DAY-2

DSA0410 – Fundamentals of Data Science

Lab Experiments:

Name: B Harish Balaji

Reg no: 192424386

Slot: D

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

```
import pandas as pd

# Load Excel file
df = pd.read_excel('grocery_purchase_30_items.xlsx')

# Calculate item total
df['Item Total'] = df['Price'] * df['Quantity']

# Subtotal
subtotal = df['Item Total'].sum()

# Assuming same discount & tax for all items
discount_rate = df['Discount (%)'].iloc[0]
```

```
tax_rate = df['Tax (%)'].iloc[0]

# Discount calculation
discount_amount = subtotal * (discount_rate / 100)
after_discount = subtotal - discount_amount

# Tax calculation
tax_amount = after_discount * (tax_rate / 100)
# Final total
final_total = after_discount + tax_amount

print("Item-wise Billing (First 10 Items):")
print(df[['Item Name', 'Price', 'Quantity', 'Item Total']].head(10))

print("\nSummary:")
print(f"Subtotal: ₹{subtotal:.2f}")
print(f"Discount ({discount_rate}%): -₹{discount_amount:.2f}")
print(f"Tax ({tax_rate}%): +₹{tax_amount:.2f}")
print(f"Total Amount Payable: ₹{final_total:.2f}")
```

sample output:

```python
print("Item-wise Billing (First 10 Items):")
print(df[['Item Name', 'Price', 'Quantity', 'Item Total']].head(10))

print("\nSummary:")
print(f"Subtotal: ₹{subtotal:.2f}")
print(f"Discount ({discount_rate}%): -₹{discount_amount:.2f}")
print(f"Tax ({tax_rate}%): +₹{tax_amount:.2f}")
print(f"Total Amount Payable: ₹{final_total:.2f}")
```

Choose Files grocery_pur..._items.xlsx

grocery_purchase_30_items.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5772 bytes, last modified: 12/17/2025 - 100% done

```
Saving grocery_purchase_30_items.xlsx to grocery_purchase_30_items.xlsx
Item-wise Billing (First 10 Items):
     Item Name  Price  Quantity  Item Total
0         Rice     60         2         120
1  Wheat Flour     45         3         135
2        Sugar     42         2          84
3         Salt     20         1          20
4         Milk     45         4         180
5         Curd     50         2         100
6       Butter    120         1         120
7       Cheese    180         1         180
8         Eggs      6        12          72
9        Bread     30         2          60

Summary:
Subtotal: ₹4032.00
Discount (10%): -₹403.20
Tax (5%): +₹181.44
Total Amount Payable: ₹3810.24
```

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.

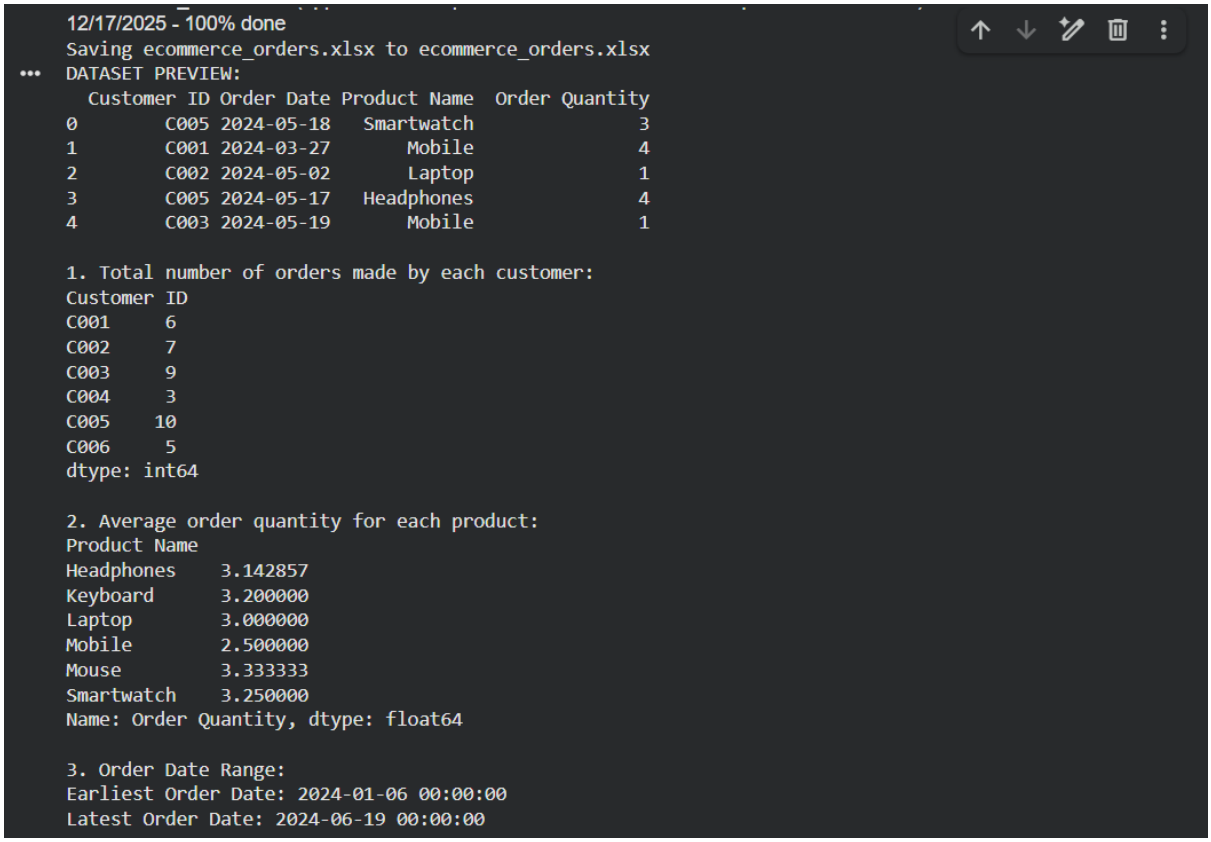
2. The average order quantity for each product.

3. The earliest and latest order dates in the dataset. give me data set in excel sheet and colab code

```
import pandas as pd

# Load the Excel dataset
order_data = pd.read_excel('ecommerce_orders.xlsx')

# ---------- 1. Total number of orders by each customer ----------
orders_per_customer = order_data.groupby('Customer ID').size()

# ---------- 2. Average order quantity for each product ----------
```

```
avg_quantity_per_product = order_data.groupby('Product
Name')['Order Quantity'].mean()

# ---------- 3. Earliest and latest order dates ----------
earliest_date = order_data['Order Date'].min()
latest_date = order_data['Order Date'].max()

# ---------- Display Results ----------
print("DATASET PREVIEW:")
print(order_data.head())

print("\n1. Total number of orders made by each customer:")
print(orders_per_customer)

print("\n2. Average order quantity for each product:")
print(avg_quantity_per_product)

print("\n3. Order Date Range:")
print(f"Earliest Order Date: {earliest_date}")
print(f"Latest Order Date: {latest_date}")
```

sample output:



```
12/17/2025 - 100% done
Saving ecommerce_orders.xlsx to ecommerce_orders.xlsx
DATASET PREVIEW:
  Customer ID Order Date Product Name  Order Quantity
0        C005 2024-05-18   Smartwatch               3
1        C001 2024-03-27       Mobile               4
2        C002 2024-05-02       Laptop               1
3        C005 2024-05-17   Headphones               4
4        C003 2024-05-19       Mobile               1

1. Total number of orders made by each customer:
Customer ID
C001     6
C002     7
C003     9
C004     3
C005    10
C006     5
dtype: int64

2. Average order quantity for each product:
Product Name
Headphones    3.142857
Keyboard      3.200000
Laptop        3.000000
Mobile        2.500000
Mouse         3.333333
Smartwatch    3.250000
Name: Order Quantity, dtype: float64

3. Order Date Range:
Earliest Order Date: 2024-01-06 00:00:00
Latest Order Date: 2024-06-19 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?

import pandas as pd

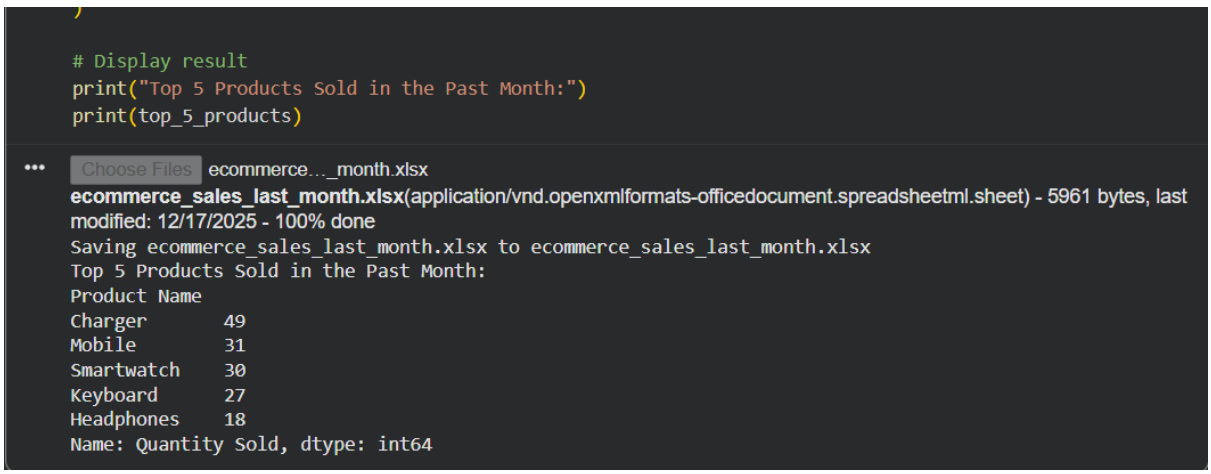
Load modified sales dataset

sales_data = pd.read_excel('ecommerce_sales_last_month.xlsx')

Convert Sale Date to datetime

```
sales_data['Sale Date'] = pd.to_datetime(sales_data['Sale Date'])

# Find top 5 products sold in the past month
top_5_products = (
    sales_data
    .groupby('Product Name')['Quantity Sold']
    .sum()
    .sort_values(ascending=False)
    .head(5)
)

# Display result
print("Top 5 Products Sold in the Past Month:")
print(top_5_products)
```

sample output:



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.

```
import pandas as pd

# Load dataset
property_data = pd.read_excel('real_estate_properties.xlsx')

# 1. Average listing price of properties in each location
avg_price_by_location = (
    property_data
    .groupby('Location')['Listing Price']
    .mean()
)
```

```
# 2. Number of properties with more than four bedrooms
properties_more_than_4_bedrooms = (
    property_data[property_data['Bedrooms'] > 4]
    .shape[0]
)

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

# Display results
print("1. Average listing price by location:")
print(avg_price_by_location)

print("\n2. Number of properties with more than 4 bedrooms:")
print(properties_more_than_4_bedrooms)

print("\n3. Property with the largest area:")
print(largest_area_property)
```

sample output:

```
print("\n3. Property with the largest area:")
print(largest_area_property)
```

```
Choose Files real_estate...operties.xlsx
real_estate_properties.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 6464 bytes, last modified: 12/17/2025 - 100% done
Saving real_estate_properties.xlsx to real_estate_properties.xlsx
1. Average listing price by location:
Location
Bangalore    2.403929e+07
Chennai      1.490584e+07
Delhi        1.853640e+07
Hyderabad    2.402568e+07
Mumbai       1.783077e+07
Pune         1.892098e+07
Name: Listing Price, dtype: float64

2. Number of properties with more than 4 bedrooms:
15

3. Property with the largest area:
Property ID           P002
Location         Bangalore
Bedrooms                 4
Area (sqft)           4182
Listing Price     35204076
Name: 1, 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?

```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
sales_data = pd.read_excel('monthly_sales_50_records.xlsx')

# ---------- 1. Line Plot ----------
plt.figure()
```

```
plt.plot(sales_data['Month'], sales_data['Sales'])
plt.xlabel('Month')
plt.ylabel('Sales')
plt.title('Monthly Sales Line Plot')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

# ---------- 2. Bar Plot ----------
plt.figure()
plt.bar(sales_data['Month'], sales_data['Sales'])
plt.xlabel('Month')
plt.ylabel('Sales')
plt.title('Monthly Sales Bar Plot')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

sample output:

```
plt.show()
```

monthly_sales_50_records.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5655 bytes, last modified: 12/17/2025 - 100% done

```
Saving monthly_sales_50_records.xlsx to monthly_sales_50_records.xlsx
```