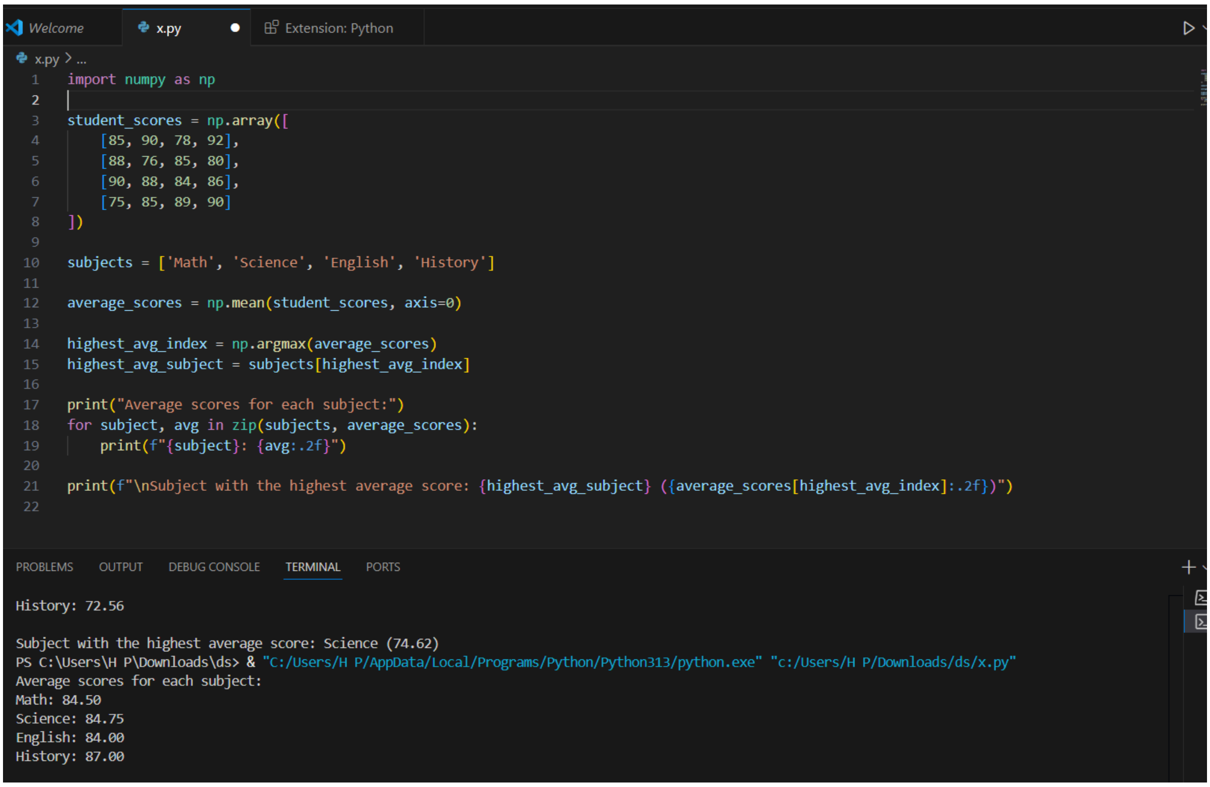
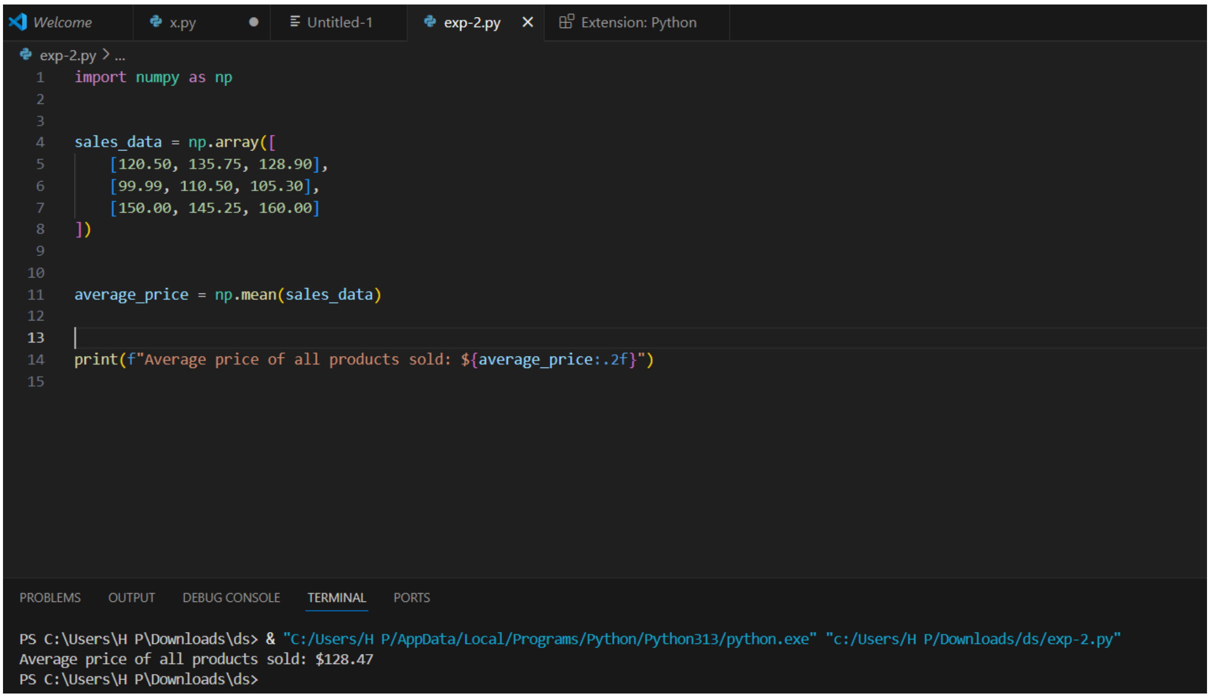
EXP-1



```python
import numpy as np

student_scores = np.array([
    [85, 90, 78, 92],
    [88, 76, 85, 80],
    [90, 88, 84, 86],
    [75, 85, 89, 90]
])

subjects = ['Math', 'Science', 'English', 'History']

average_scores = np.mean(student_scores, axis=0)

highest_avg_index = np.argmax(average_scores)
highest_avg_subject = subjects[highest_avg_index]

print("Average scores for each subject:")
for subject, avg in zip(subjects, average_scores):
    print(f"{subject}: {avg:.2f}")

print(f"\nSubject with the highest average score: {highest_avg_subject} ({average_scores[highest_avg_index]:.2f})")

```

```
History: 72.56

Subject with the highest average score: Science (74.62)
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/x.py"
Average scores for each subject:
Math: 84.50
Science: 84.75
English: 84.00
History: 87.00
```

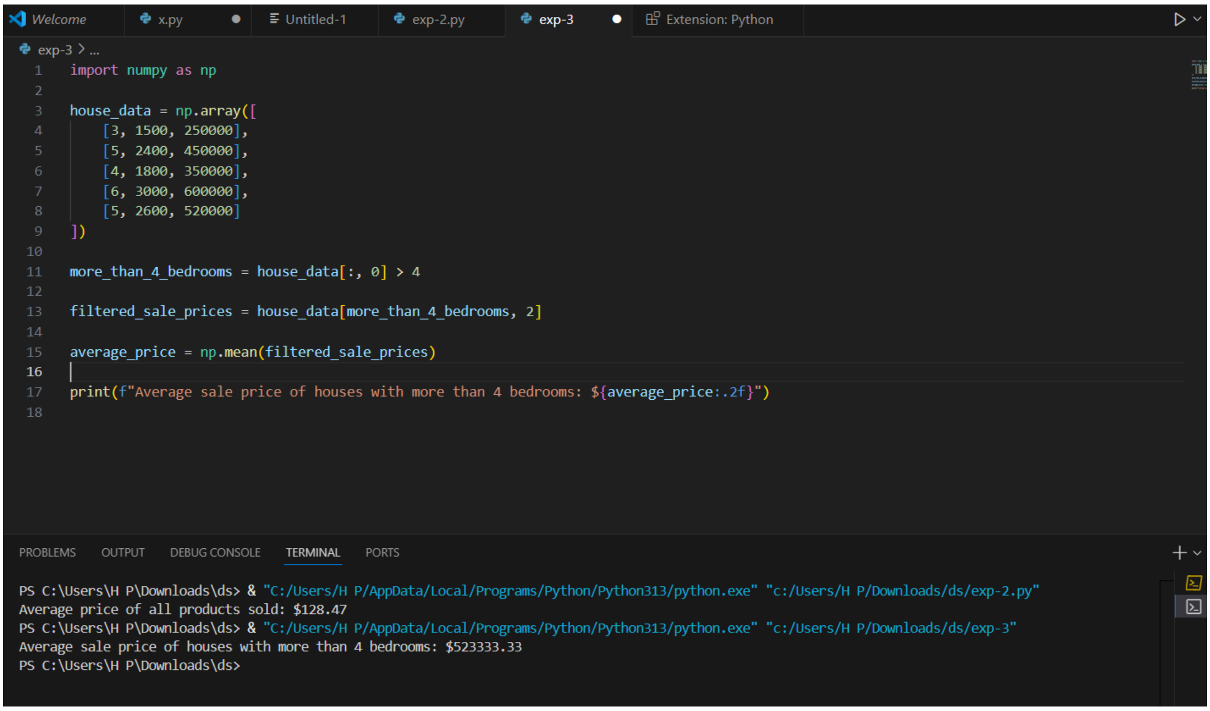
EXP-2



```python
import numpy as np


sales_data = np.array([
    [120.50, 135.75, 128.90],
    [99.99, 110.50, 105.30],
    [150.00, 145.25, 160.00]
])


average_price = np.mean(sales_data)


print(f"Average price of all products sold: ${average_price:.2f}")

```

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-2.py"
Average price of all products sold: $128.47
PS C:\Users\H P\Downloads\ds>
```

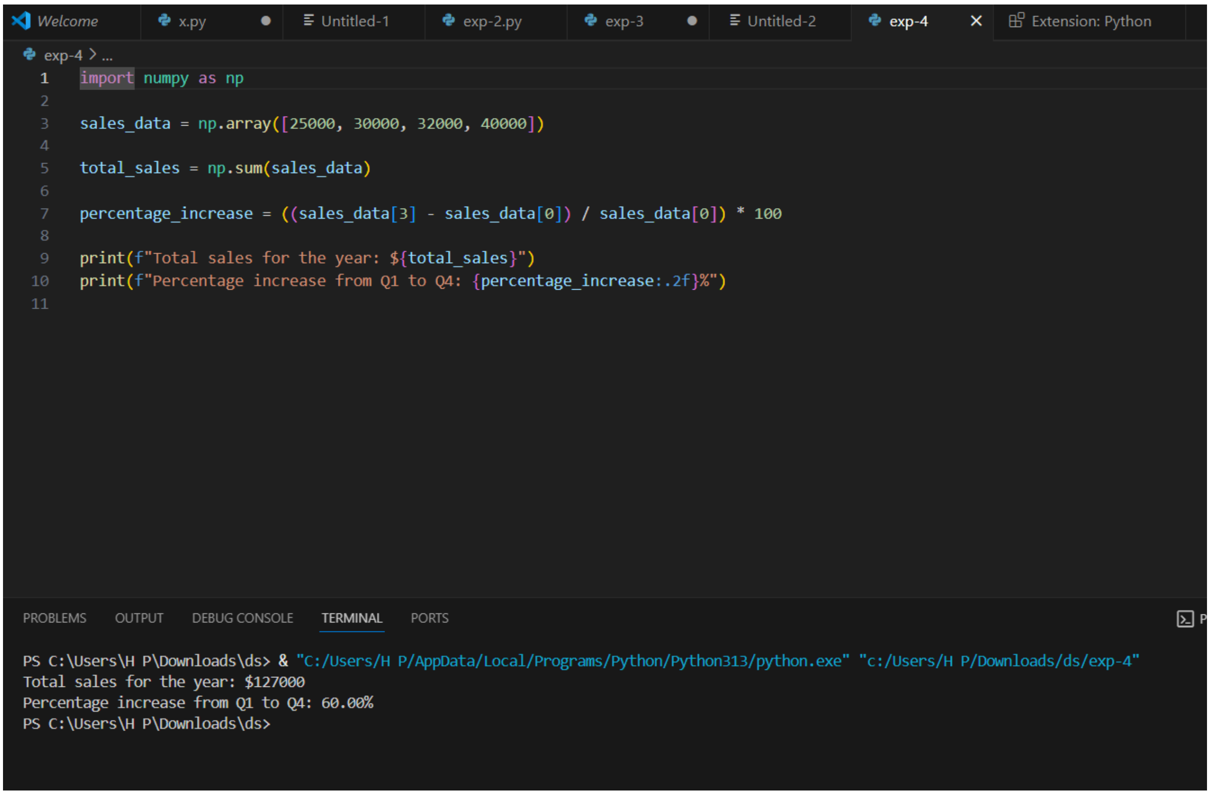
EXP-3



```python
import numpy as np

house_data = np.array([
    [3, 1500, 250000],
    [5, 2400, 450000],
    [4, 1800, 350000],
    [6, 3000, 600000],
    [5, 2600, 520000]
])

more_than_4_bedrooms = house_data[:, 0] > 4

filtered_sale_prices = house_data[more_than_4_bedrooms, 2]

average_price = np.mean(filtered_sale_prices)

print(f"Average sale price of houses with more than 4 bedrooms: ${average_price:.2f}")
```

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-2.py"
Average price of all products sold: $128.47
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-3"
Average sale price of houses with more than 4 bedrooms: $523333.33
PS C:\Users\H P\Downloads\ds>
```

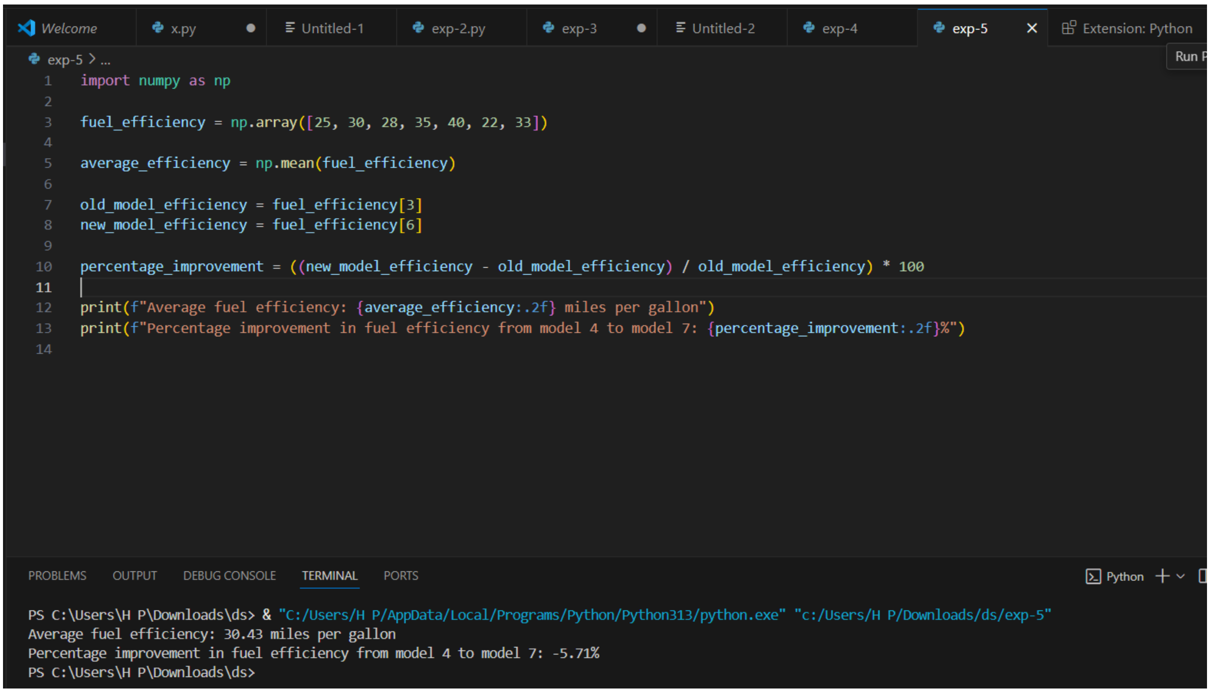
EXP-4



```python
import numpy as np

sales_data = np.array([25000, 30000, 32000, 40000])

total_sales = np.sum(sales_data)

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

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

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-4"
Total sales for the year: $127000
Percentage increase from Q1 to Q4: 60.00%
PS C:\Users\H P\Downloads\ds>
```

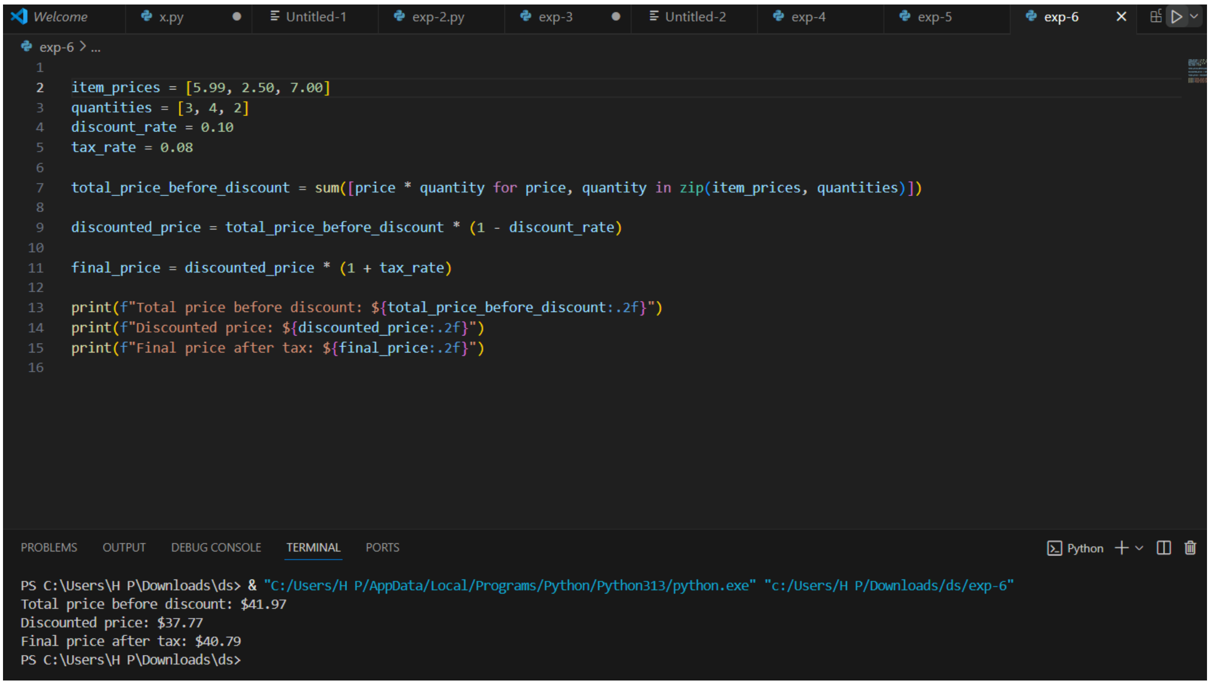
EXP-5



```python
import numpy as np

fuel_efficiency = np.array([25, 30, 28, 35, 40, 22, 33])

average_efficiency = np.mean(fuel_efficiency)

old_model_efficiency = fuel_efficiency[3]
new_model_efficiency = fuel_efficiency[6]

percentage_improvement = ((new_model_efficiency - old_model_efficiency) / old_model_efficiency) * 100

print(f"Average fuel efficiency: {average_efficiency:.2f} miles per gallon")
print(f"Percentage improvement in fuel efficiency from model 4 to model 7: {percentage_improvement:.2f}%")
```

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-5"
Average fuel efficiency: 30.43 miles per gallon
Percentage improvement in fuel efficiency from model 4 to model 7: -5.71%
PS C:\Users\H P\Downloads\ds>
```

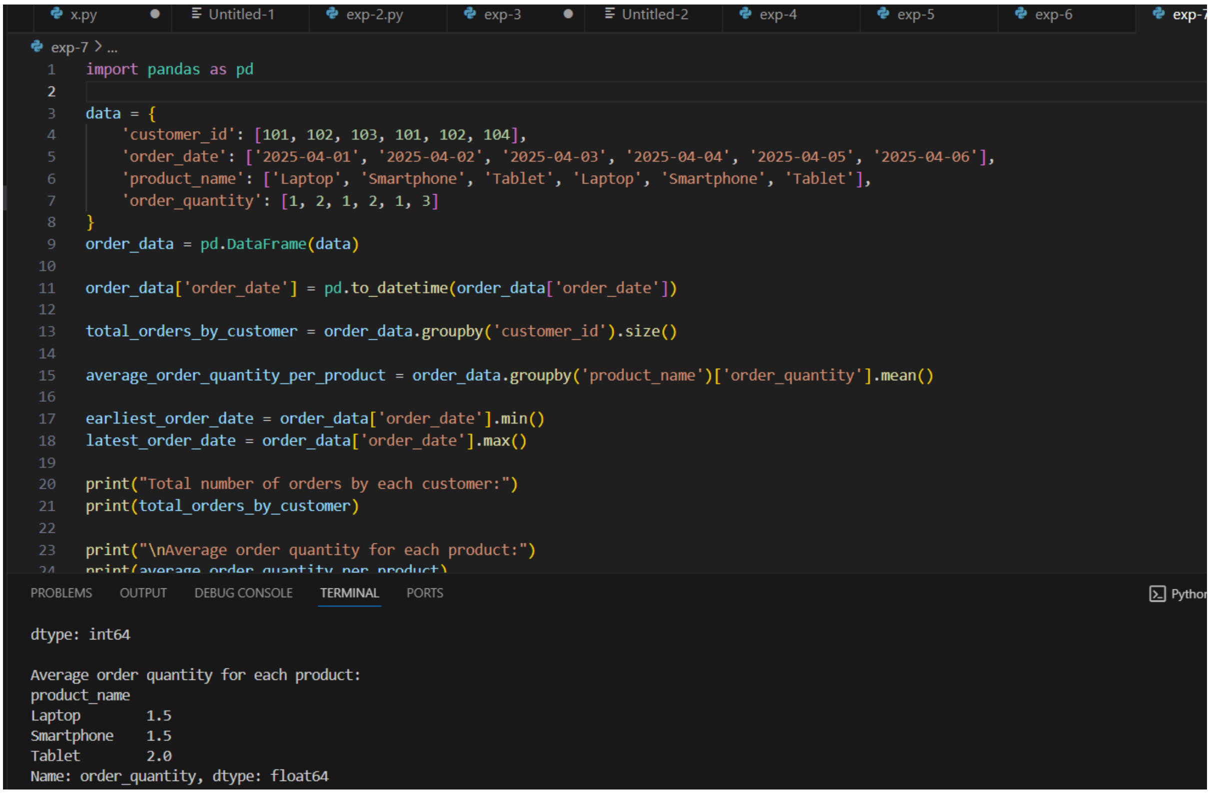
EXP-6



```python

item_prices = [5.99, 2.50, 7.00]
quantities = [3, 4, 2]
discount_rate = 0.10
tax_rate = 0.08

total_price_before_discount = sum([price * quantity for price, quantity in zip(item_prices, quantities)])

discounted_price = total_price_before_discount * (1 - discount_rate)

final_price = discounted_price * (1 + tax_rate)

print(f"Total price before discount: ${total_price_before_discount:.2f}")
print(f"Discounted price: ${discounted_price:.2f}")
print(f"Final price after tax: ${final_price:.2f}")
```

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-6"
Total price before discount: $41.97
Discounted price: $37.77
Final price after tax: $40.79
PS C:\Users\H P\Downloads\ds>
```

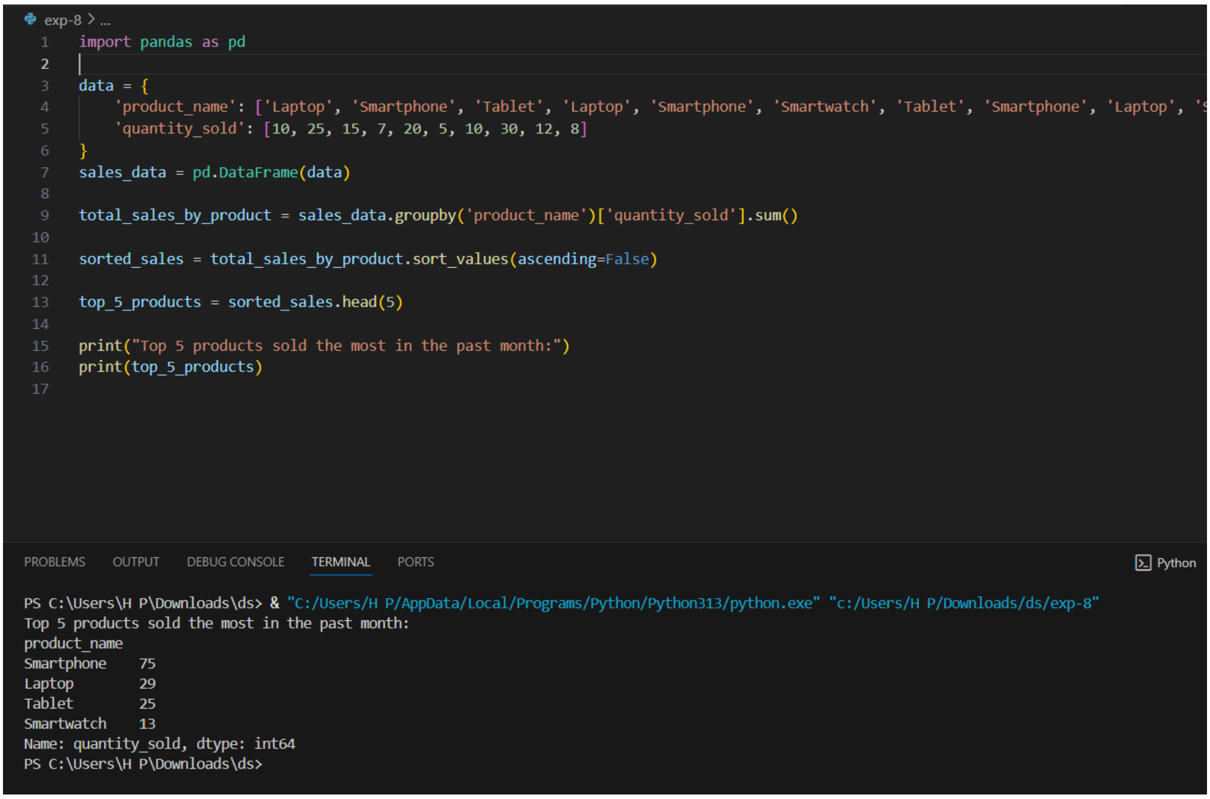
EXP-7



```python
import pandas as pd

data = {
    'customer_id': [101, 102, 103, 101, 102, 104],
    'order_date': ['2025-04-01', '2025-04-02', '2025-04-03', '2025-04-04', '2025-04-05', '2025-04-06'],
    'product_name': ['Laptop', 'Smartphone', 'Tablet', 'Laptop', 'Smartphone', 'Tablet'],
    'order_quantity': [1, 2, 1, 2, 1, 3]
}
order_data = pd.DataFrame(data)

order_data['order_date'] = pd.to_datetime(order_data['order_date'])

total_orders_by_customer = order_data.groupby('customer_id').size()

average_order_quantity_per_product = order_data.groupby('product_name')['order_quantity'].mean()

earliest_order_date = order_data['order_date'].min()
latest_order_date = order_data['order_date'].max()

print("Total number of orders by each customer:")
print(total_orders_by_customer)

print("\nAverage order quantity for each product:")
print(average_order_quantity_per_product)
```

```
dtype: int64

Average order quantity for each product:
product_name
Laptop        1.5
Smartphone    1.5
Tablet        2.0
Name: order_quantity, dtype: float64
```

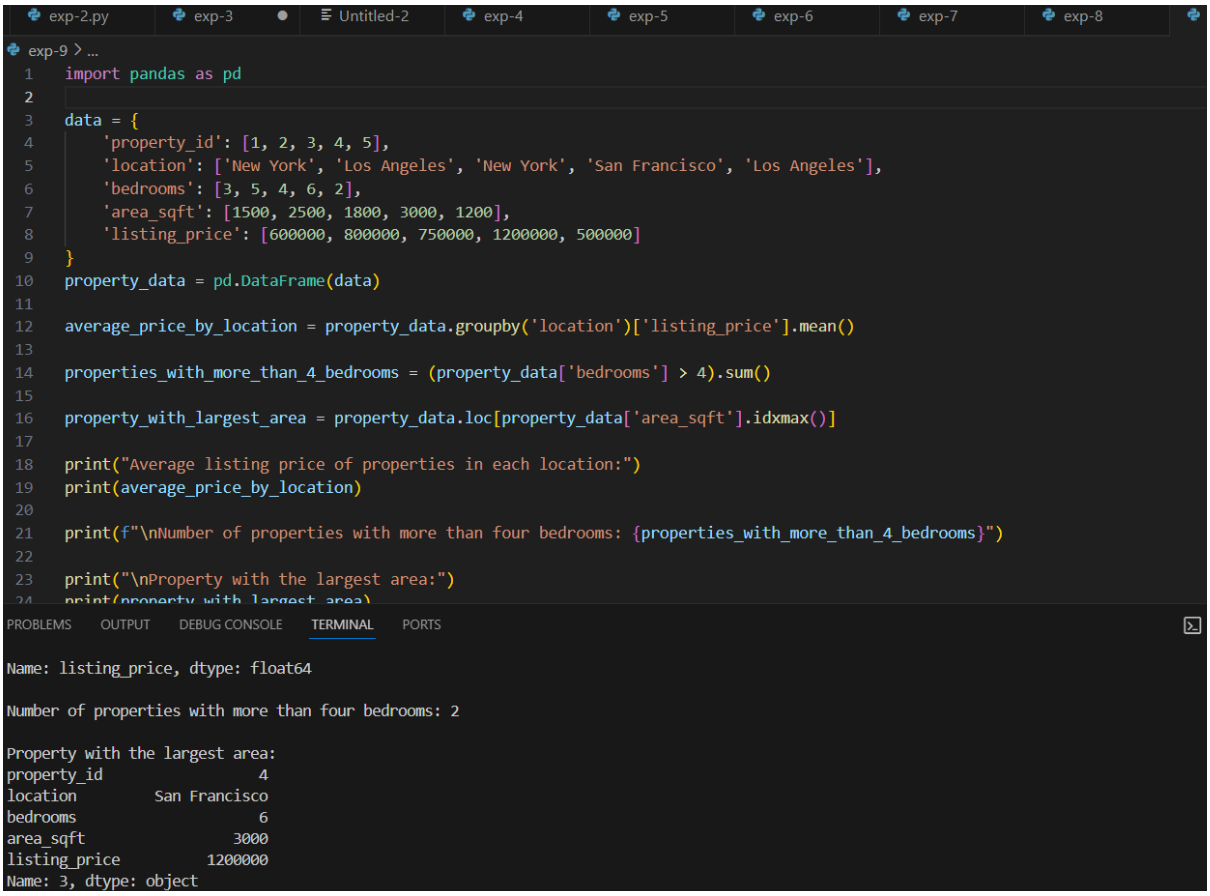
EXP-8



```python
import pandas as pd

data = {
    'product_name': ['Laptop', 'Smartphone', 'Tablet', 'Laptop', 'Smartphone', 'Smartwatch', 'Tablet', 'Smartphone', 'Laptop', 'S
    'quantity_sold': [10, 25, 15, 7, 20, 5, 10, 30, 12, 8]
}
sales_data = pd.DataFrame(data)

total_sales_by_product = sales_data.groupby('product_name')['quantity_sold'].sum()

sorted_sales = total_sales_by_product.sort_values(ascending=False)

top_5_products = sorted_sales.head(5)

print("Top 5 products sold the most in the past month:")
print(top_5_products)
```

```
PS C:\Users\H P\Downloads\ds> & "C:/Users/H P/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/H P/Downloads/ds/exp-8"
Top 5 products sold the most in the past month:
product_name
Smartphone    75
Laptop        29
Tablet        25
Smartwatch    13
Name: quantity_sold, dtype: int64
PS C:\Users\H P\Downloads\ds>
```

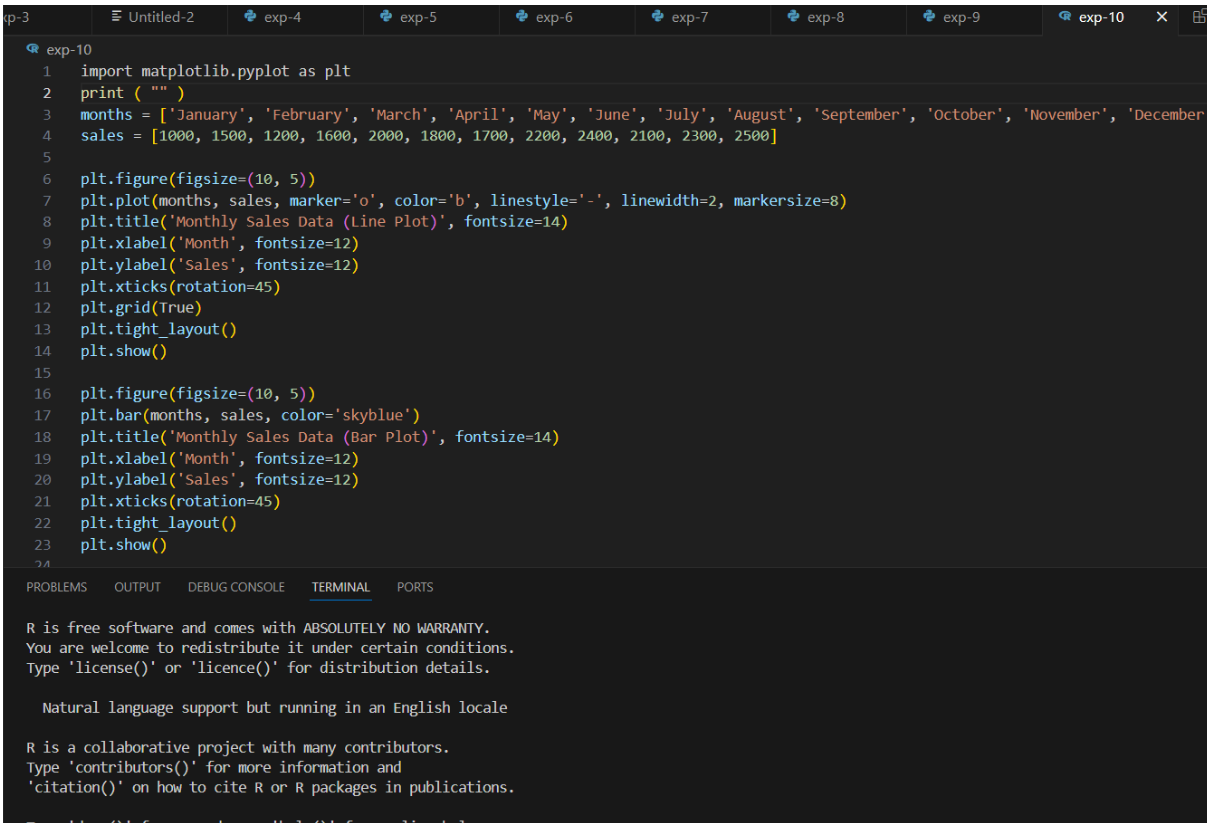
EXP-9



```python
import pandas as pd

data = {
    'property_id': [1, 2, 3, 4, 5],
    'location': ['New York', 'Los Angeles', 'New York', 'San Francisco', 'Los Angeles'],
    'bedrooms': [3, 5, 4, 6, 2],
    'area_sqft': [1500, 2500, 1800, 3000, 1200],
    'listing_price': [600000, 800000, 750000, 1200000, 500000]
}
property_data = pd.DataFrame(data)

average_price_by_location = property_data.groupby('location')['listing_price'].mean()

properties_with_more_than_4_bedrooms = (property_data['bedrooms'] > 4).sum()

property_with_largest_area = property_data.loc[property_data['area_sqft'].idxmax()]

print("Average listing price of properties in each location:")
print(average_price_by_location)

print(f"\nNumber of properties with more than four bedrooms: {properties_with_more_than_4_bedrooms}")

print("\nProperty with the largest area:")
print(property_with_largest_area)
```

```
Name: listing_price, dtype: float64

Number of properties with more than four bedrooms: 2

Property with the largest area:
property_id                4
location       San Francisco
bedrooms                   6
area_sqft               3000
listing_price        1200000
Name: 3, dtype: object
```

EXP-10



```python
import matplotlib.pyplot as plt
print ( "" )
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December
sales = [1000, 1500, 1200, 1600, 2000, 1800, 1700, 2200, 2400, 2100, 2300, 2500]

plt.figure(figsize=(10, 5))
plt.plot(months, sales, marker='o', color='b', linestyle='-', linewidth=2, markersize=8)
plt.title('Monthly Sales Data (Line Plot)', fontsize=14)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Sales', fontsize=12)
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 5))
plt.bar(months, sales, color='skyblue')
plt.title('Monthly Sales Data (Bar Plot)', fontsize=14)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Sales', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```