IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 6.py =
Average Fuel Efficiency of Model A: 28.4 MPG
Average Fuel Efficiency of Model B: 33.6 MPG
Percentage Improvement in Fuel Efficiency: 18.31%
>>>
>>>
```

program 6.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 6.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Fuel efficiency data for two car models (in miles per gallon)
model_a_efficiency = np.array([25, 30, 28, 32, 27])
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 2.py =
Average Price: 165.55555555555554
>>>
```

program 2.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 2.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Sample sales data for 3 products over 3 weeks
sales_data = np.array([[100, 150, 200],
                       [120, 180, 240],
                       [130, 160, 210]])

# Calculate the average price of all products sold in the past month
average_price = np.mean(sales_data)
print("Average Price:", average_price)
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 3.py =
350000.0
>>>
```

program 3.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 3.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Sample data: [bedrooms, sale_price]
data = np.array([[3, 250000],
                 [5, 300000],
                 [4, 275000],
                 [6, 400000],
                 [2, 200000],
                 [5, 350000]])

# Filter houses with more than four bedrooms
filtered_houses = data[data[:, 0] > 4]

# Calculate the average sale price
average_price = np.mean(filtered_houses[:, 1])

print(average_price)
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/Program 1.py ==========
Average Scores for Each Subject: [83.25 82.75 82.   89.25]
Subject with Highest Average Score: 4
>>>
```

Program 1.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/Program 1.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Sample 4x4 matrix representing marks of 4 students in 4 subjects
marks = np.array([[85, 90, 78, 92],
                  [88, 76, 95, 89],
                  [90, 85, 80, 91],
                  [70, 80, 75, 85]])

# Calculate average score for each subject
average_scores = np.mean(marks, axis=0)

# Determine the subject with the highest average score
highest_average_subject = np.argmax(average_scores)

print("Average Scores for Each Subject:", average_scores)
print("Subject with Highest Average Score:", highest_average_subject + 1)  # Add
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 4.py =
>>>
========== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 4.py ==========
>>>
========== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 4.py ==========
Total sales for the year: 90000
Percentage increase from Q1 to Q4: 100.00%
>>>
```

program 4.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 4.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Sales data for each quarter
sales = np.array([15000, 20000, 25000, 30000])

# Total sales for the year
total_sales = np.sum(sales)

# Percentage increase from the first quarter to the fourth quarter
percentage_increase = ((sales[3] - sales[0]) / sales[0]) * 100

# Display the results
print("Total sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4: {:.2f}%".format(percentage_increase))
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 6.py =
Average Fuel Efficiency of Model A: 28.4 MPG
Average Fuel Efficiency of Model B: 33.6 MPG
Percentage Improvement in Fuel Efficiency: 18.31%
>>>
```

program 6.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 6.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Fuel efficiency data for two car models (in miles per gallon)
model_a_efficiency = np.array([25, 30, 28, 32, 27])
model_b_efficiency = np.array([30, 35, 33, 36, 34])

# Calculate average fuel efficiency for both models
average_a = np.mean(model_a_efficiency)
average_b = np.mean(model_b_efficiency)

# Calculate percentage improvement in fuel efficiency
percentage_improvement = ((average_b - average_a) / average_a) * 100

print(f"Average Fuel Efficiency of Model A: {average_a} MPG")
print(f"Average Fuel Efficiency of Model B: {average_b} MPG")
print(f"Percentage Improvement in Fuel Efficiency: {percentage_improvement:.2f}%
-
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 9.py =
Average Listing Price by Location:
 location
Chicago        630000.0
Los Angeles    785000.0
New York       885000.0
Name: listing_price, dtype: float64

Number of Properties with More Than Four Bedrooms: 2

Property with the Largest Area:
 location         Chicago
listing_price     630000
bedrooms               6
area                3000
Name: 3, dtype: object
>>>
```

program 9.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 9.py (3.8.10)

File Edit Format Run Options Window Help

```
import pandas as pd

# Sample data for demonstration (you can replace this with your actual DataFrame
data = {
    'location': ['New York', 'Los Angeles', 'New York', 'Chicago', 'Los Angeles'
    'listing_price': [850000, 760000, 920000, 630000, 810000],
    'bedrooms': [3, 5, 4, 6, 2],
    'area': [1800, 2500, 1600, 3000, 2200]
}

# Create the DataFrame
property_data = pd.DataFrame(data)

# 1. Average listing price of properties in each location
average_price = property_data.groupby('location')['listing_price'].mean()

# 2. Number of properties with more than four bedrooms
properties_with_more_than_four_bedrooms = property_data[property_data['bedrooms'

# 3. Property with the largest area
largest_property = property_data.loc[property_data['area'].idxmax()]

# Output the results
print("Average Listing Price by Location:\n", average_price)
print("\nNumber of Properties with More Than Four Bedrooms:", properties_with_mo
print("\nProperty with the Largest Area:\n", largest_property)
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 7.py =
Total sales for the year: 90000
Percentage increase from Q1 to Q4: 100.00%
>>>
```

program 7.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 7.py (3.8.10)

File Edit Format Run Options Window Help

```
import numpy as np

# Sales data for each quarter
sales = np.array([15000, 20000, 25000, 30000])

# Total sales for the year
total_sales = np.sum(sales)

# Percentage increase from the first quarter to the fourth quarter
percentage_increase = ((sales[3] - sales[0]) / sales[0]) * 100

# Output the results
print("Total sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4: {:.2f}%".format(percentage_increase))
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license
>>>
== RESTART: C:/Users/suman/AppData/Local/Progra
   product_id product_name  quantity_sold
3           9            I            450
5           7            G            400
7           8            H            350
3           4            D            300
9          10            J            300
>>>
```

program 8.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 8.py (3.8.10)

File Edit Format Run Options Window Help

```
# Top 5 Sold Products in the Past Month

import pandas as pd
from datetime import datetime, timedelta

# Sample data
data = {
    'product_id': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'product_name': ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'],
    'quantity_sold': [150, 200, 50, 300, 100, 250, 400, 350, 450, 300],
    'sale_date': [
        datetime.now() - timedelta(days=5),
        datetime.now() - timedelta(days=10),
        datetime.now() - timedelta(days=15),
        datetime.now() - timedelta(days=20),
        datetime.now() - timedelta(days=25),
        datetime.now() - timedelta(days=30),
        datetime.now() - timedelta(days=5),
        datetime.now() - timedelta(days=10),
        datetime.now() - timedelta(days=15),
        datetime.now() - timedelta(days=20)
    ]
}

# Create DataFrame
df = pd.DataFrame(data)

# Filter for the past month
one_month_ago = datetime.now() - timedelta(days=30)
recent_sales = df[df['sale_date'] >= one_month_ago]

# Get top 5 products sold
top_products = recent_sales.nlargest(5, 'quantity_sold')

print(top_products[['product_id', 'product_name', 'quantity_sold']])
```

program 10.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 10.py (3.8.10)

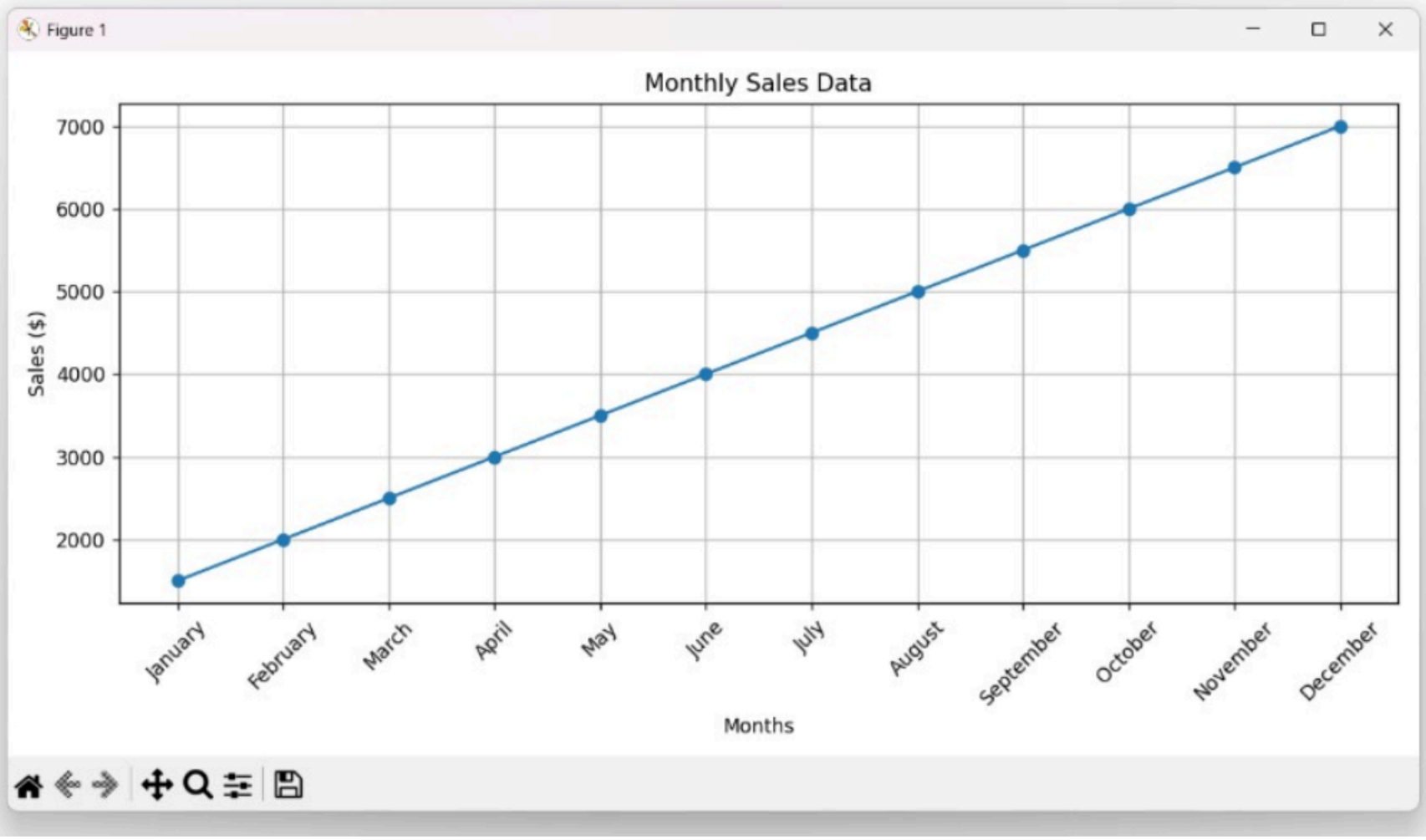
File Edit Format Run Options Window Help

```
# Monthly Sales Data Line Plot

import matplotlib.pyplot as plt

# Sample monthly sales data
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
sales = [1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000]

# Create the line plot
plt.figure(figsize=(10, 5))
plt.plot(months, sales, marker='o')
plt.title('Monthly Sales Data')
plt.xlabel('Months')
plt.ylabel('Sales ($)')
plt.xticks(rotation=45)
plt.grid()
plt.tight_layout()
plt.show()
```



IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/suman/AppData/Local/Programs/Python/Python38/program 5.py =
Total cost: $614.25
>>>
```

program 5.py - C:/Users/suman/AppData/Local/Programs/Python/Python38/program 5.py (3.8.10)

File Edit Format Run Options Window Help

```
def calculate_total_cost(item_prices, quantities, discount_rate, tax_rate):
    subtotal = sum(price * quantity for price, quantity in zip(item_prices, quan
    discount = subtotal * discount_rate
    total_after_discount = subtotal - discount
    tax = total_after_discount * tax_rate
    total_cost = total_after_discount + tax
    return total_cost

# Example usage
item_prices = [100, 200, 50]
quantities = [1, 2, 3]
discount_rate = 0.1  # 10% discount
tax_rate = 0.05      # 5% tax

total = calculate_total_cost(item_prices, quantities, discount_rate, tax_rate)
print(f'Total cost: ${total:.2f}')
```