1. You are working on a project that involves analyzing student performance data for a class of 32 students. The data is stored in a NumPy array named student_scores, where each row represents a student and each column represents a different subject. The subjects are arranged in the following order: Math, Science, English, and History. Your task is to calculate the average score for each subject and identify the subject with the highest average score.
   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.

```python
import numpy as np
student_scores = np.array([
    [85, 78, 92, 88],
    [76, 85, 83, 91],
    [90, 88, 79, 85],
    [70, 75, 88, 80]
])
subject_averages = np.mean(student_scores, axis=0)
subjects = ["Math", "Science", "English", "History"]
max_avg_index = np.argmax(subject_averages)
highest_subject = subjects[max_avg_index]
highest_avg_score = subject_averages[max_avg_index]
print("Average scores for each subject:")
for i in range(len(subjects)):
    print(f"{subjects[i]}: {subject_averages[i]:.2f}")
print(f"\nSubject with the highest average score: {highest_subject}
    ({highest_avg_score:.2f})")
```

```
Average scores for each subject:
Math: 80.25
Science: 81.50
English: 85.50
History: 86.00

Subject with the highest average score: History (86.00)

=== Code Execution Successful ===
```

2. Scenario: You are a data analyst working for a company that sells products online. You have been tasked with analyzing the sales data for the past month. The data is stored in a NumPy array.
   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

```python
import numpy as np

sales_data = np.array([
    [100, 150, 200],
    [80, 120, 160],
    [90, 110, 130]
])

all_prices = sales_data.flatten()

average_price = np.mean(all_prices)

print("All product prices from sales:")
print(all_prices)

print(f"\nAverage price of all products sold: {average_price:.2f}")
```

```
All product prices from sales:
[100 150 200  80 120 160  90 110 130]

Average price of all products sold: 126.67

=== Code Execution Successful ===
```

3. Scenario: You are working on a project that involves analyzing a dataset containing information about houses in a neighborhood. The dataset is stored in a CSV file, and you have imported it into a NumPy array named house_data. Each row of the array represents a house, and the columns contain various features such as the number of bedrooms, square footage, and sale price.
   Question: Using NumPy arrays and operations, how would you find the average sale price of houses with more than four bedrooms in the neighborhood?

```python
import numpy as np

house_data = np.array([
    [3, 1800, 250000],
    [5, 2400, 450000],
    [6, 3000, 520000],
    [4, 2000, 300000],
    [7, 3500, 600000],
    [2, 1500, 200000],
    [5, 2800, 480000],
    [6, 3200, 550000]
])

houses_with_more_than_4_bedrooms = house_data[house_data[:, 0] > 4]

sale_prices = houses_with_more_than_4_bedrooms[:, 2]

average_sale_price = np.mean(sale_prices)

print("Average sale price of houses with more than 4 bedrooms:", round
    (average_sale_price, 2))

```

```
Average sale price of houses with more than 4 bedrooms: 520000.0

=== Code Execution Successful ===
```

4. Scenario: You are working on a project that involves analyzing the sales performance of a company over the past four quarters. The quarterly sales data is stored in a NumPy array named sales_data, where each element represents the sales amount for a specific quarter. Your task is to calculate the total sales for the year and determine the percentage increase in sales from the first quarter to the fourth quarter.
   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?

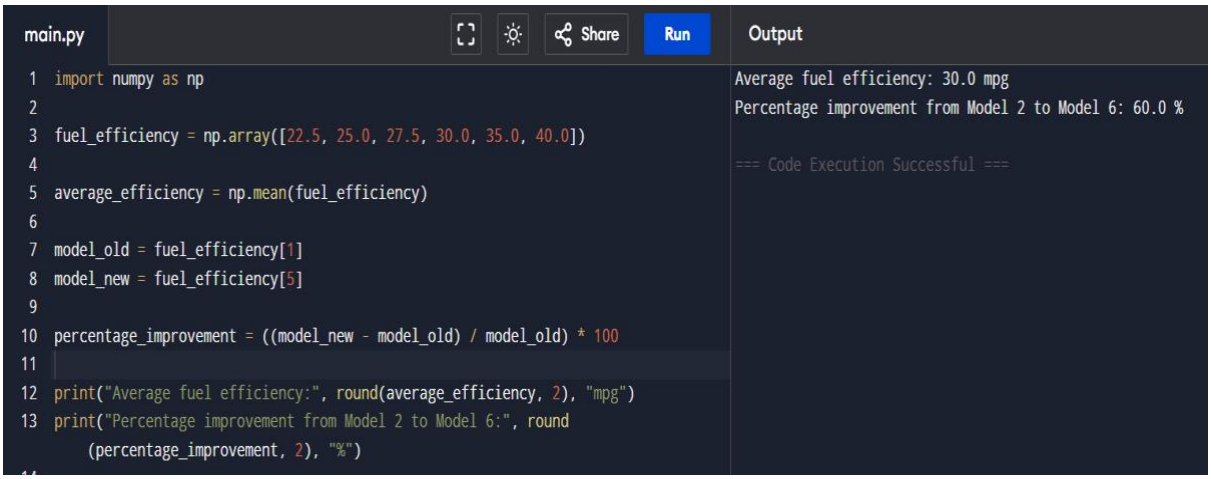
main.py | Share | Run | Output

```python
import numpy as np

sales_data = np.array([100000, 125000, 140000, 180000])

total_sales = np.sum(sales_data)

percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100

print("Total sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4:", round(percentage_increase, 2), "%")

```

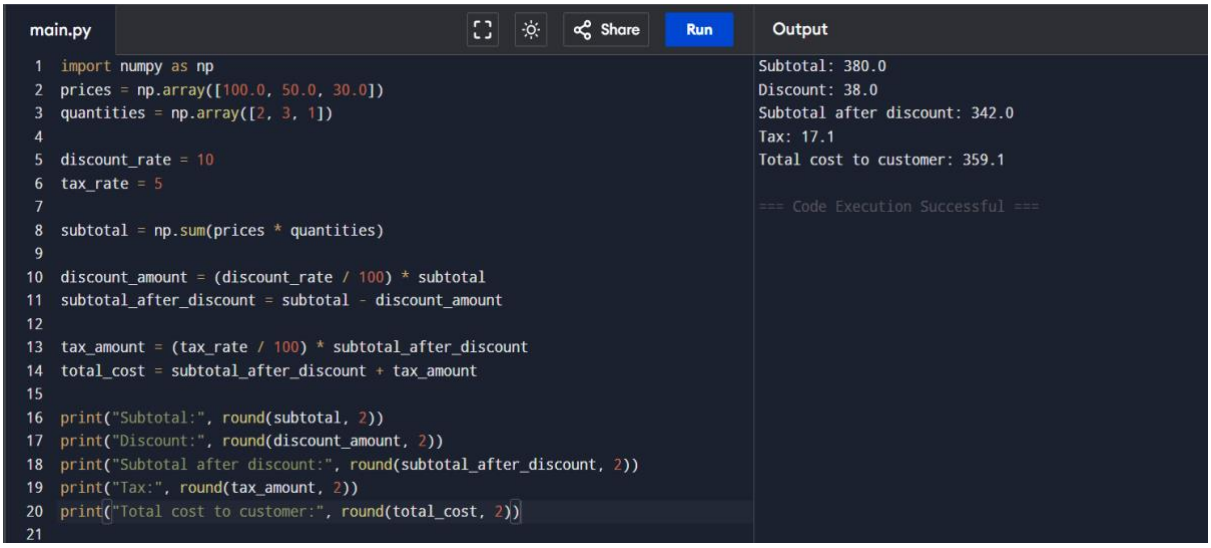
```
Total sales for the year: 545000
Percentage increase from Q1 to Q4: 80.0 %

=== Code Execution Successful ===
```

5. Scenario: You are a data analyst working for a car manufacturing company. As part of your analysis, you have a dataset containing information about the fuel efficiency of different car models. The dataset is stored in a NumPy array named fuel_efficiency, where each element represents the fuel efficiency (in miles per gallon) of a specific car model. Your task is to calculate the average fuel efficiency and determine the percentage improvement in fuel efficiency between two car models.

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?

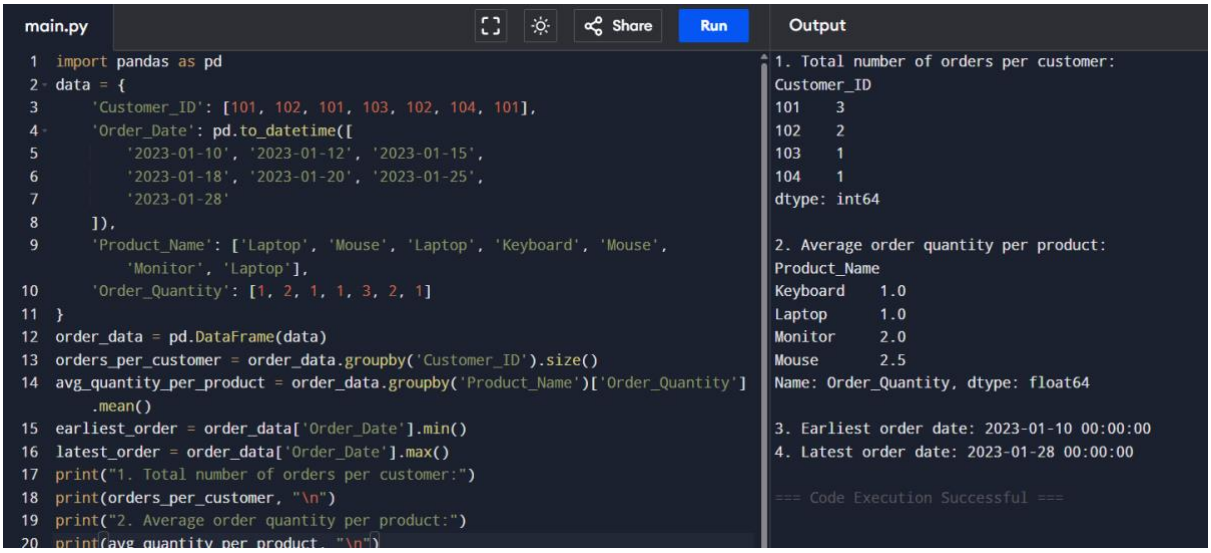


6. Scenario: You are a cashier at a grocery store and need to calculate the total cost of a customer's purchase, including applicable discounts and taxes. You have the item prices and quantities in separate lists, and the discount and tax rates are given as percentages. Your task is to calculate the total cost for the customer.
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?

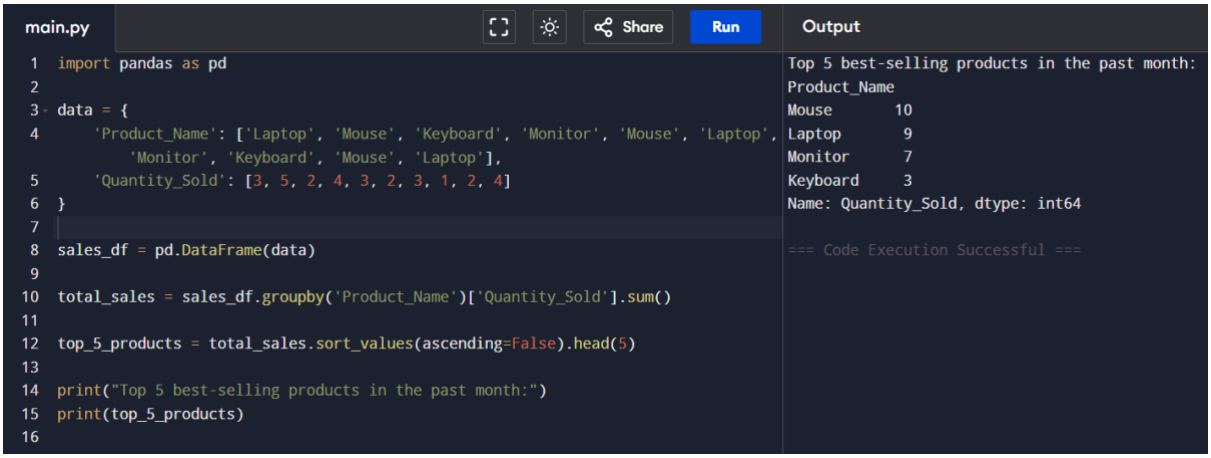


7. Scenario: You are working as a data analyst for an e-commerce company. You have been given a dataset containing information about customer orders, stored in a Pandas DataFrame named order_data. The DataFrame has columns for customer ID, order date, product name, and order quantity. Your task is to analyze the data and answer specific questions about the orders.
Question: Using Pandas DataFrame operations, how would you find the following information from the order_data DataFrame:
1. The total number of orders made by each customer.
2. The average order quantity for each product.

3. The earliest and latest order dates in the dataset.



8. Scenario: You are a data scientist working for a company that sells products online. You have been tasked with analyzing the sales data for the past month. The data is stored in a Pandas data frame.
Question: How would you find the top 5 products that have been sold the most in the past month?



9. Scenario: You work for a real estate agency and have been given a dataset containing information about properties for sale. The dataset is stored in a Pandas DataFrame named property_data. The DataFrame has columns for property ID, location, number of bedrooms, area in square feet, and listing price. Your task is to analyze the data and answer specific questions about the properties.
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.

main.py

```python
import pandas as pd
data = {
    'Property_ID': [101, 102, 103, 104, 105, 106],
    'Location': ['Downtown', 'Uptown', 'Downtown', 'Midtown', 'Uptown',
        'Midtown'],
    'Bedrooms': [3, 5, 4, 2, 6, 4],
    'Area_sqft': [1500, 2500, 1800, 1300, 2700, 2200],
    'Listing_Price': [300000, 450000, 350000, 280000, 500000, 400000]
}
property_data = pd.DataFrame(data)

avg_price_per_location = property_data.groupby('Location')['Listing_Price'].mean
    ()
num_properties_more_than_4_bed = property_data[property_data['Bedrooms'] > 4]
    .shape[0]
property_largest_area = property_data.loc[property_data['Area_sqft'].idxmax()]
print("1. Average Listing Price by Location:")
print(avg_price_per_location, "\n")
print(f"2. Number of properties with more than 4 bedrooms:
    {num_properties_more_than_4_bed}\n")
print("3. Property with the largest area:")
print(property_largest_area)
```

Output

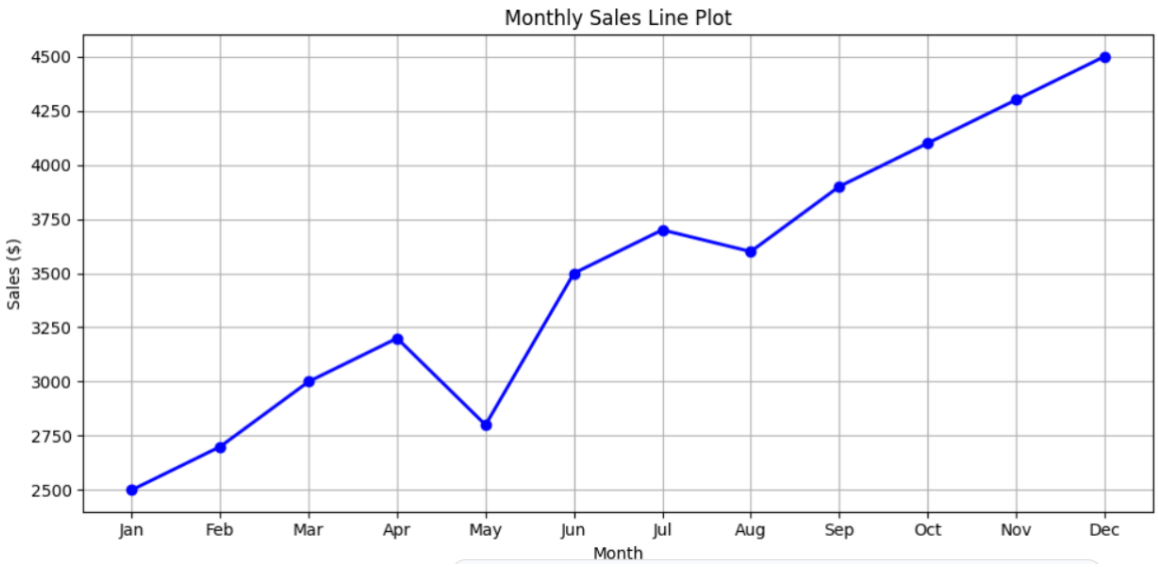
```
1. Average Listing Price by Location:
Location
Downtown    325000.0
Midtown     340000.0
Uptown      475000.0
Name: Listing_Price, dtype: float64

2. Number of properties with more than 4 bedrooms: 2

3. Property with the largest area:
Property_ID         105
Location         Uptown
Bedrooms              6
Area_sqft          2700
Listing_Price    500000
Name: 4, dtype: object

=== Code Execution Successful ===
```

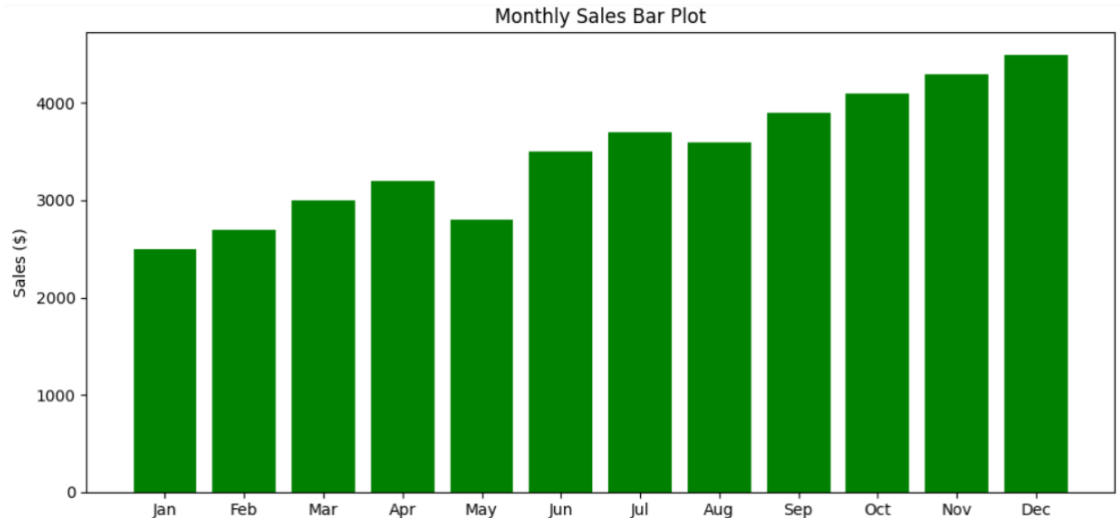
10. Scenario: You are working on a data visualization project and need to create basic plots using Matplotlib. You have a dataset containing the monthly sales data for a company, including the month and corresponding sales values. Your task is to develop a Python program that generates line plots and bar plots to visualize the sales data.

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?

```python
import matplotlib.pyplot as plt

# Sample sales data
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
sales = [2500, 2700, 3000, 3200, 2800, 3500, 3700, 3600, 3900, 4100, 4300, 4500]

# 1. Line Plot of Monthly Sales
plt.figure(figsize=(10, 5))
plt.plot(months, sales, marker='o', color='blue', linestyle='-', linewidth=2)
plt.title('Monthly Sales Line Plot')
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.grid(True)
plt.tight_layout()
plt.show()

# 2. Bar Plot of Monthly Sales
plt.figure(figsize=(10, 5))
plt.bar(months, sales, color='green')
plt.title('Monthly Sales Bar Plot')
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.tight_layout()
plt.show()
```


Monthly Sales Line Plot
Sales ($)
4500
4250
4000
3750
3500
3250
3000
2750
2500
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Monthly Sales Bar Plot
Sales ($)
4000
3000
2000
1000
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec