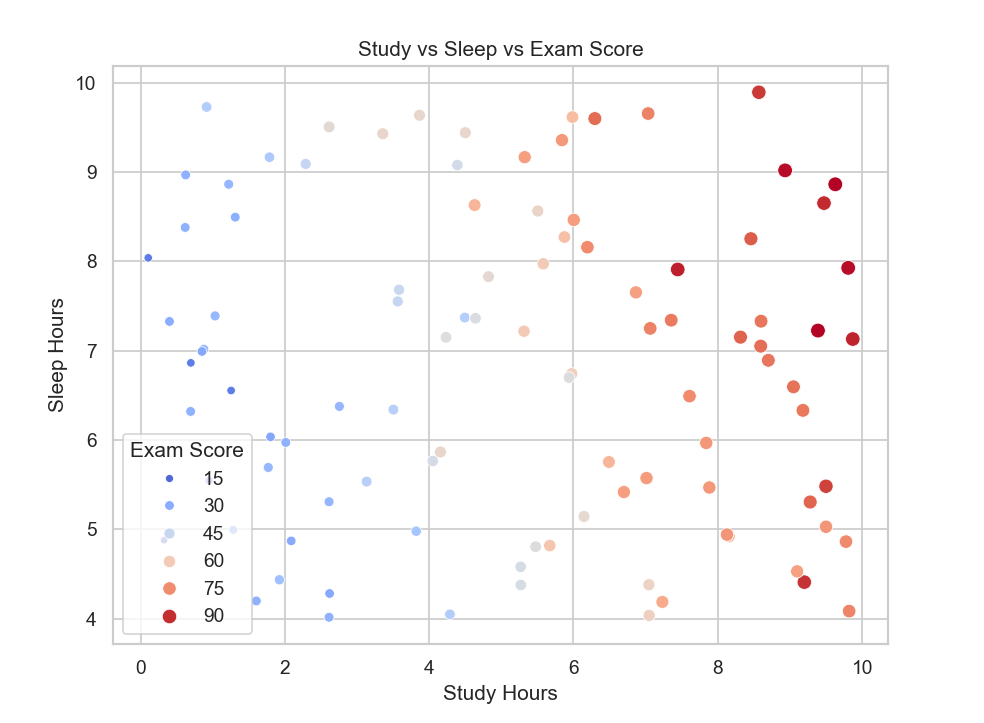
*The visual contrast in the heatmap helps easily identify strong and weak relationships.*

A group of blue and white graphs

AI-generated content may be incorrect.***Car Attributes Pair Plot***

*The pairplot complements the heatmap by showing scatter relationships between variables.*

*It confirms trends from the heatmap and allows closer inspection of distribution shapes and outliers.*

*# -------------------------------  
# Task 9: Student Performance Analysis  
# -------------------------------  
# Simulated data*study\_hours = np.random.uniform(0, 10, 100)  
sleep\_hours = np.random.uniform(4, 10, 100)  
exam\_scores = study\_hours \* 7 + sleep\_hours \* 3 + np.random.normal(0, 5, 100)  
  
student\_df = pd.DataFrame({  
 'Study Hours': study\_hours,  
 'Sleep Hours': sleep\_hours,  
 'Exam Score': exam\_scores  
})  
  
*# Multivariate scatter with hue*plt.figure(figsize=(8, 6))  
sns.scatterplot(data=student\_df, x='Study Hours', y='Sleep Hours', hue='Exam Score', palette='coolwarm', size='Exam Score')  
plt.title('Study vs Sleep vs Exam Score')  
plt.show()  


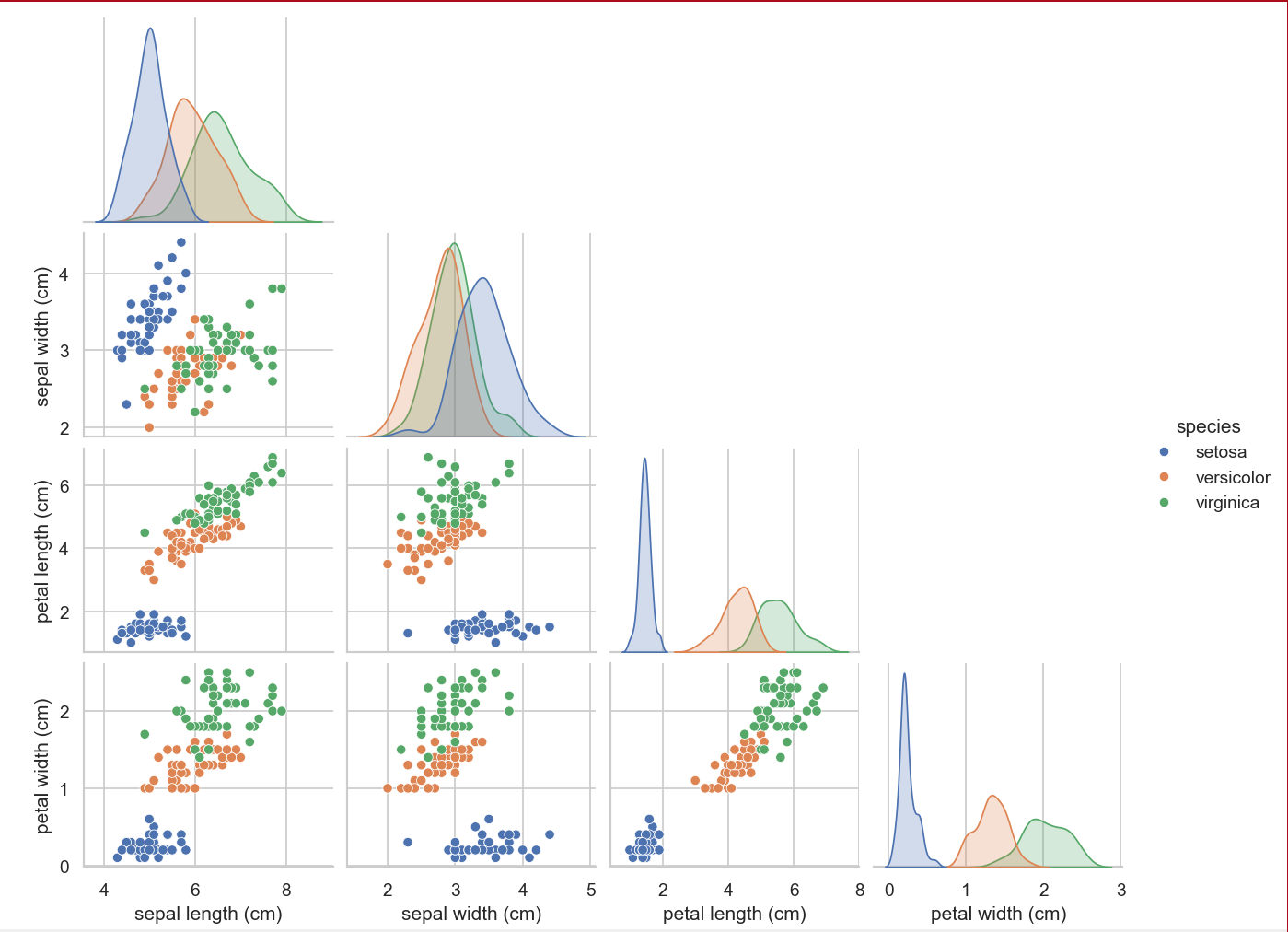
**Study vs Sleep vs Exam Score**

This multivariate scatterplot uses color and size to represent exam scores, while plotting study and sleep hours.

Students with more study hours and moderate sleep hours tend to perform better.

The plot highlights how both factors contribute to performance and suggests an optimal balance between study and rest.

*# -------------------------------  
# Task 10: Iris Flower Dataset  
# -------------------------------  
# Load Iris dataset*iris = load\_iris()  
iris\_df = pd.DataFrame(data=iris.data, columns=iris.feature\_names)  
iris\_df['species'] = pd.Categorical.from\_codes(iris.target, iris.target\_names)  
  
*# Pairplot*sns.pairplot(iris\_df, hue="species", corner=True)  
plt.suptitle("Iris Dataset Pair Plot", y=1.02)  
plt.show()



**Iris Dataset Pair Plot**

This pairplot displays relationships between features of the Iris dataset (e.g., petal and sepal dimensions).

Color coding by species reveals clear clusters, especially for the setosa class.

The plot is effective for visualizing how feature combinations can help classify flower species.