DSA0410 – Fundamentals of Data Science

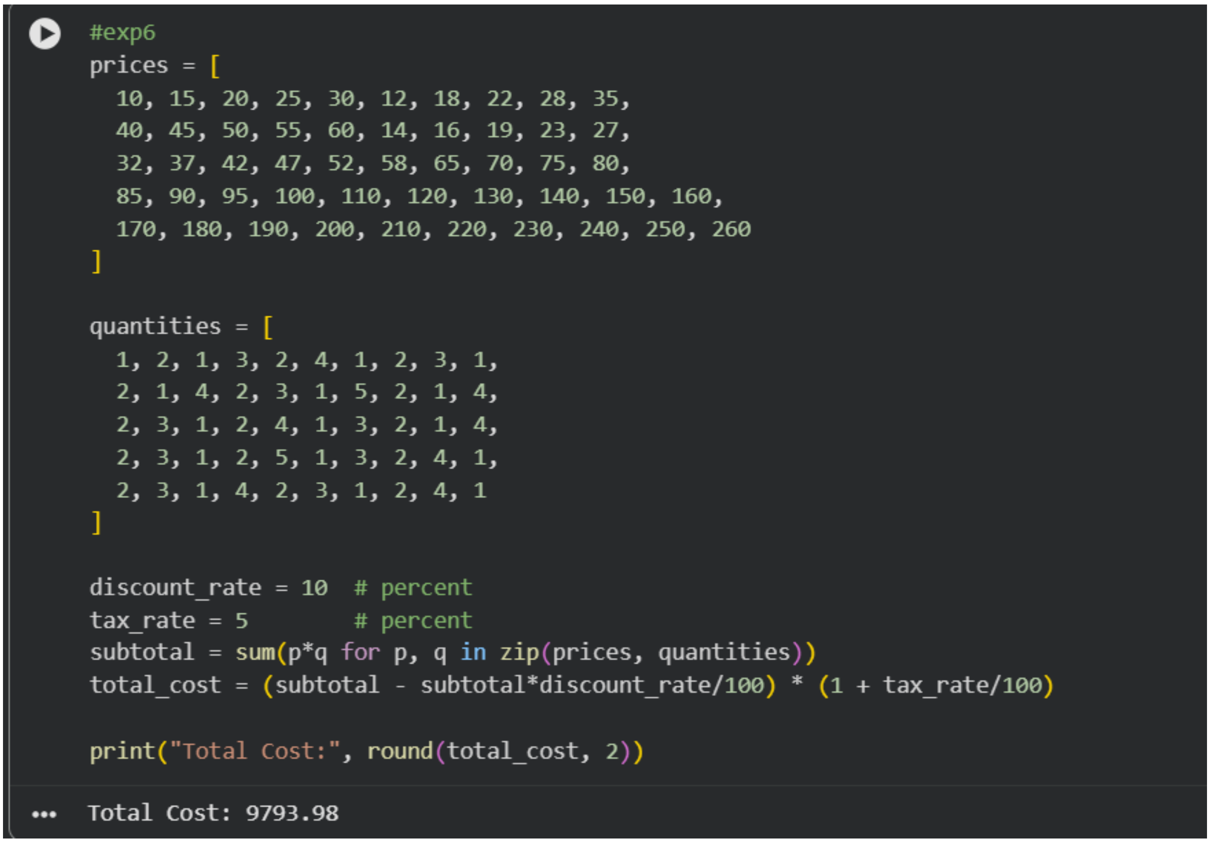
Day - 2

Name : Abinesh H

Reg.no.: 192424387

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?



```
#exp6
prices = [
  10, 15, 20, 25, 30, 12, 18, 22, 28, 35,
  40, 45, 50, 55, 60, 14, 16, 19, 23, 27,
  32, 37, 42, 47, 52, 58, 65, 70, 75, 80,
  85, 90, 95, 100, 110, 120, 130, 140, 150, 160,
  170, 180, 190, 200, 210, 220, 230, 240, 250, 260
]

quantities = [
  1, 2, 1, 3, 2, 4, 1, 2, 3, 1,
  2, 1, 4, 2, 3, 1, 5, 2, 1, 4,
  2, 3, 1, 2, 4, 1, 3, 2, 1, 4,
  2, 3, 1, 2, 5, 1, 3, 2, 4, 1,
  2, 3, 1, 4, 2, 3, 1, 2, 4, 1
]

discount_rate = 10  # percent
tax_rate = 5        # percent
subtotal = sum(p*q for p, q in zip(prices, quantities))
total_cost = (subtotal - subtotal*discount_rate/100) * (1 + tax_rate/100)

print("Total Cost:", round(total_cost, 2))
```

```
Total Cost: 9793.98
```

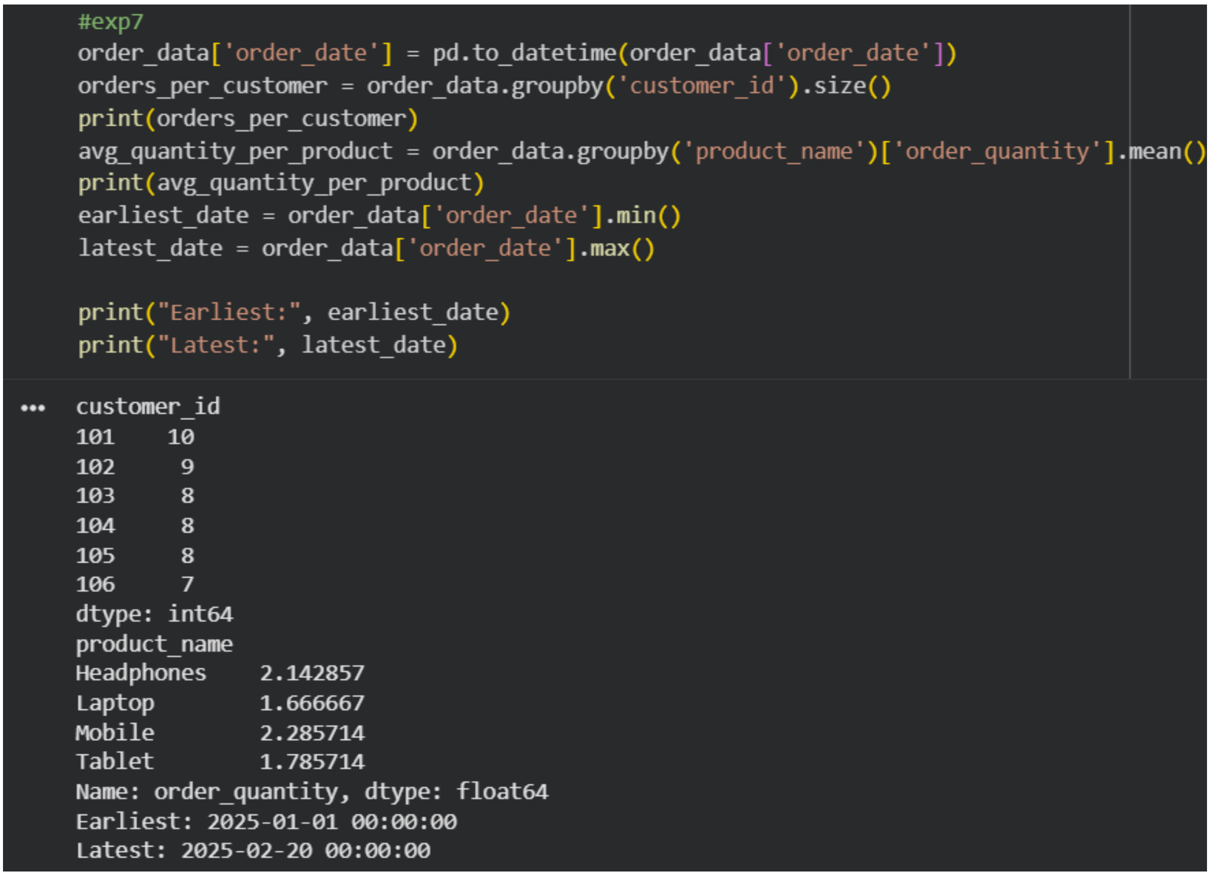
7. Scenario: You are working as a data analyst for an e-commerce company. You have been given a dataset containing information about customer orders, stored in a Pandas DataFrame named order_data. The DataFrame has columns for customer ID, order date, product name, and order quantity. Your task is to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

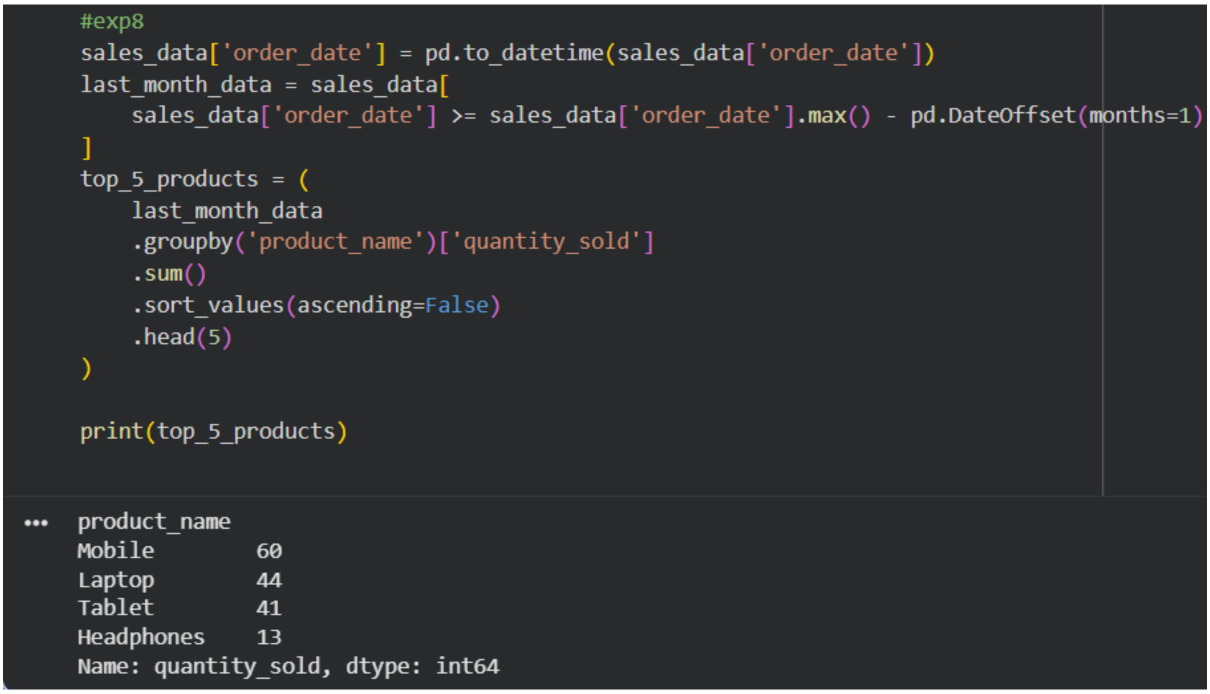


```
#exp7
order_data['order_date'] = pd.to_datetime(order_data['order_date'])
orders_per_customer = order_data.groupby('customer_id').size()
print(orders_per_customer)
avg_quantity_per_product = order_data.groupby('product_name')['order_quantity'].mean()
print(avg_quantity_per_product)
earliest_date = order_data['order_date'].min()
latest_date = order_data['order_date'].max()

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

```
customer_id
101    10
102     9
103     8
104     8
105     8
106     7
dtype: int64
product_name
Headphones    2.142857
Laptop        1.666667
Mobile        2.285714
Tablet        1.785714
Name: order_quantity, dtype: float64
Earliest: 2025-01-01 00:00:00
Latest: 2025-02-20 00:00:00
```

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?



```
#exp8
sales_data['order_date'] = pd.to_datetime(sales_data['order_date'])
last_month_data = sales_data[
    sales_data['order_date'] >= sales_data['order_date'].max() - pd.DateOffset(months=1)
]
top_5_products = (
    last_month_data
    .groupby('product_name')['quantity_sold']
    .sum()
    .sort_values(ascending=False)
    .head(5)
)

print(top_5_products)
```

```
product_name
Mobile        60
Laptop        44
Tablet        41
Headphones    13
Name: quantity_sold, dtype: int64
```

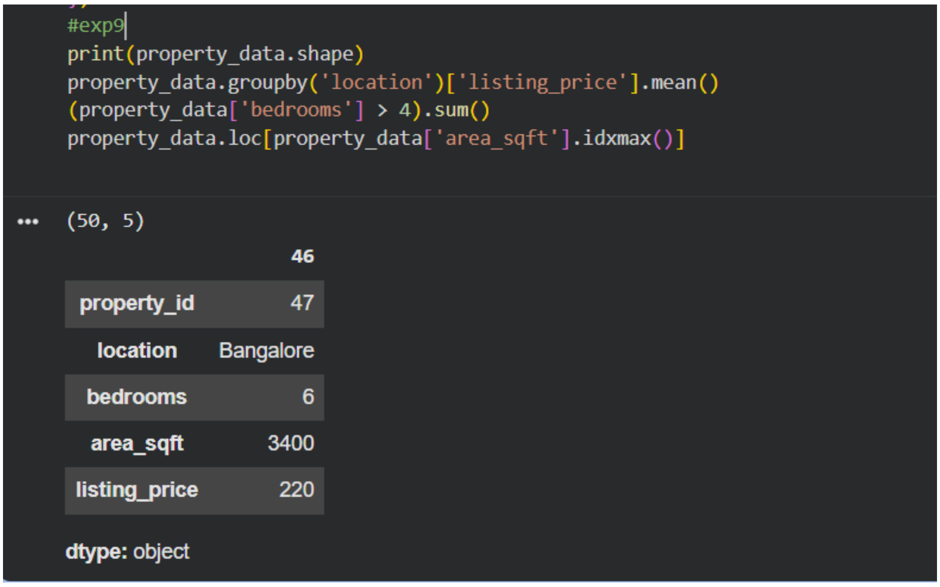
9. Scenario: You work for a real estate agency and have been given a dataset containing information about properties for sale. The dataset is stored in a Pandas DataFrame named property_data. The DataFrame has columns for property ID, location, number of bedrooms, area in square feet, and listing price.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



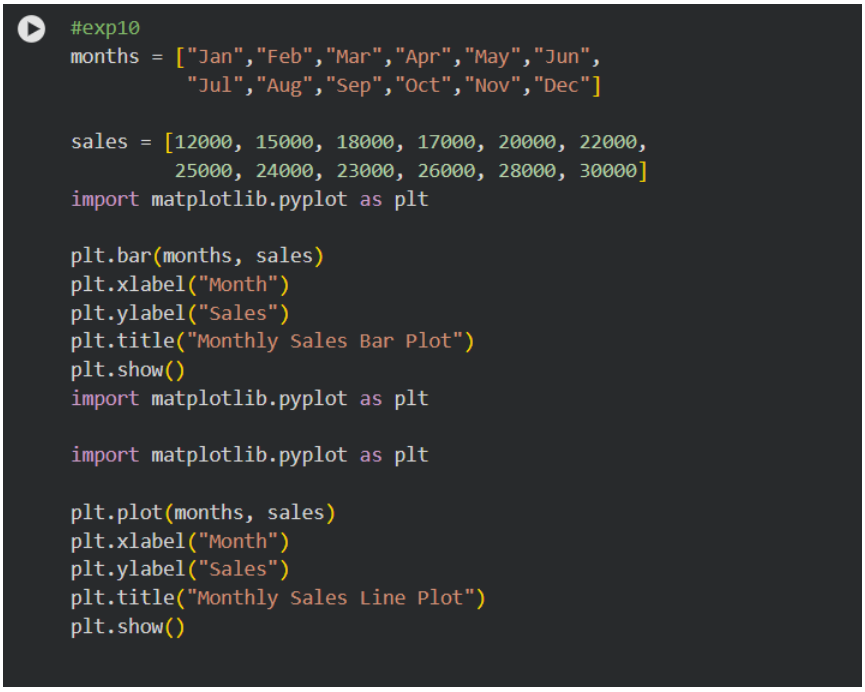
```
#exp9
print(property_data.shape)
property_data.groupby('location')['listing_price'].mean()
(property_data['bedrooms'] > 4).sum()
property_data.loc[property_data['area_sqft'].idxmax()]
```

```
(50, 5)
                       46
property_id            47
location        Bangalore
bedrooms                6
area_sqft            3400
listing_price         220

dtype: object
```

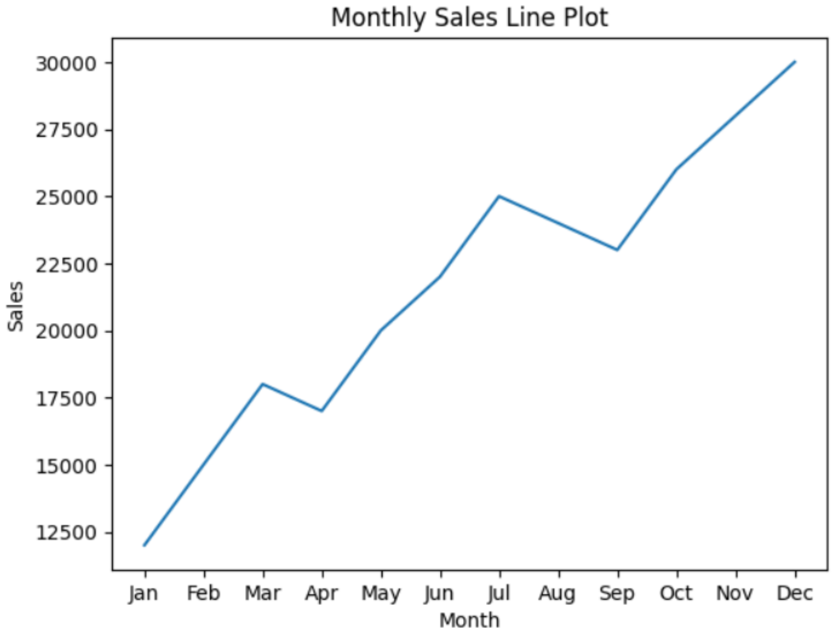
10. Scenario: You are working on a data visualization project and need to create basic plots using Matplotlib. You have a dataset containing the monthly sales data for a company, including the month and corresponding sales values. Your task is to develop a Python program that generates line plots and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

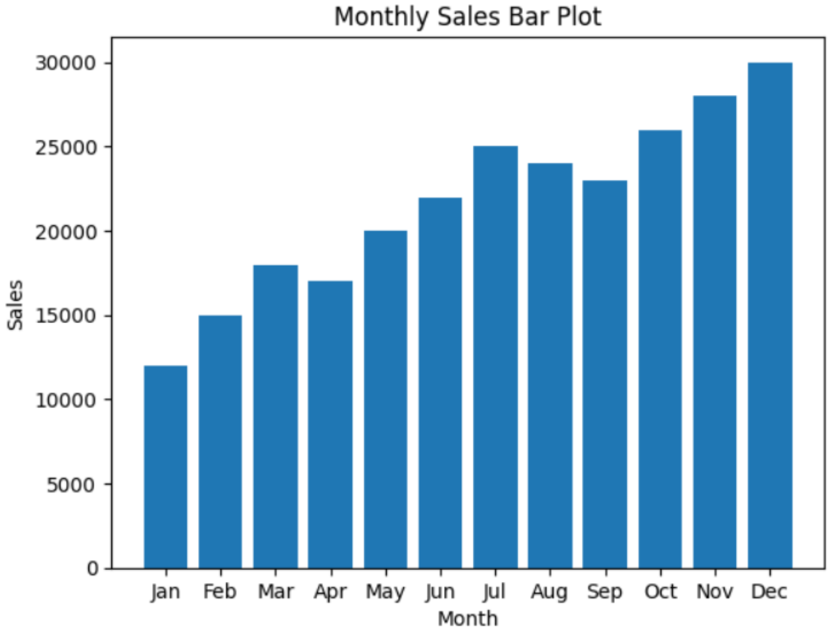
Question:

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?



```
#exp10
months = ["Jan","Feb","Mar","Apr","May","Jun",
          "Jul","Aug","Sep","Oct","Nov","Dec"]

sales = [12000, 15000, 18000, 17000, 20000, 22000,
         25000, 24000, 23000, 26000, 28000, 30000]
import matplotlib.pyplot as plt

plt.bar(months, sales)
plt.xlabel("Month")
plt.ylabel("Sales")
plt.title("Monthly Sales Bar Plot")
plt.show()
import matplotlib.pyplot as plt

import matplotlib.pyplot as plt

plt.plot(months, sales)
plt.xlabel("Month")
plt.ylabel("Sales")
plt.title("Monthly Sales Line Plot")
plt.show()
```


Monthly Sales Line Plot
Sales
30000
27500
25000
22500
20000
17500
15000
12500
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Monthly Sales Bar Plot
Sales
30000
25000
20000
15000
10000
5000
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month