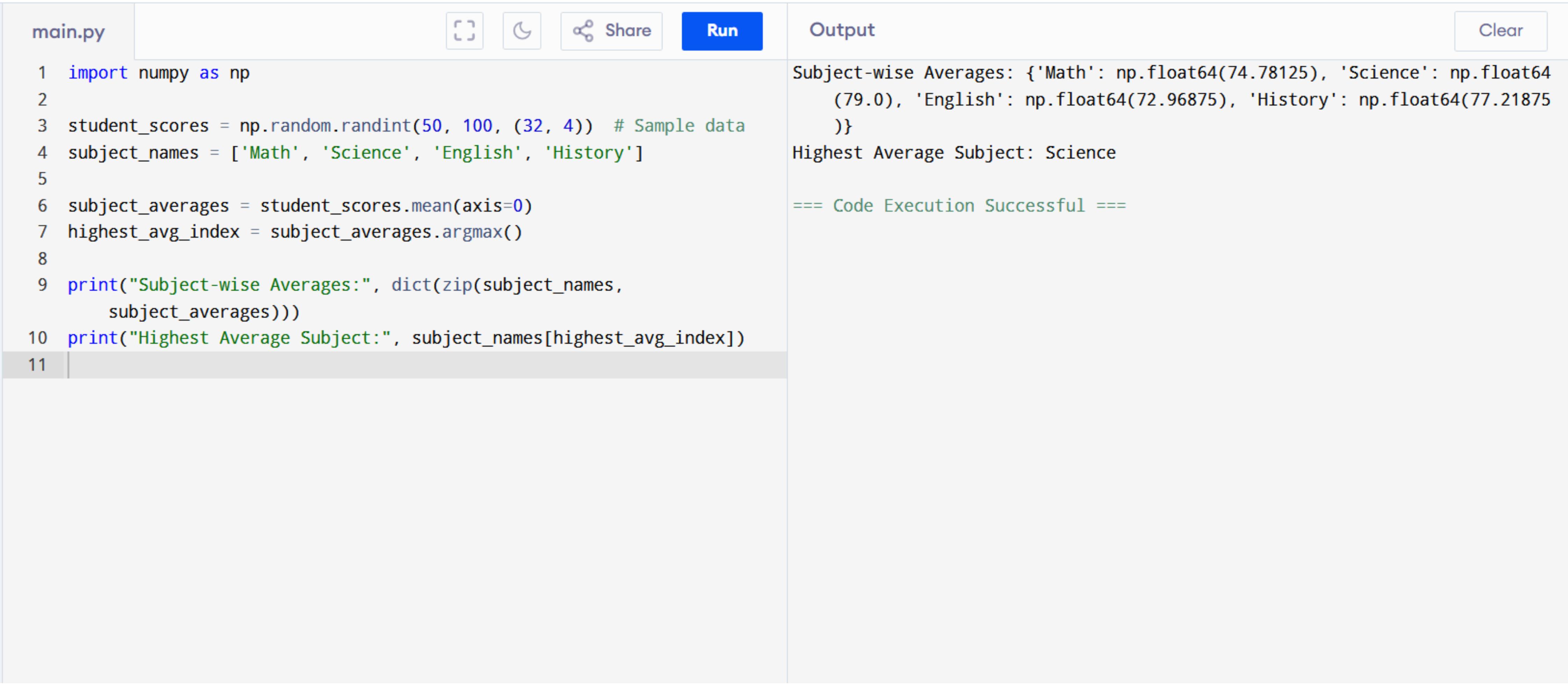
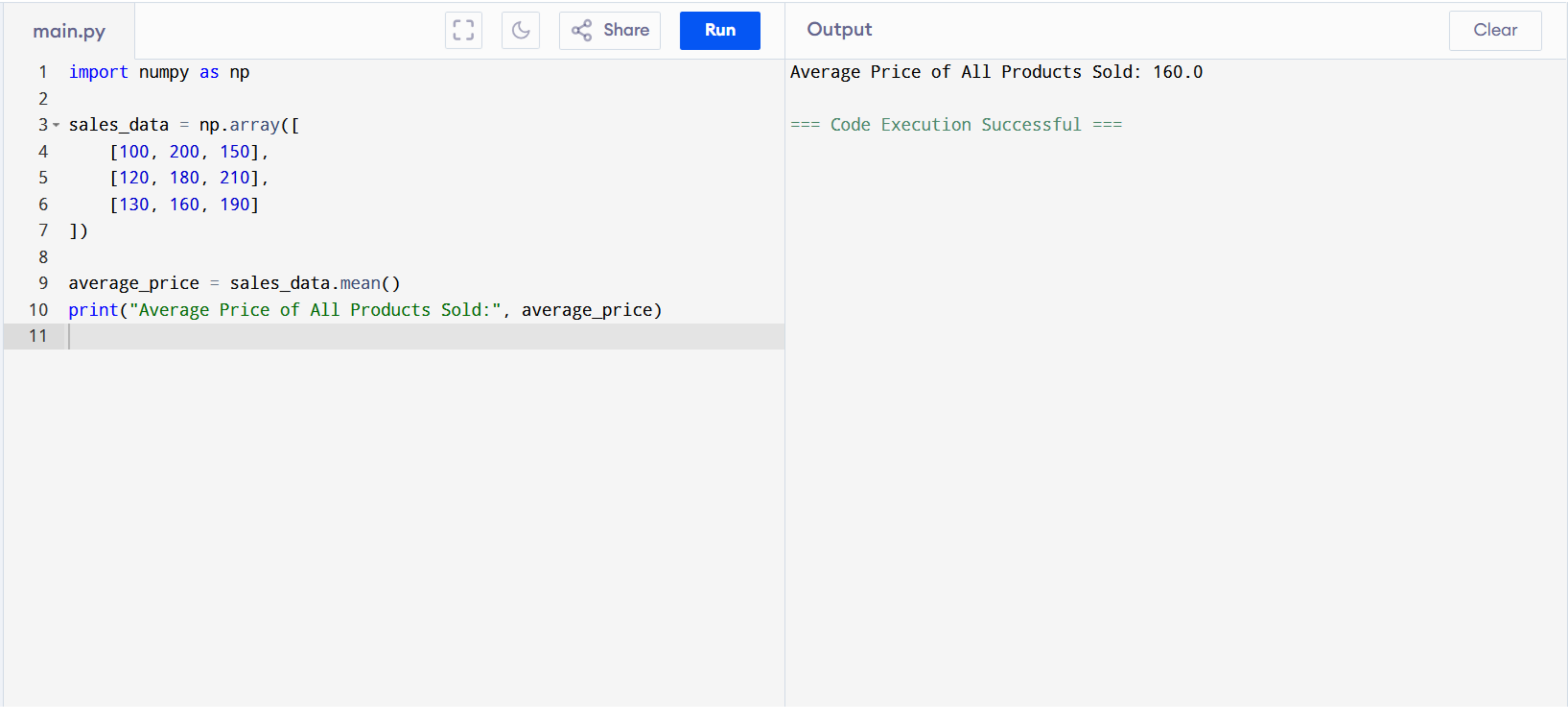


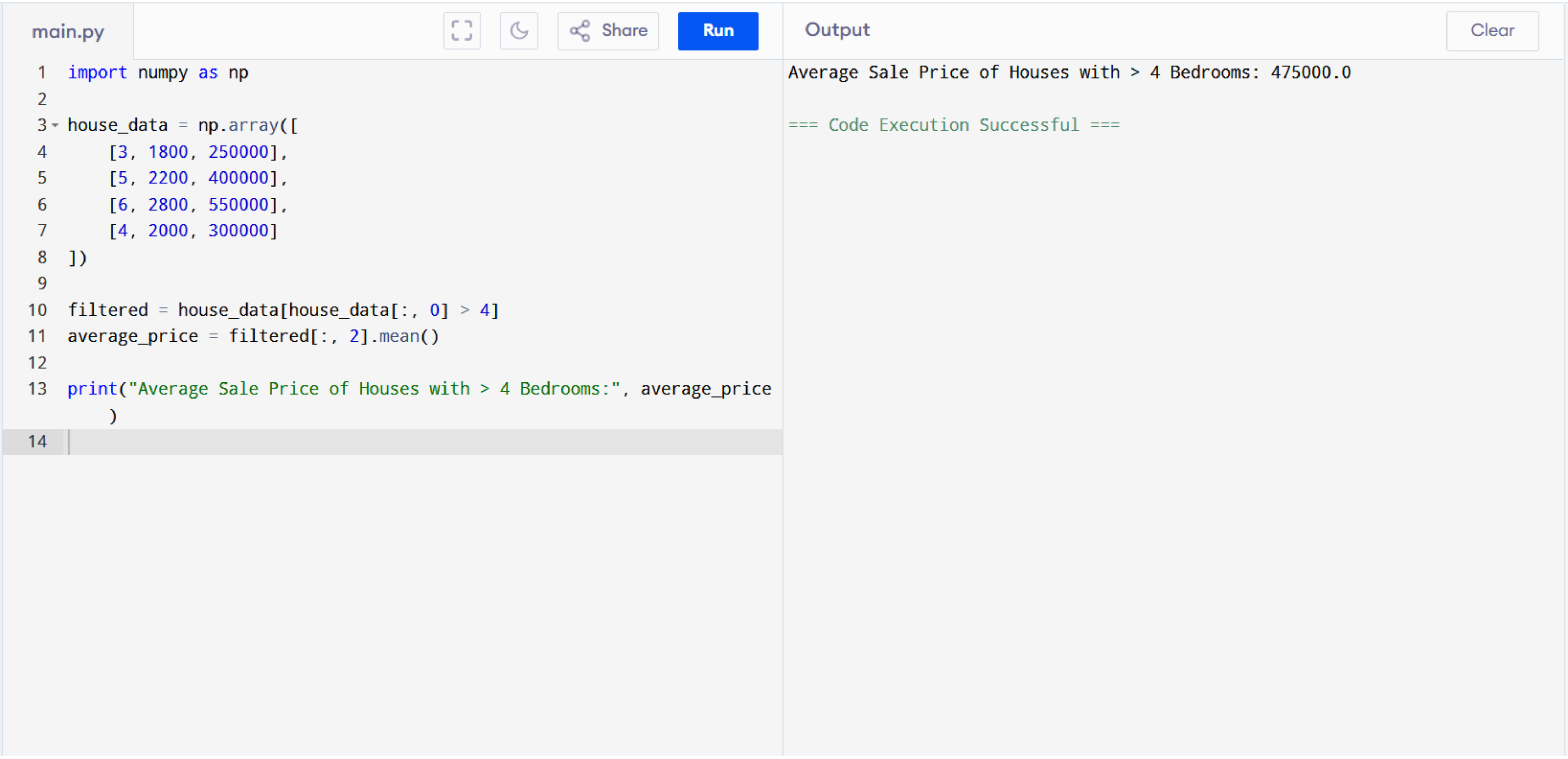
```python
import numpy as np

student_scores = np.random.randint(50, 100, (32, 4))  # Sample data
subject_names = ['Math', 'Science', 'English', 'History']

subject_averages = student_scores.mean(axis=0)
highest_avg_index = subject_averages.argmax()

print("Subject-wise Averages:", dict(zip(subject_names,
    subject_averages)))
print("Highest Average Subject:", subject_names[highest_avg_index])
```

```
Subject-wise Averages: {'Math': np.float64(74.78125), 'Science': np.float64
    (79.0), 'English': np.float64(72.96875), 'History': np.float64(77.21875
    )}
Highest Average Subject: Science

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



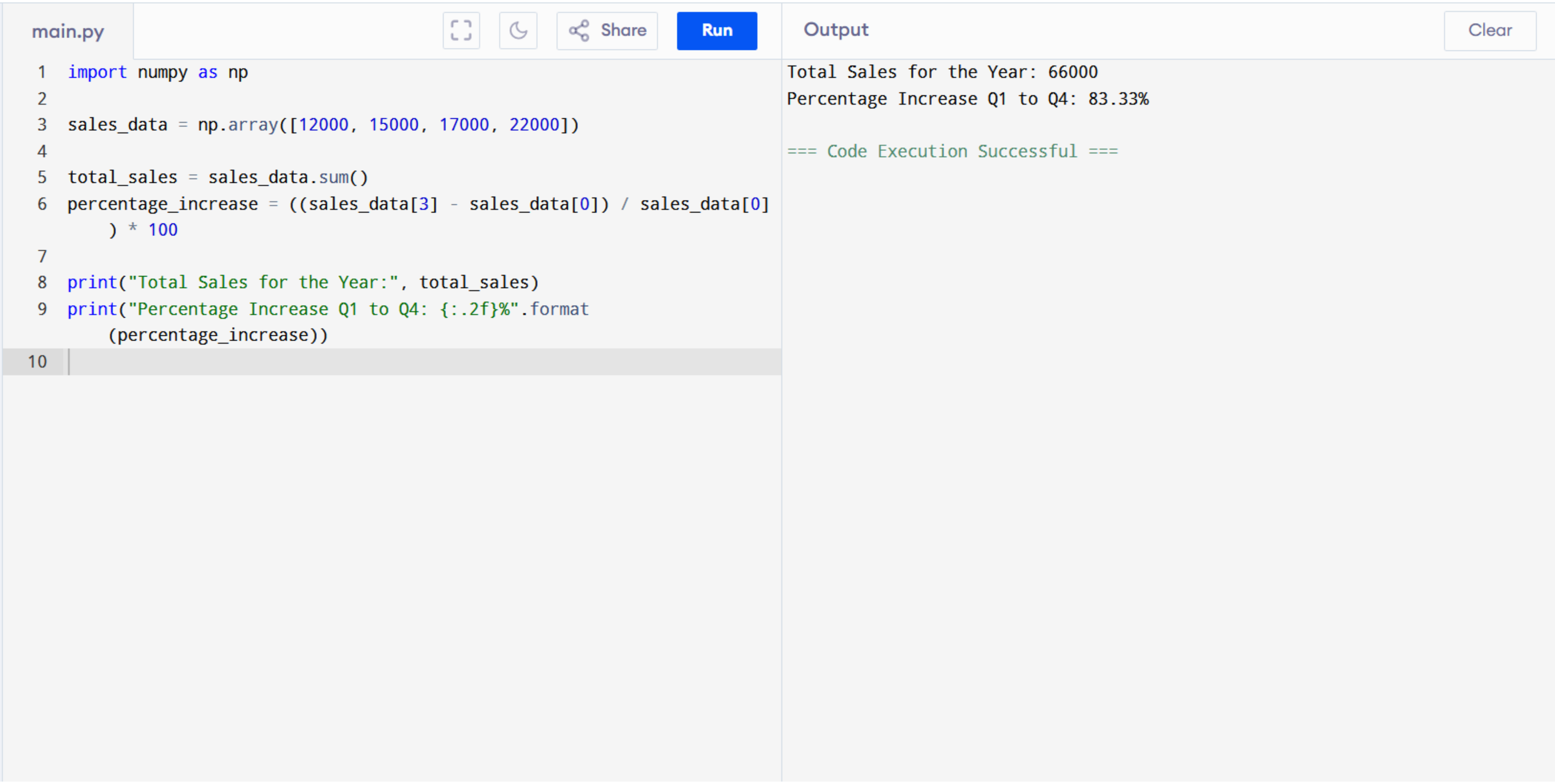
```python
import numpy as np

sales_data = np.array([
    [100, 200, 150],
    [120, 180, 210],
    [130, 160, 190]
])

average_price = sales_data.mean()
print("Average Price of All Products Sold:", average_price)
```

```
Average Price of All Products Sold: 160.0

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



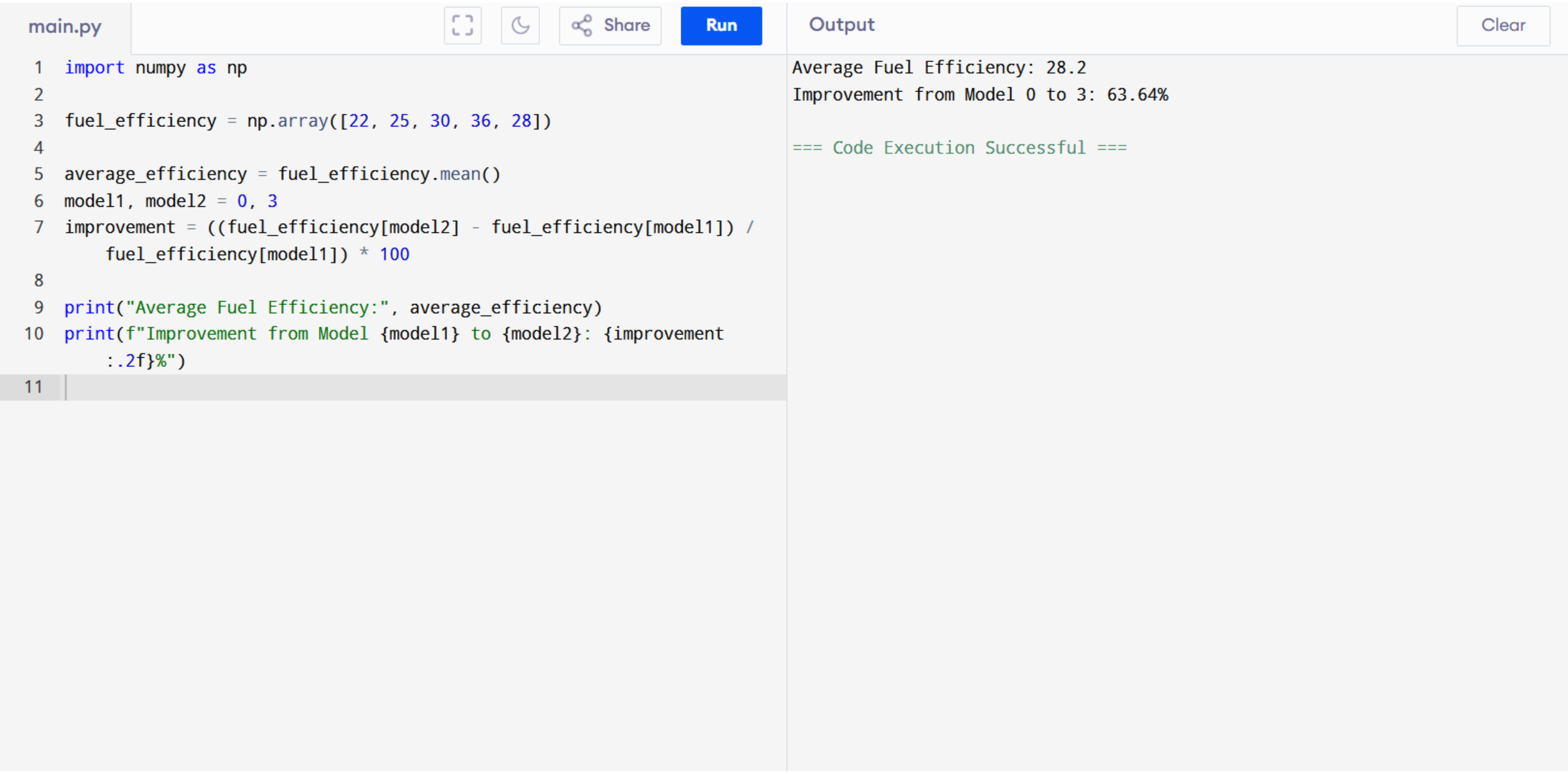
```python
import numpy as np

house_data = np.array([
    [3, 1800, 250000],
    [5, 2200, 400000],
    [6, 2800, 550000],
    [4, 2000, 300000]
])

filtered = house_data[house_data[:, 0] > 4]
average_price = filtered[:, 2].mean()

print("Average Sale Price of Houses with > 4 Bedrooms:", average_price
    )
```

```
Average Sale Price of Houses with > 4 Bedrooms: 475000.0

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



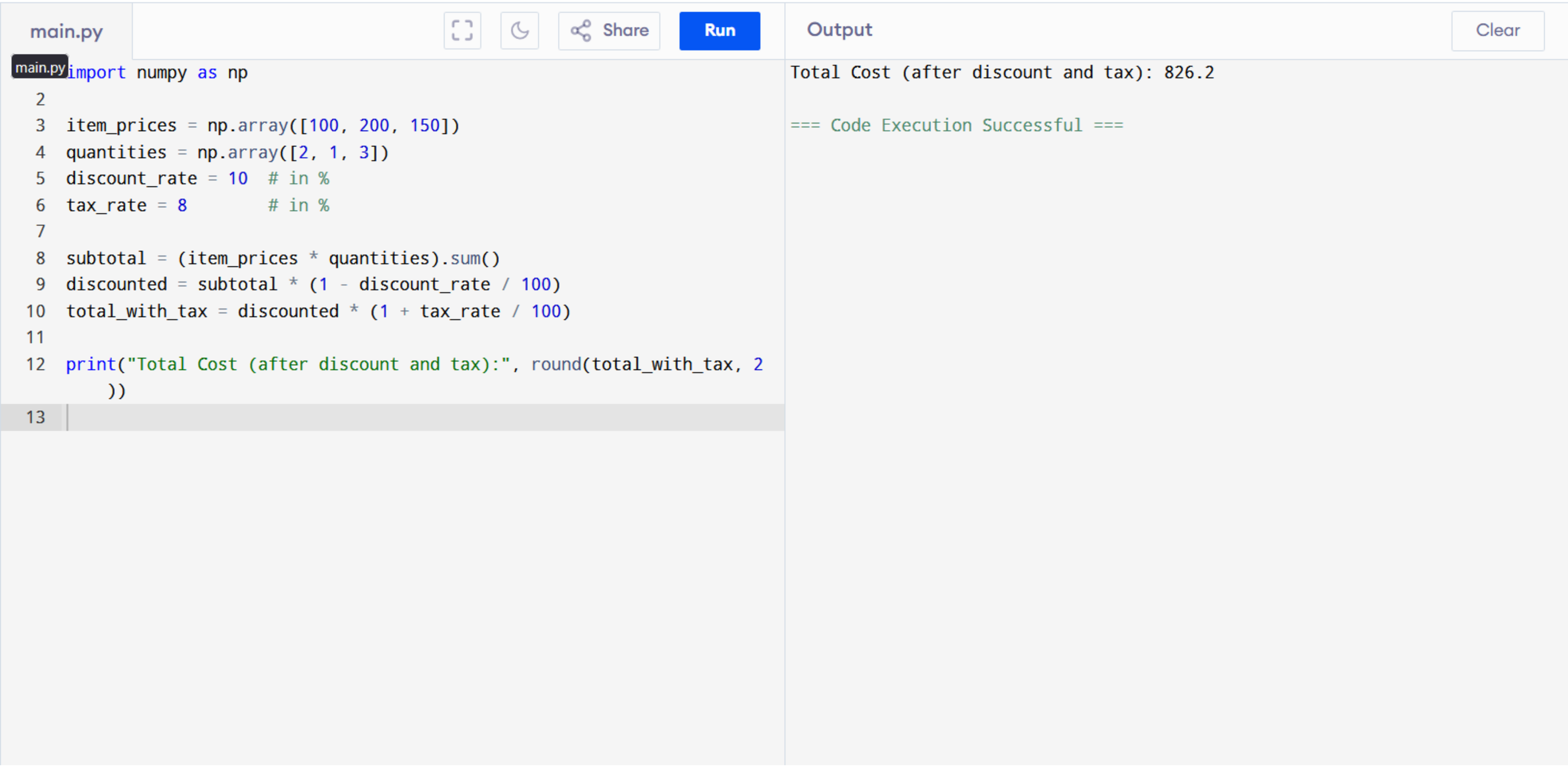
```python
import numpy as np

sales_data = np.array([12000, 15000, 17000, 22000])

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

print("Total Sales for the Year:", total_sales)
print("Percentage Increase Q1 to Q4: {:.2f}%".format
    (percentage_increase))
```

```
Total Sales for the Year: 66000
Percentage Increase Q1 to Q4: 83.33%

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



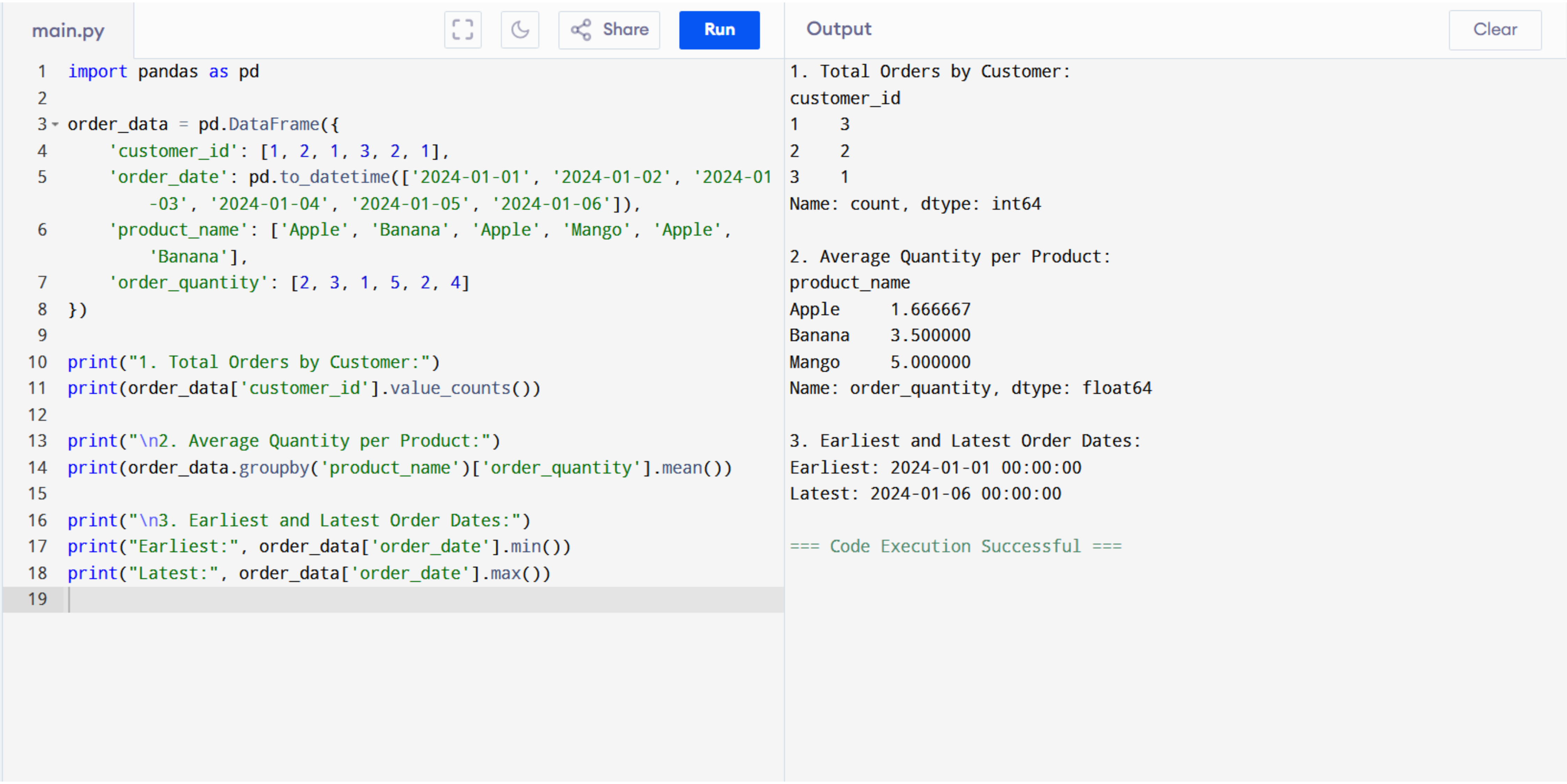
```python
import numpy as np

fuel_efficiency = np.array([22, 25, 30, 36, 28])

average_efficiency = fuel_efficiency.mean()
model1, model2 = 0, 3
improvement = ((fuel_efficiency[model2] - fuel_efficiency[model1]) /
    fuel_efficiency[model1]) * 100

print("Average Fuel Efficiency:", average_efficiency)
print(f"Improvement from Model {model1} to {model2}: {improvement
    :.2f}%")
```

```
Average Fuel Efficiency: 28.2
Improvement from Model 0 to 3: 63.64%

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



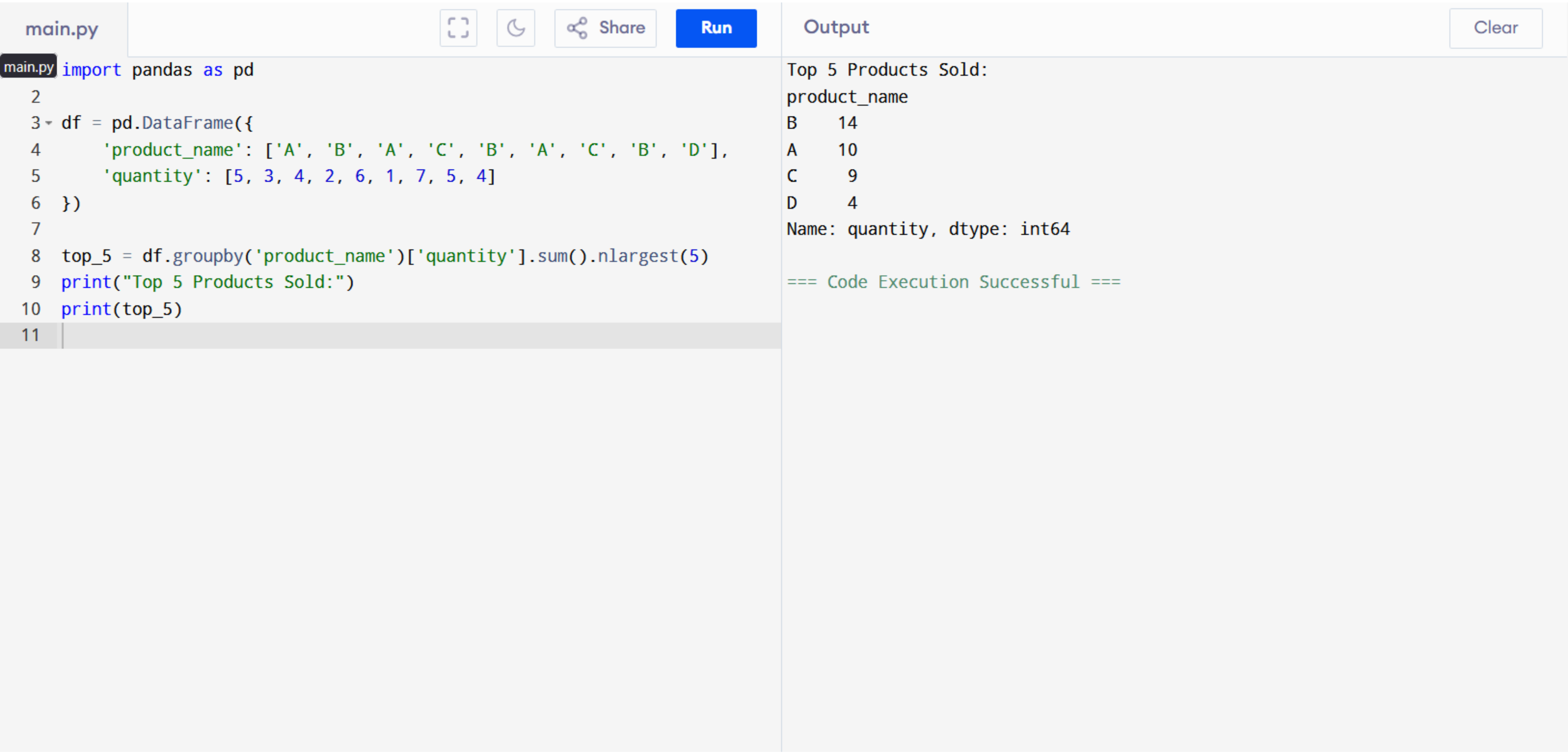
```python
import numpy as np

item_prices = np.array([100, 200, 150])
quantities = np.array([2, 1, 3])
discount_rate = 10  # in %
tax_rate = 8        # in %

subtotal = (item_prices * quantities).sum()
discounted = subtotal * (1 - discount_rate / 100)
total_with_tax = discounted * (1 + tax_rate / 100)

print("Total Cost (after discount and tax):", round(total_with_tax, 2
    ))

```

```
Total Cost (after discount and tax): 826.2

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



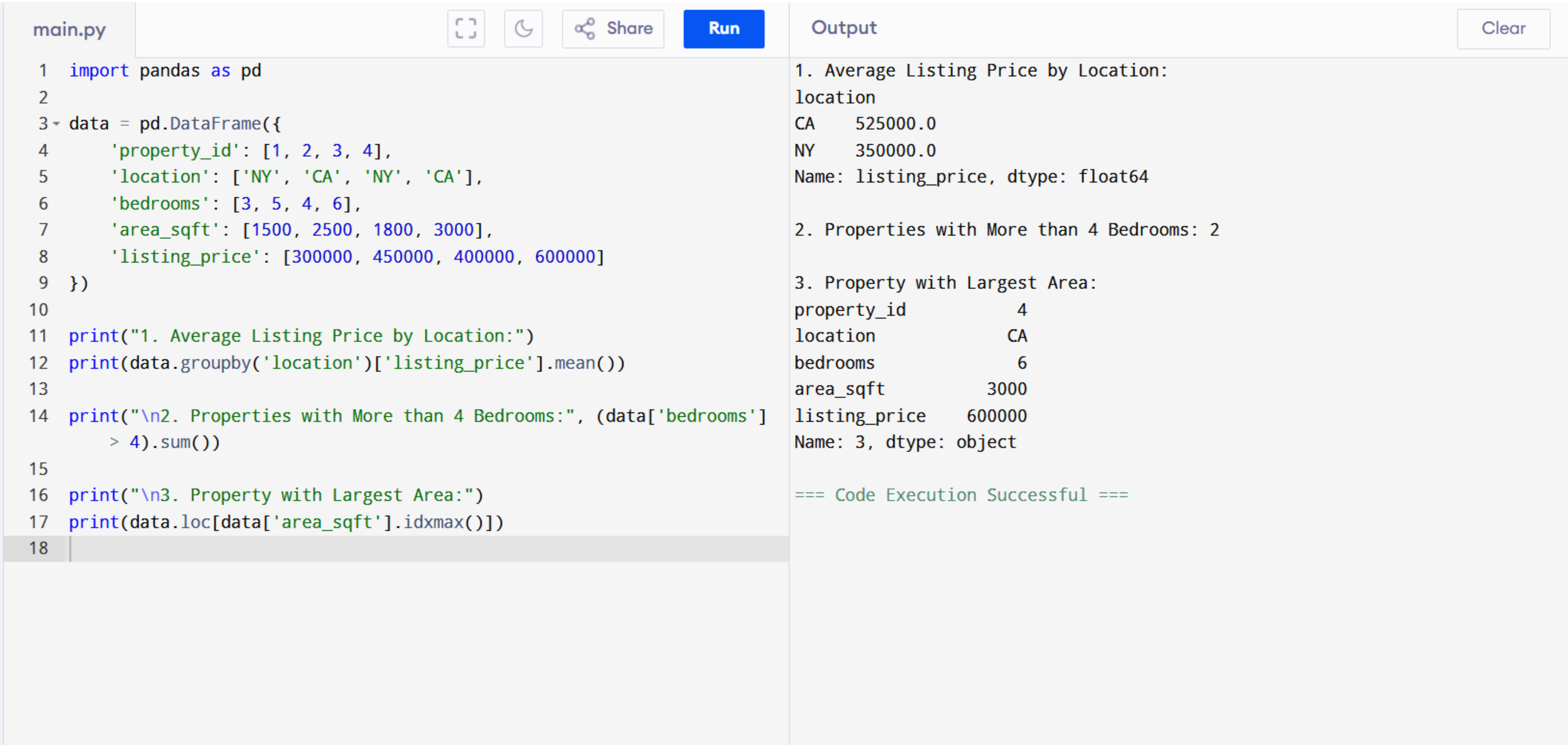
```python
import pandas as pd

order_data = pd.DataFrame({
    'customer_id': [1, 2, 1, 3, 2, 1],
    'order_date': pd.to_datetime(['2024-01-01', '2024-01-02', '2024-01-03', '2024-01-04', '2024-01-05', '2024-01-06']),
    'product_name': ['Apple', 'Banana', 'Apple', 'Mango', 'Apple', 'Banana'],
    'order_quantity': [2, 3, 1, 5, 2, 4]
})

print("1. Total Orders by Customer:")
print(order_data['customer_id'].value_counts())

print("\n2. Average Quantity per Product:")
print(order_data.groupby('product_name')['order_quantity'].mean())

print("\n3. Earliest and Latest Order Dates:")
print("Earliest:", order_data['order_date'].min())
print("Latest:", order_data['order_date'].max())
```

```
1. Total Orders by Customer:
customer_id
1    3
2    2
3    1
Name: count, dtype: int64

2. Average Quantity per Product:
product_name
Apple     1.666667
Banana    3.500000
Mango     5.000000
Name: order_quantity, dtype: float64

3. Earliest and Latest Order Dates:
Earliest: 2024-01-01 00:00:00
Latest: 2024-01-06 00:00:00

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



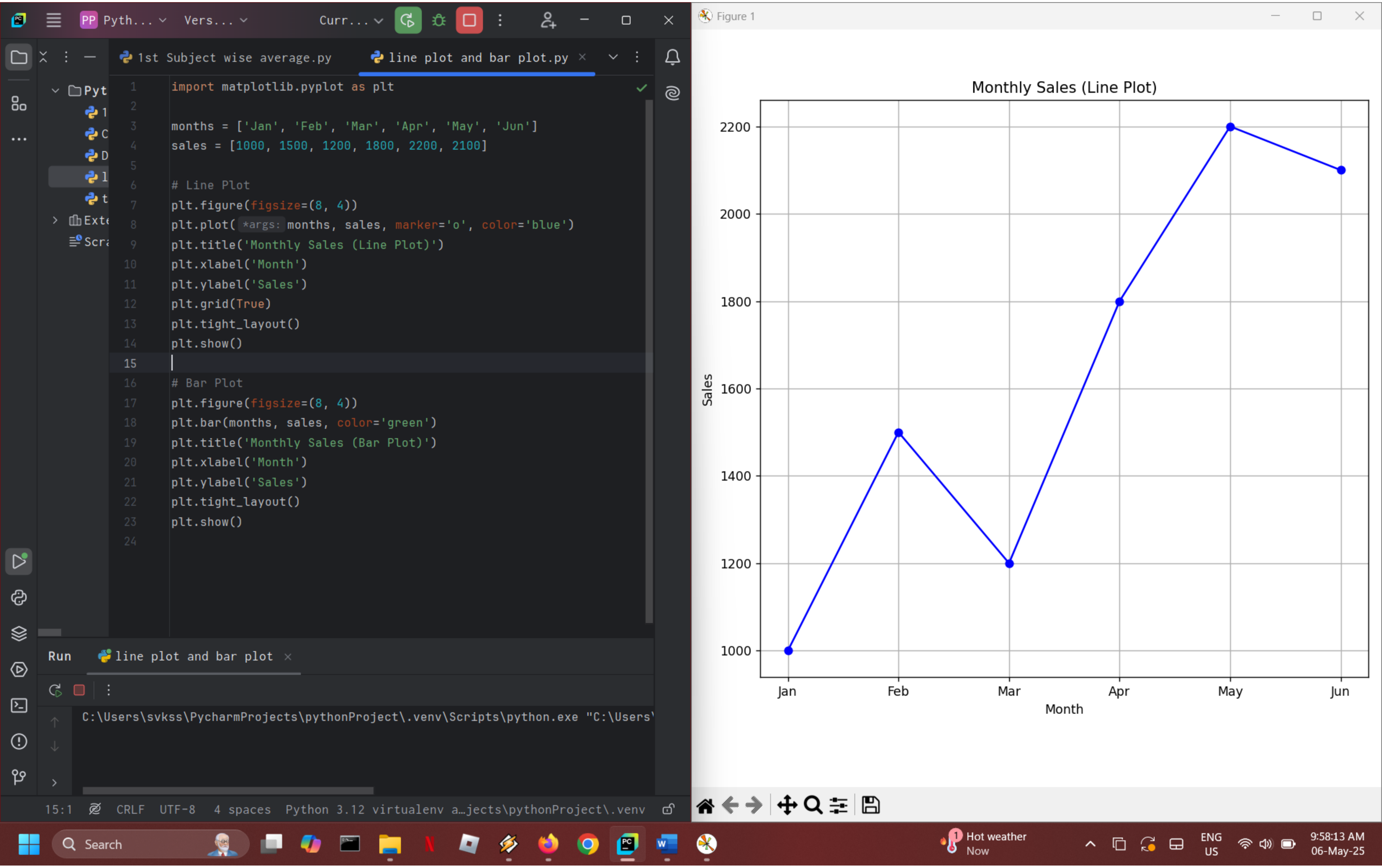
```python
import pandas as pd

df = pd.DataFrame({
    'product_name': ['A', 'B', 'A', 'C', 'B', 'A', 'C', 'B', 'D'],
    'quantity': [5, 3, 4, 2, 6, 1, 7, 5, 4]
})

top_5 = df.groupby('product_name')['quantity'].sum().nlargest(5)
print("Top 5 Products Sold:")
print(top_5)
```

```
Top 5 Products Sold:
product_name
B    14
A    10
C     9
D     4
Name: quantity, dtype: int64

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



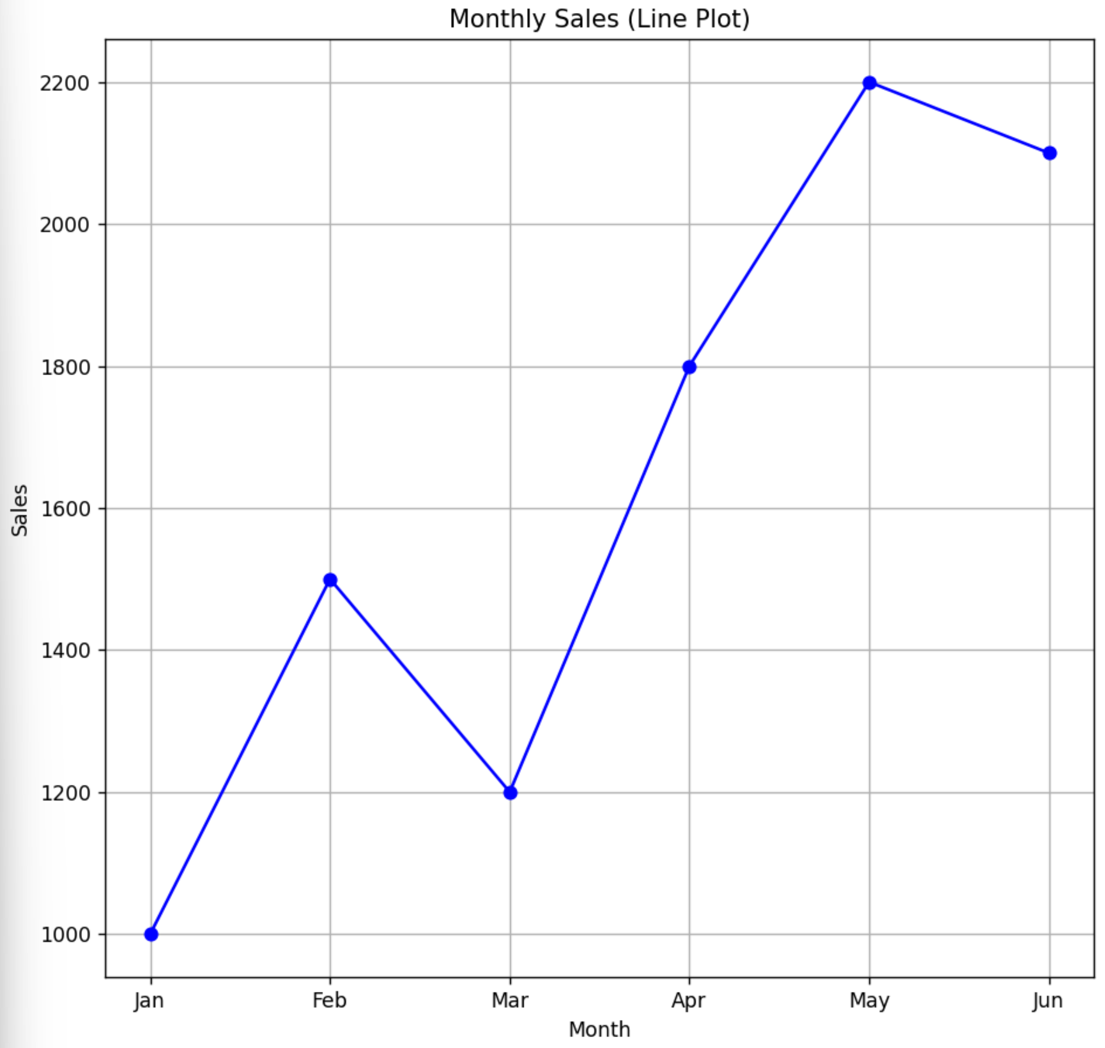
```python
import pandas as pd

data = pd.DataFrame({
    'property_id': [1, 2, 3, 4],
    'location': ['NY', 'CA', 'NY', 'CA'],
    'bedrooms': [3, 5, 4, 6],
    'area_sqft': [1500, 2500, 1800, 3000],
    'listing_price': [300000, 450000, 400000, 600000]
})

print("1. Average Listing Price by Location:")
print(data.groupby('location')['listing_price'].mean())

print("\n2. Properties with More than 4 Bedrooms:", (data['bedrooms']
    > 4).sum())

print("\n3. Property with Largest Area:")
print(data.loc[data['area_sqft'].idxmax()])

```

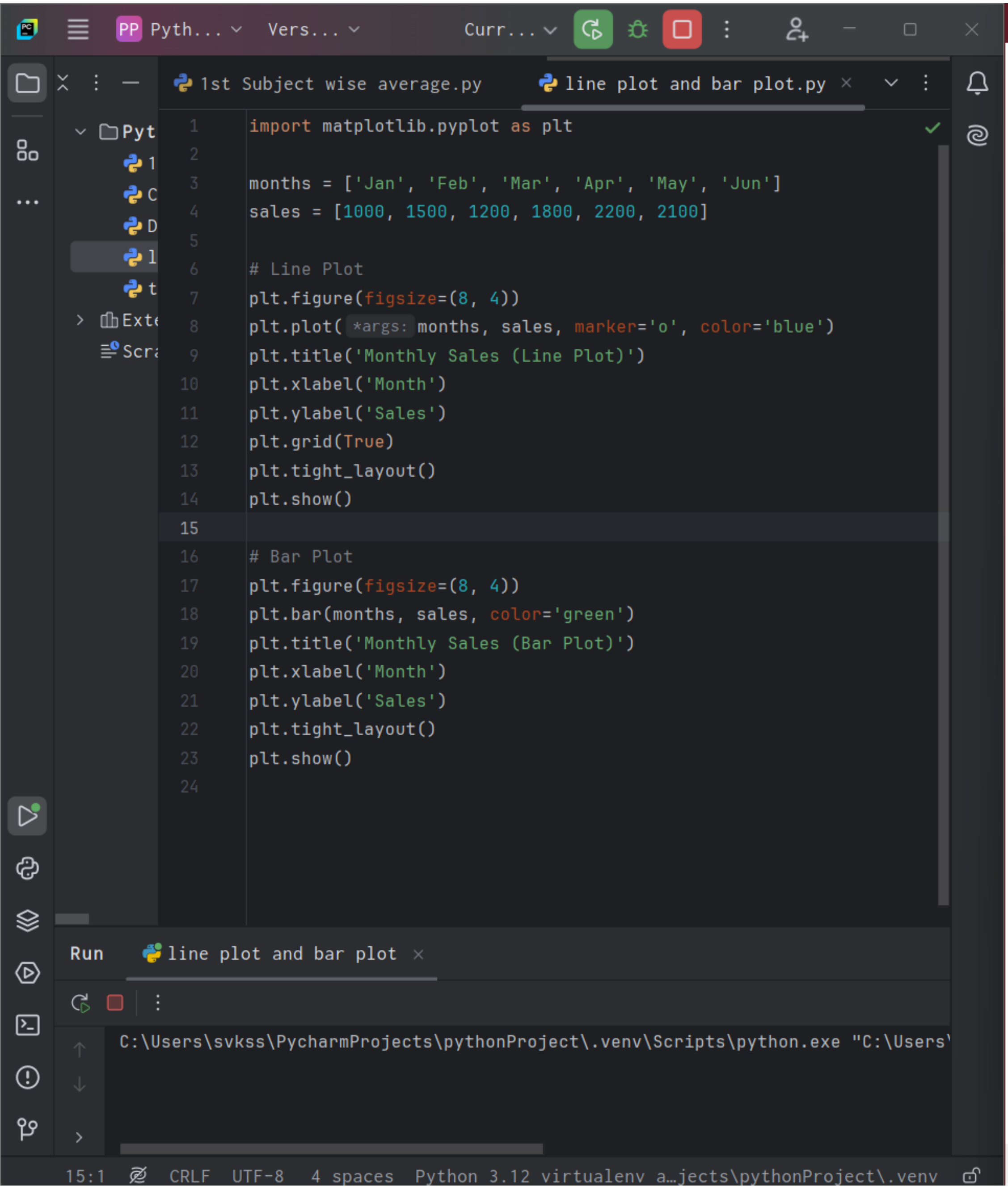
```
1. Average Listing Price by Location:
location
CA    525000.0
NY    350000.0
Name: listing_price, dtype: float64

2. Properties with More than 4 Bedrooms: 2

3. Property with Largest Area:
property_id           4
location             CA
bedrooms              6
area_sqft          3000
listing_price    600000
Name: 3, dtype: object

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



```python
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
sales = [1000, 1500, 1200, 1800, 2200, 2100]

# Line Plot
plt.figure(figsize=(8, 4))
plt.plot( *args: months, sales, marker='o', color='blue')
plt.title('Monthly Sales (Line Plot)')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.tight_layout()
plt.show()

# Bar Plot
plt.figure(figsize=(8, 4))
plt.bar(months, sales, color='green')
plt.title('Monthly Sales (Bar Plot)')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.tight_layout()
plt.show()
```

```
C:\Users\svkss\PycharmProjects\pythonProject\.venv\Scripts\python.exe "C:\Users'
```




Pyth...
Vers...
Curr...
1st Subject wise average.py
line plot and bar plot.py
Run
line plot and bar plot
15:1 CRLF UTF-8 4 spaces Python 3.12 virtualenv a…jects\pythonProject\.venv

```python
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
sales = [1000, 1500, 1200, 1800, 2200, 2100]

# Line Plot
plt.figure(figsize=(8, 4))
plt.plot(months, sales, marker='o', color='blue')
plt.title('Monthly Sales (Line Plot)')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.tight_layout()
plt.show()

# Bar Plot
plt.figure(figsize=(8, 4))
plt.bar(months, sales, color='green')
plt.title('Monthly Sales (Bar Plot)')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.tight_layout()
plt.show()
```

```
C:\Users\svkss\PycharmProjects\pythonProject\.venv\Scripts\python.exe "C:\Users\
```

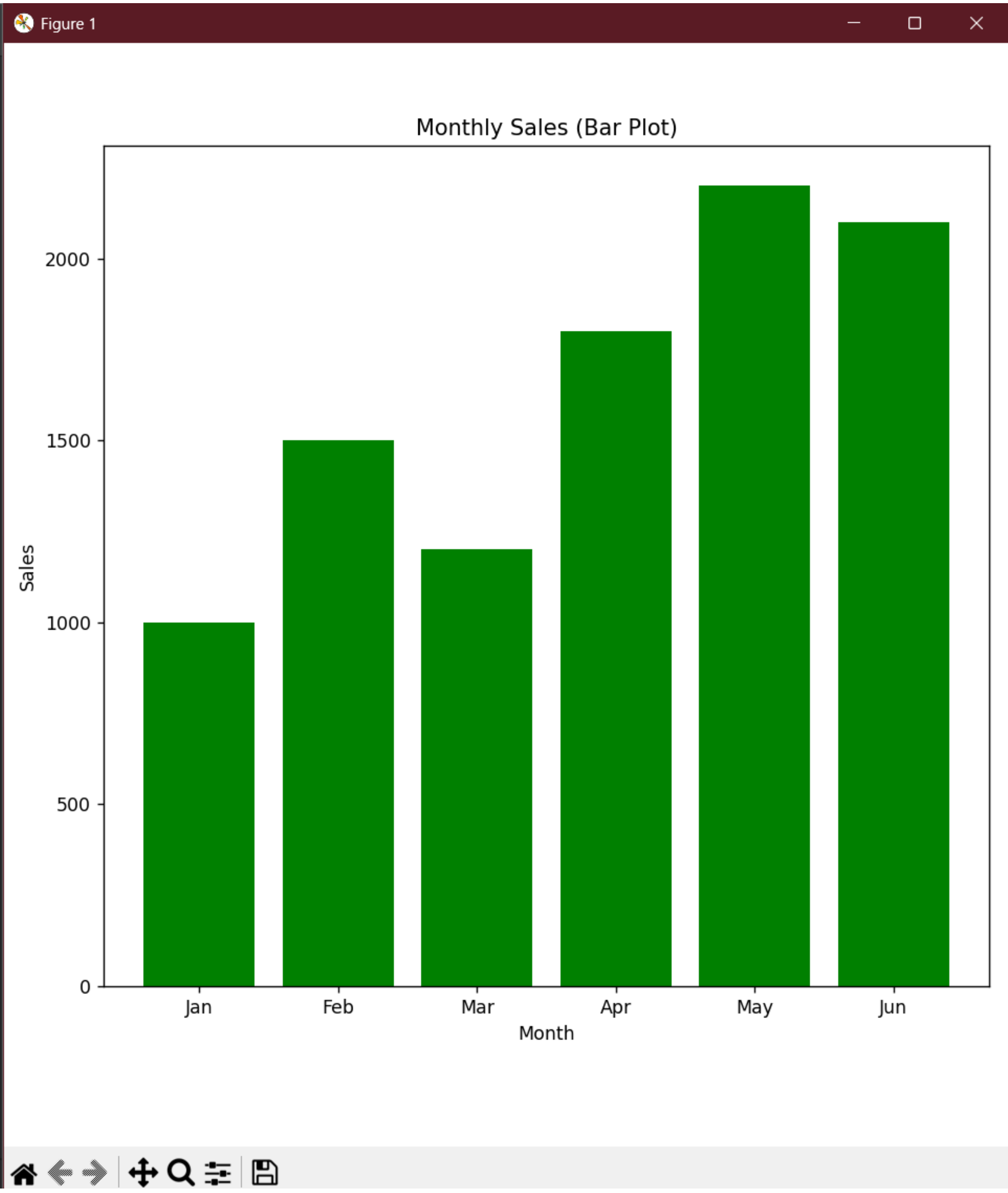


Figure 1
Monthly Sales (Bar Plot)
Sales
Jan
Feb
Mar
Apr
May
Jun
Month