1. **Question**: How would you use NumPy arrays to calculate the average score for each subject and determine the subject with the highest average score? Assume 4x4 matrix that stores marks of each student in given order.

**Program:-**

import numpy as np

student\_scores = np.array([

[78, 89, 92, 85],

[92, 84, 79, 88],

[87, 76, 90, 94],

[90, 85, 88, 92]

])

subject\_averages = np.mean(student\_scores, axis=0)

highest\_average\_subject\_index = np.argmax(subject\_averages)

subject\_names = ['Math', 'Science', 'English', 'History']

highest\_average\_subject = subject\_names[highest\_average\_subject\_index]

print("Average scores for each subject:", subject\_averages)

print("Subject with the highest average score:", highest\_average\_subject)

**Output:-**

Average scores for each subject: [86.75 83.5 87.25 89.75]

Subject with the highest average score: History

**2.Question:** How would you find the average price of all the products sold in the past month? Assume 3x3 matrix with each row representing the sales for a different product.

**Program:-**

import numpy as np

sales\_data = np.array([[150, 200, 100],

[40, 110, 90],

[70, 500, 80]])

average\_price = np.mean(sales\_data)

print("Average Price of All Products Sold in the Past Month:", average\_price)

**Output:-**

Average Price of All Products Sold in the Past Month: 148.88888888888889

**3.Question**: Using NumPy arrays and operations, how would you find the average sale price of houses with more than four bedrooms in the neighborhood?

**Program:**

import numpy as np

house\_data = np.array([

[5, 200, 2500],

[4, 180, 2200],

[6, 220, 2800],

[3, 160, 1800],

[5, 210, 2600]

])

bedroom\_condition = house\_data[:, 0] > 4

houses\_more\_than\_four\_bedrooms = house\_data[bedroom\_condition]

average\_sale\_price = np.mean(houses\_more\_than\_four\_bedrooms[:, 2])

print("Average Sale Price of Houses with More than Four Bedrooms:", average\_sale\_price)

**Output:**

Average Sale Price of Houses with More than Four Bedrooms: 2633.3333333333335

4.**Question:** Using NumPy arrays and arithmetic operations calculate the total sales for the year and determine the percentage increase in sales from the first quarter to the fourth quarter?

**Program:**

import numpy as np

sales\_data = np.array([260000, 290000, 320000, 350000])

total\_sales = np.sum(sales\_data)

first\_quarter\_sales = sales\_data[0]

fourth\_quarter\_sales = sales\_data[-1]

percentage\_increase = ((fourth\_quarter\_sales - first\_quarter\_sales) / first\_quarter\_sales) \* 100

print("Total Sales for the Year:", total\_sales)

print("Percentage Increase from Q1 to Q4:", percentage\_increase, "%")

**Output:**

Total Sales for the Year: 1220000

Percentage Increase from Q1 to Q4: 34.61538461538461 %

1. **Question:** How would you use NumPy arrays and arithmetic operations to calculate the average fuel efficiency and determine the percentage improvement in fuel efficiency between two car models?

**Program:**

import numpy as np

fuel\_efficiency = np.array([30.5, 28.2, 35.8, 32.0, 30.5, 29.8])

average\_fuel\_efficiency = np.mean(fuel\_efficiency)

model\_2\_efficiency = fuel\_efficiency[1]

model\_5\_efficiency = fuel\_efficiency[4]

percentage\_improvement = ((model\_5\_efficiency - model\_2\_efficiency) / model\_2\_efficiency) \* 100

print("Average Fuel Efficiency:", average\_fuel\_efficiency, "MPG")

print("Percentage Improvement between Model 2 and Model 5:", percentage\_improvement, "%")

**Output:**

Average Fuel Efficiency: 31.133333333333336 MPG

Percentage Improvement between Model 2 and Model 5: 8.156028368794328 %

1. **Question:** Use arithmetic operations to calculate the total cost of a customer's purchase, including discounts and taxes, given the item prices, quantities, discount rate, and tax rate?

**Program:**

item\_prices = [2.5, 1.0, 3.0, 4.5]

item\_quantities = [3, 2, 1, 4]

discount\_rate = 10

tax\_rate = 7.5

total\_cost\_before\_discount = sum([price \* quantity for price, quantity in zip(item\_prices, item\_quantities)])

discount = (discount\_rate / 100) \* total\_cost\_before\_discount

total\_cost\_after\_discount = total\_cost\_before\_discount - discount

tax\_amount = (tax\_rate / 100) \* total\_cost\_after\_discount

final\_total\_cost = total\_cost\_after\_discount + tax\_amount

print("Total Cost Before Discount: $", total\_cost\_before\_discount)

print("Discount Amount: $", discount)

print("Total Cost After Discount: $", total\_cost\_after\_discount)

print("Tax Amount: $", tax\_amount)

print("Final Total Cost: $", final\_total\_cost)

**Output:**

Total Cost Before Discount: $ 30.5

Discount Amount: $ 3.0500000000000003

Total Cost After Discount: $ 27.45

Tax Amount: $ 2.05875

Final Total Cost: $ 29.50875

1. **Question:**Using Pandas DataFrame operations, how would you find the following information from the order\_data DataFrame:
2. The total number of orders made by each customer.
3. The average order quantity for each product.
4. The earliest and latest order dates in the dataset.

**Program:**

import pandas as pd

data = {

'Customer ID': [101, 102],

'Order Date': ['2023-08-01', '2023-08-02'],

'Product Name': ['Product A', 'Product B'],

'Order Quantity': [2, 1]

}

order\_data = pd.DataFrame(data)

customer\_order\_counts = order\_data['Customer ID'].value\_counts()

print("Total number of orders made by each customer:\n", customer\_order\_counts)

average\_order\_quantity\_per\_product = order\_data.groupby('Product Name')['Order Quantity'].mean()

print("Average order quantity for each product:\n", average\_order\_quantity\_per\_product)

earliest\_order\_date = order\_data['Order Date'].min()

latest\_order\_date = order\_data['Order Date'].max()

print("Earliest Order Date:", earliest\_order\_date)

print("Latest Order Date:", latest\_order\_date)

**Output:**

Total number of orders made by each customer:

Customer ID

101 1

102 1

Name: count, dtype: int64

Average order quantity for each product:

Product Name

Product A 2.0

Product B 1.0

Name: Order Quantity, dtype: float64

Earliest Order Date: 2023-08-01

Latest Order Date: 2023-08-02

1. **Question:** How would you find the top 5 products that have been sold the most in the past month?

**Program:**

import pandas as pd

data = {

'Product Name': ['Product A', 'Product B', 'Product A', 'Product C'],

'Quantity Sold': [10, 5, 8, 12]

}

sales\_data = pd.DataFrame(data)

product\_sales = sales\_data.groupby('Product Name')['Quantity Sold'].sum()

top\_2\_products = product\_sales.sort\_values(ascending=False).head(2)

print("Top 2 products sold the most in the past month:")

print(top\_2\_products)

**Output:**

Top 2 products sold the most in the past month:

Product Name

Product A 18

Product C 12

Name: Quantity Sold, dtype: int64

**9.Question:** Using Pandas DataFrame operations, how would you find the following information from the property\_data DataFrame:

1. The average listing price of properties in each location.
2. The number of properties with more than four bedrooms.
3. The property with the largest area.

**Program:**

import pandas as pd

data = {

'property\_id': [1, 2, 3, 4, 5],

'location': ['City A', 'City B', 'City A', 'City C', 'City B'],

'number\_of\_bedrooms': [3, 2, 5, 3, 5],

'area\_sqft': [1500, 1200, 1800, 1600, 2200],

'listing\_price': [250000, 180000, 320000, 275000, 420000]

}

property\_data = pd.DataFrame(data)

average\_price\_by\_location = property\_data.groupby('location')['listing\_price'].mean()

properties\_with\_more\_than\_four\_bedrooms = len(property\_data[property\_data['number\_of\_bedrooms'] > 4])

property\_with\_largest\_area = property\_data[property\_data['area\_sqft'] == property\_data['area\_sqft'].max()]

print("Average Listing Price by Location:")

print(average\_price\_by\_location)

print("\nNumber of Properties with More than Four Bedrooms:", properties\_with\_more\_than\_four\_bedrooms)

print("\nProperty with the Largest Area:")

print(property\_with\_largest\_area)

**Output:**

Average Listing Price by Location:

location

City A 285000.0

City B 300000.0

City C 275000.0

Name: listing\_price, dtype: float64

Number of Properties with More than Four Bedrooms: 2

Property with the Largest Area:

property\_id location number\_of\_bedrooms area\_sqft listing\_price

4 5 City B 5 2200 420000

**10.Question**: 1. How would you develop a Python program to create a line plot of the monthly sales data?

2: How would you develop a Python program to create a bar plot of the monthly sales data?

**Program:**

import matplotlib.pyplot as plt

months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]

sales = [10000, 12000, 11000, 13000, 14000, 16000, 17000, 15000, 13000, 12000, 11000, 10000]

plt.figure(figsize=(10, 6))

plt.plot(months, sales, marker='o', linestyle='-', color='b', label='Monthly Sales')

plt.title('Monthly Sales Data')

plt.xlabel('Month')

plt.ylabel('Sales')

plt.legend()

plt.grid(True)

plt.show()

plt.figure(figsize=(10, 6))

plt.bar(months, sales, color='b', label='Monthly Sales')

plt.title('Monthly Sales Data')

plt.xlabel('Month')

plt.ylabel('Sales')

plt.legend()

plt.grid(axis='y')

plt.show()

**Output:**

