DAY 2 LAB EXPERIMENTS

Name: M SANTHOSH

Reg.no: 192324014

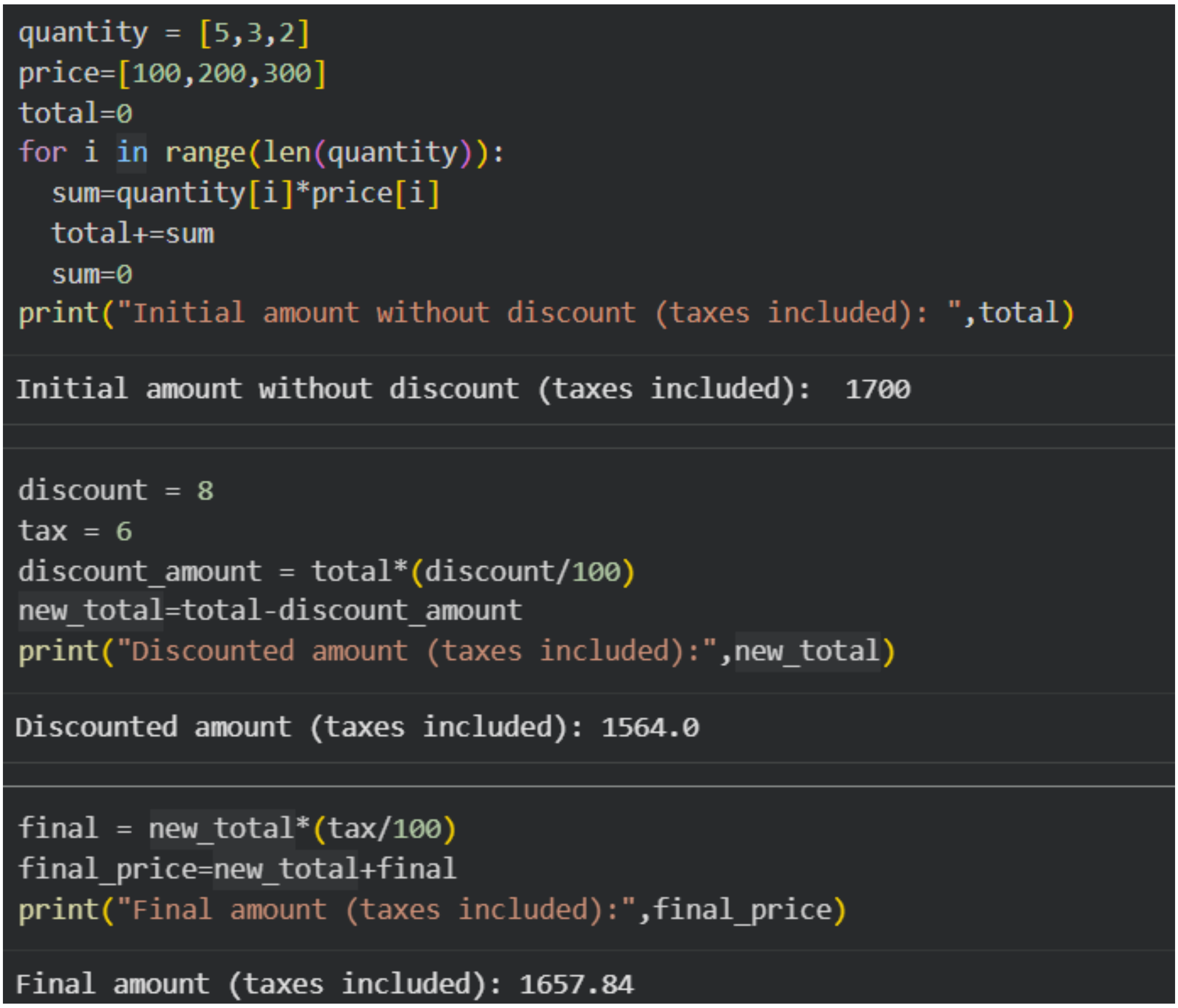
Dept: B.Tech AI & DS

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?

Solution:

```
quantity = [5,3,2]
price=[100,200,300]
total=0
for i in range(len(quantity)):
  sum=quantity[i]*price[i]
  total+=sum
  sum=0
print("Initial amount without discount (taxes included): ",total)
discount = 8
tax = 6
discount_amount = total*(discount/100)
new_total=total-discount_amount
print("Discounted amount (taxes included):",new_total)
final = new_total*(tax/100)
final_price=new_total+final
print("Final amount (taxes included):",final_price)
```



```
quantity = [5,3,2]
price=[100,200,300]
total=0
for i in range(len(quantity)):
  sum=quantity[i]*price[i]
  total+=sum
  sum=0
print("Initial amount without discount (taxes included): ",total)
```

```
Initial amount without discount (taxes included):  1700
```

```
discount = 8
tax = 6
discount_amount = total*(discount/100)
new_total=total-discount_amount
print("Discounted amount (taxes included):",new_total)
```

```
Discounted amount (taxes included): 1564.0
```

```
final = new_total*(tax/100)
final_price=new_total+final
print("Final amount (taxes included):",final_price)
```

```
Final amount (taxes included): 1657.84
```

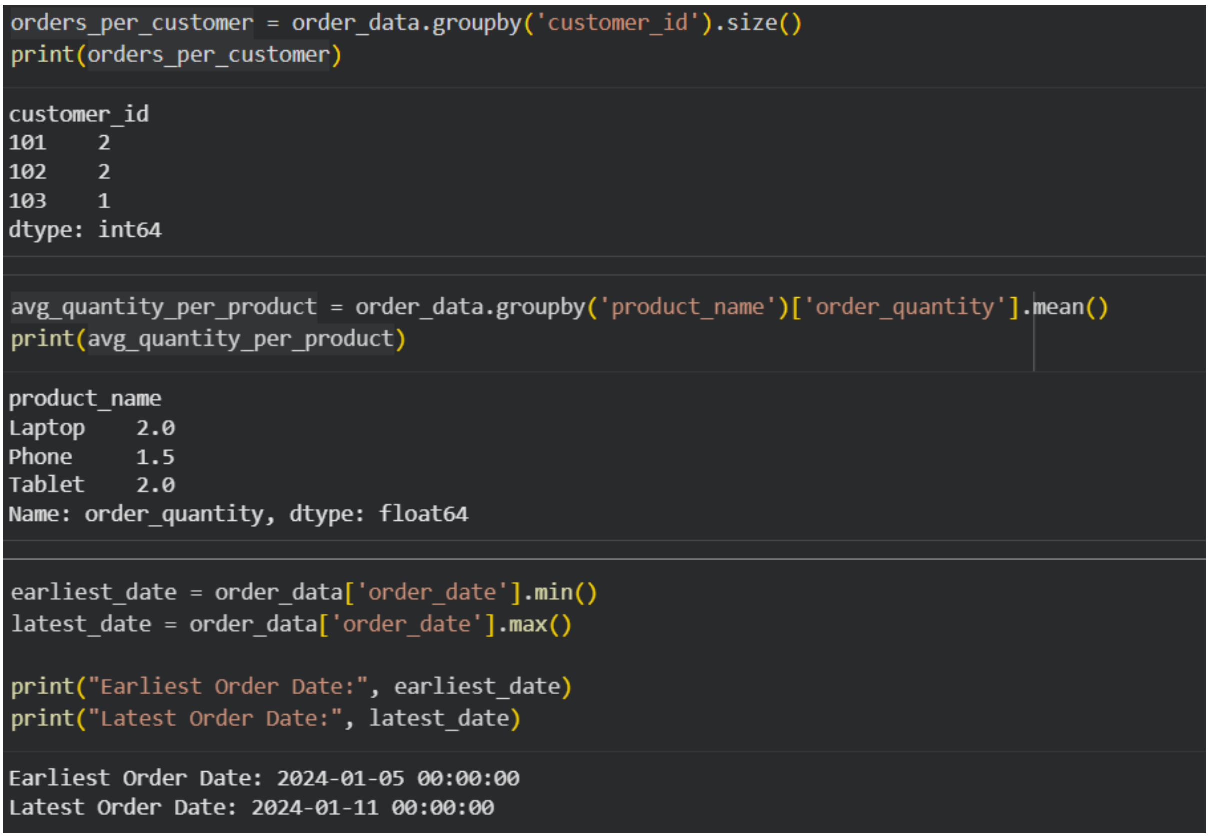
2.**Scenario**: You are working as a data analyst for an e-commerce company. You have been given a dataset containing information about customer orders, stored in a Pandas DataFrame named order_data. The DataFrame has columns for customer ID, order date, product name, and order quantity. Your task is 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.

Solution:

```
import pandas as pd

order_data = pd.DataFrame({
    'customer_id': [101, 102, 101, 103, 102],
    'order_date': pd.to_datetime([
        '2024-01-05', '2024-01-07', '2024-01-9',
        '2024-01-10', '2024-01-11'
    ]),
    'product_name': ['Laptop', 'Phone', 'Phone', 'Laptop', 'Tablet'],
    'order_quantity': [1, 2, 1, 3, 2]
})
orders_per_customer = order_data.groupby('customer_id').size()
print(orders_per_customer)
avg_quantity_per_product = order_data.groupby('product_name')['order_quantity'].mean()
print(avg_quantity_per_product)
earliest_date = order_data['order_date'].min()
latest_date = order_data['order_date'].max()

print("Earliest Order Date:", earliest_date)
print("Latest Order Date:", latest_date)
```



```python
orders_per_customer = order_data.groupby('customer_id').size()
print(orders_per_customer)
```

```
customer_id
101    2
102    2
103    1
dtype: int64
```

```python
avg_quantity_per_product = order_data.groupby('product_name')['order_quantity'].mean()
print(avg_quantity_per_product)
```

```
product_name
Laptop    2.0
Phone     1.5
Tablet    2.0
Name: order_quantity, dtype: float64
```

```python
earliest_date = order_data['order_date'].min()
latest_date = order_data['order_date'].max()

print("Earliest Order Date:", earliest_date)
print("Latest Order Date:", latest_date)
```

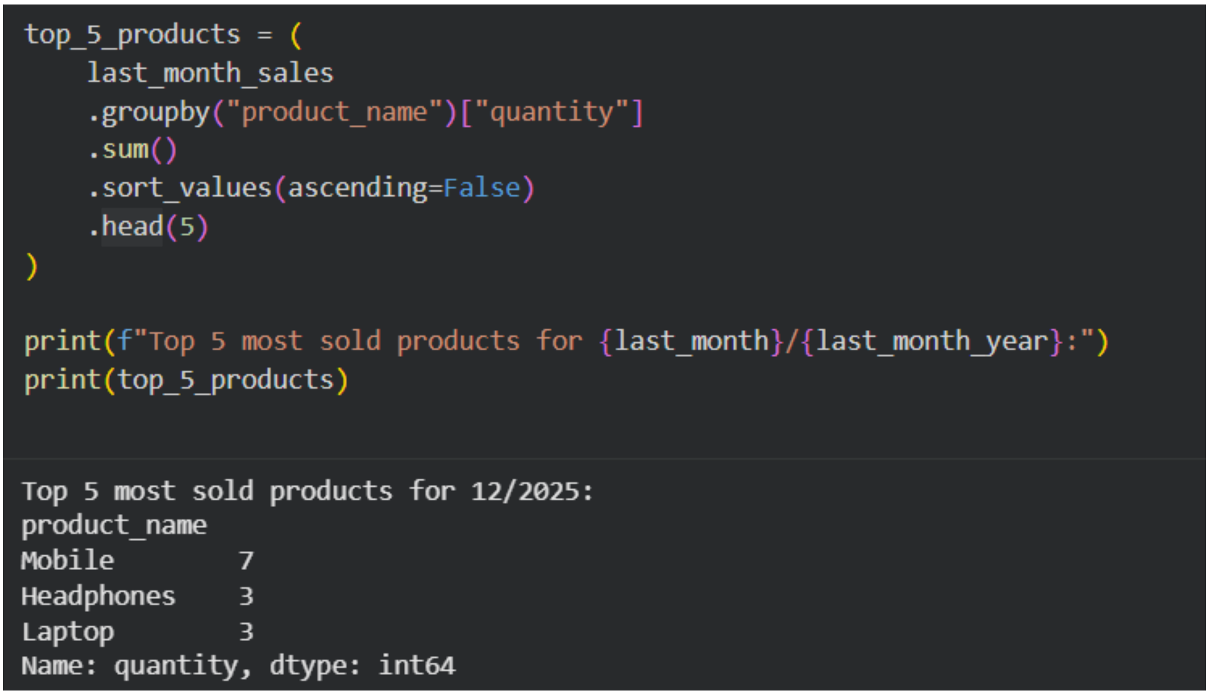
```
Earliest Order Date: 2024-01-05 00:00:00
Latest Order Date: 2024-01-11 00:00:00
```

3. **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?

Solution:

```
import pandas as pd
from datetime import datetime

data = {
    "product_name": [
        "Laptop", "Mobile", "Laptop", "Headphones", "Mobile",
        "Keyboard", "Laptop", "Mouse", "Mobile", "Headphones"
    ],
    "quantity": [2, 5, 1, 3, 2, 4, 3, 6, 1, 2],
    "purchase_date": [
        "2025-12-15", "2025-12-20", "2025-12-25", "2025-12-28", "2025-12-30",
        "2026-01-02", "2026-01-05", "2026-01-06", "2026-01-07", "2026-01-08"
    ]
}
df = pd.DataFrame(data)
df["purchase_date"] = pd.to_datetime(df["purchase_date"])

today = datetime.now()

if today.month == 1:
    last_month = 12
    last_month_year = today.year - 1
else:
    last_month = today.month - 1
    last_month_year = today.year

last_month_sales = df[
    (df["purchase_date"].dt.month == last_month) &
    (df["purchase_date"].dt.year == last_month_year)
]
top_5_products = (
    last_month_sales
    .groupby("product_name")["quantity"]
    .sum()
    .sort_values(ascending=False)
    .head(5)
)

print(f"Top 5 most sold products for {last_month}/{last_month_year}:")
print(top_5_products)
```



```
top_5_products = (
    last_month_sales
    .groupby("product_name")["quantity"]
    .sum()
    .sort_values(ascending=False)
    .head(5)
)

print(f"Top 5 most sold products for {last_month}/{last_month_year}:")
print(top_5_products)
```

```
Top 5 most sold products for 12/2025:
product_name
Mobile        7
Headphones    3
Laptop        3
Name: quantity, dtype: int64
```

4. **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

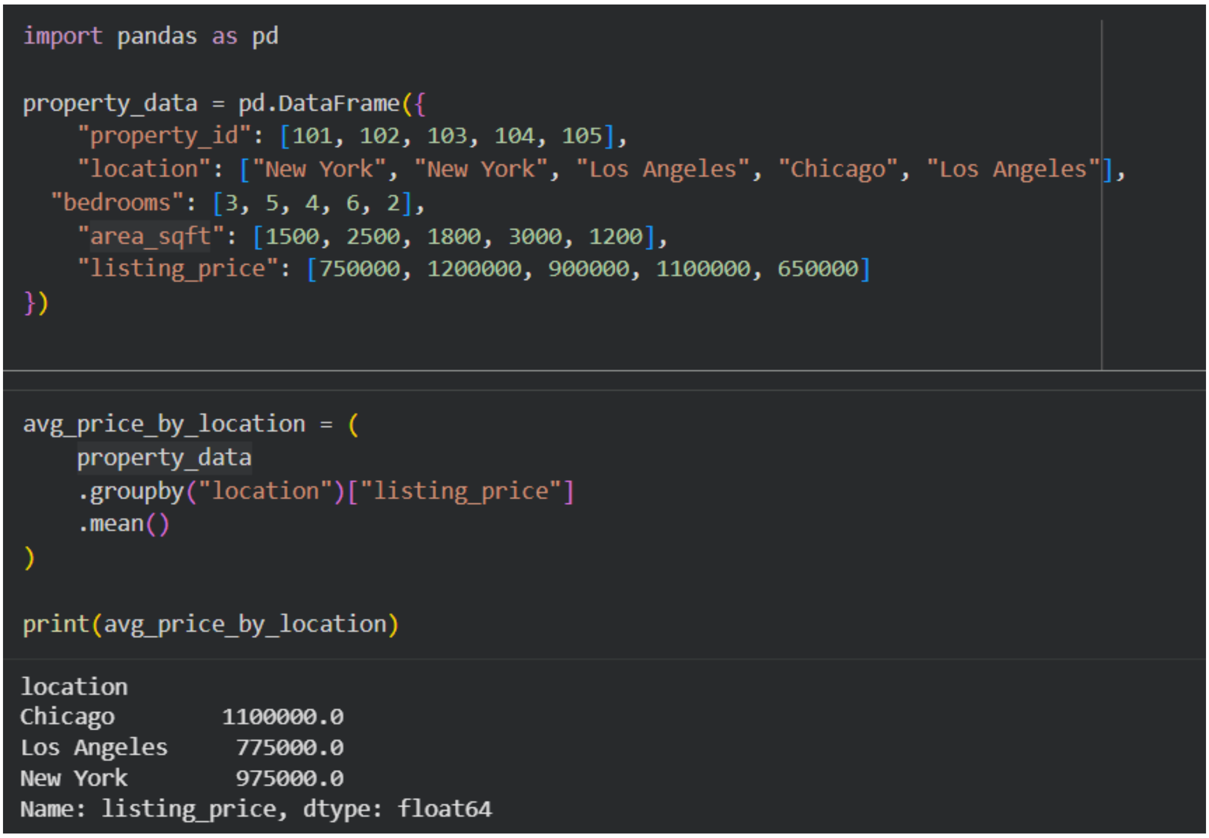
property_data. The DataFrame has columns for property ID, location, number of bedrooms, area in square feet, and listing price. Your task is to analyze the data and answer specific questions about the properties.

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.

Solution:

```
import pandas as pd
property_data = pd.DataFrame({
    "property_id": [101, 102, 103, 104, 105],
    "location": ["New York", "New York", "Los Angeles", "Chicago", "Los Angeles"],
  "bedrooms": [3, 5, 4, 6, 2],
    "area_sqft": [1500, 2500, 1800, 3000, 1200],
    "listing_price": [750000, 1200000, 900000, 1100000, 650000]
})
avg_price_by_location = (
    property_data
    .groupby("location")["listing_price"]
    .mean()
)
print(avg_price_by_location)
```



```python
import pandas as pd

property_data = pd.DataFrame({
    "property_id": [101, 102, 103, 104, 105],
    "location": ["New York", "New York", "Los Angeles", "Chicago", "Los Angeles"],
  "bedrooms": [3, 5, 4, 6, 2],
    "area_sqft": [1500, 2500, 1800, 3000, 1200],
    "listing_price": [750000, 1200000, 900000, 1100000, 650000]
})
```

```python
avg_price_by_location = (
    property_data
    .groupby("location")["listing_price"]
    .mean()
)

print(avg_price_by_location)
```

```
location
Chicago        1100000.0
Los Angeles     775000.0
New York        975000.0
Name: listing_price, dtype: float64
```

5. **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.

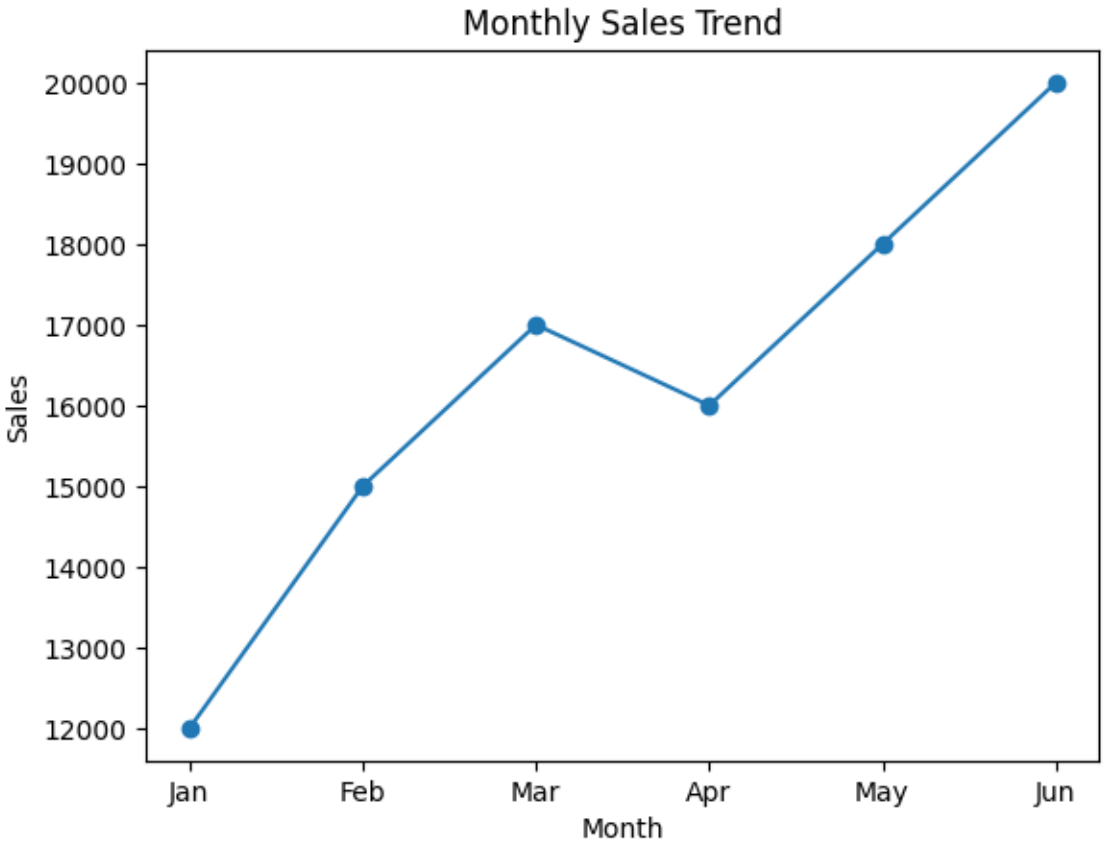
Question:

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

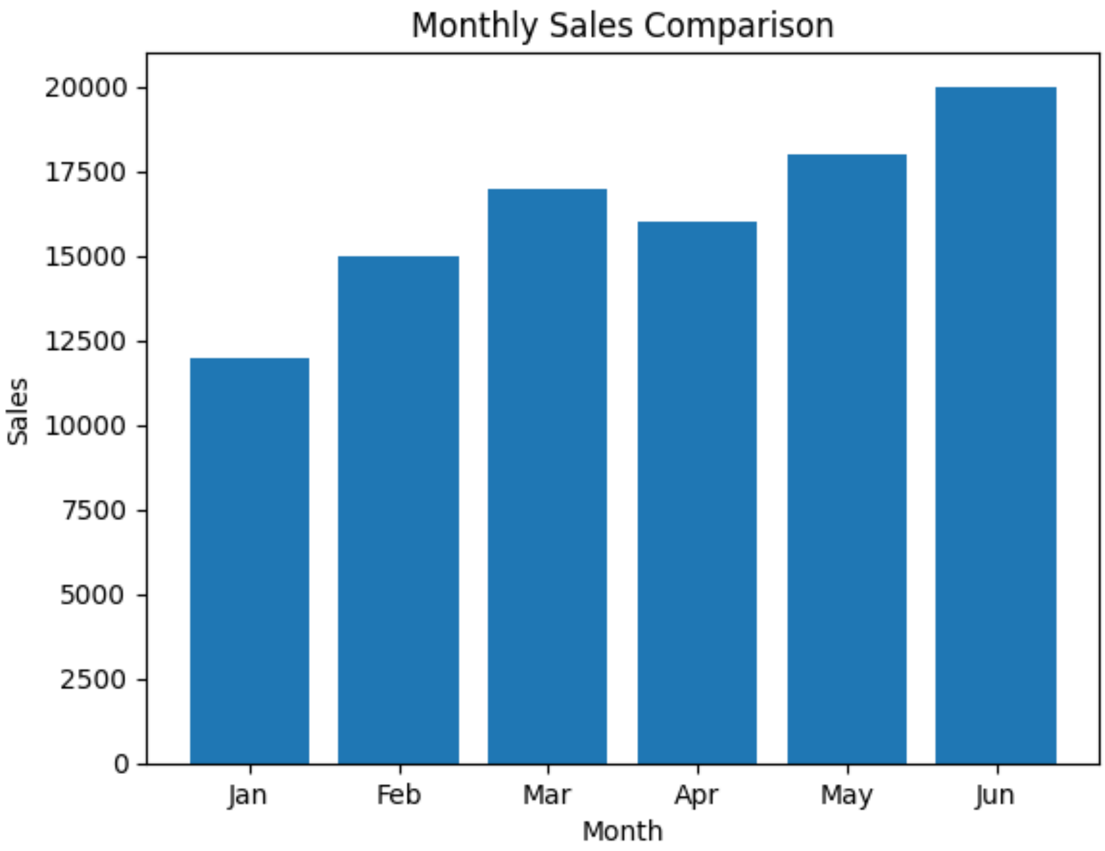
2: How would you develop a Python program to create a bar plot of the monthly sales data?

Solution:

```
import matplotlib.pyplot as plt

months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun"]

sales = [12000, 15000, 17000, 16000, 18000, 20000]

plt.plot(months, sales, marker='o')

plt.xlabel("Month")

plt.ylabel("Sales")

plt.title("Monthly Sales Trend")

plt.show()
```



```
plt.figure()

plt.bar(months, sales)

plt.xlabel("Month")

plt.ylabel("Sales")

plt.title("Monthly Sales Comparison")

plt.show()
```


Monthly Sales Comparison
Sales
Jan
Feb
Mar
Apr
May
Jun
Month