

TANX.FI

TASK SUBMISSION

PROBLEM STATEMENT:

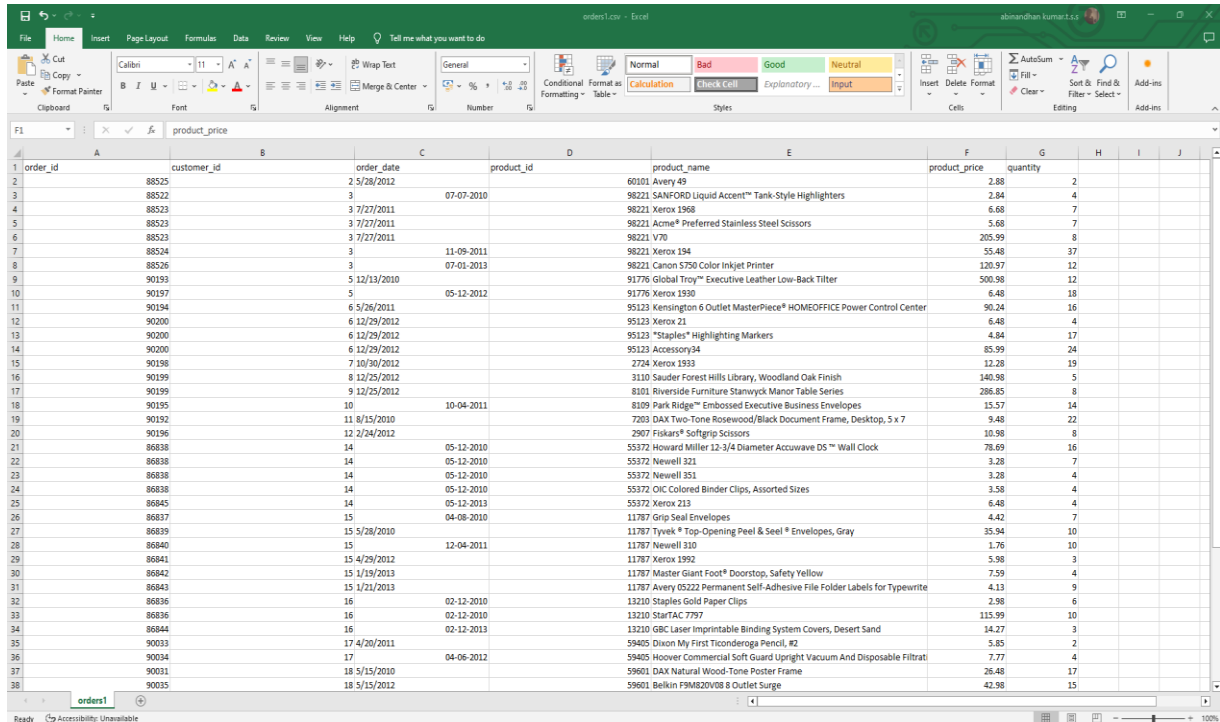
You have been given a dataset of customer orders from an online store. The data is in a CSV file `orders.csv` with the following columns:

- **Order id:** unique identifier for each order
- **Customer id:** unique identifier for each customer
- **Order date:** date when the order was placed
- **Product id:** unique identifier for each product
- **Product name:** name of the product
- **Product price:** price of the product
- **quantity:** quantity of the product ordered

Your task is to write a Python program that reads the data from the CSV file and performs

the following tasks:

Dataset:



order_id	customer_id	order_date	product_id	product_name	product_price	quantity
1	88525	2 5/28/2012		60101 Avery 49	2.88	2
2	88522	3 07-07-2010		98221 SANFORD Liquid Accent™ Tank-Style Highlighters	2.84	4
3	88523	3 7/27/2011		98221 Xerox 1968	6.68	7
4	88523	3 7/27/2011		98221 Acme® Preferred Stainless Steel Scissors	5.68	7
5	88523	3 7/27/2011		98221 V70	205.99	8
6	88524	3	11-09-2011	98221 Xerox 194	55.48	37
7	88526	3	07-01-2013	98221 Canon S750 Color Inkjet Printer	120.97	12
8	90193	5 12/13/2010		91776 Global Troy™ Executive Leather Low-Back Tilter	500.98	12
9	90197	5	05-12-2012	91776 Xerox 1930	6.48	18
10	90194	6 5/26/2011		95123 Kensington 6 Outlet MasterPiece® HOMEOFFICE Power Control Center	90.24	16
11	90200	6 12/29/2012		95123 Xerox 21	6.48	4
12	90200	6 12/29/2012		95123 "Staples" Highlighting Markers	4.84	17
13	90200	6 12/29/2012		95123 Accessory34	85.99	24
14	90198	7 10/30/2012		2724 Xerox 1938	12.28	19
15	90199	8 12/25/2012		3110 Sauder Forest Hills Library, Woodland Oak Finish	140.98	5
16	90199	9 12/25/2012		8101 Riverside Furniture Stanwyck Manor Table Series	286.85	8
17	90195	10	10-04-2011	8109 Park Ridge™ Embossed Executive Business Envelopes	15.57	14
18	90192	11 8/15/2010		7203 DAX Two-Tone Rosewood/Black Document Frame, Desktop, 5 x 7	9.48	22
19	90196	12 2/24/2012		2907 Fiskars® Softgrip Scissors	10.98	8
20	86838	14	05-12-2010	55372 Howard Miller 12-3/4 Diameter Accuwave DS™ Wall Clock	78.69	16
21	86838	14	05-12-2010	55372 Newell 321	3.28	7
22	86838	14	05-12-2010	55372 Newell 351	3.28	4
23	86838	14	05-12-2010	55372 OIC Colored Binder Clips, Assorted Sizes	3.58	4
24	86945	14	05-12-2013	55372 Xerox 213	6.48	4
25	86837	15	04-08-2010	11787 Grip Seal Envelopes	4.42	7
26	86839	15 5/26/2010		11787 Tyvek® Top-Opening Peel & Seal® Envelopes, Gray	35.94	10
27	86840	15	12-04-2011	11787 Newell 310	1.76	10
28	86841	15 4/29/2012		11787 Xerox 1992	5.98	3
29	86842	15 1/19/2013		11787 Master Giant Foot® Doorstop, Safety Yellow	7.59	4
30	86843	15 1/21/2013		11787 Avery 05222 Permanent Self-Adhesive File Folder Labels for Typewrite	4.13	9
31	86836	16	02-12-2010	13210 Staples Gold Paper Clips	3.98	6
32	86836	16	02-12-2010	13210 StarTAC 7797	115.99	10
33	86844	16	02-12-2013	13210 GBC Laser Imprintable Binding System Covers, Desert Sand	14.27	3
34	90033	17 4/20/2011		59405 Dixon My First Ticonderoga Pencil, #2	5.85	2
35	90034	17	04-06-2012	59405 Hoover Commercial Soft Guard Upright Vacuum And Disposable Filtrat	7.77	4
36	90031	18 5/15/2010		59601 DAX Natural Wood-Tone Poster Frame	26.48	17
37	90035	18 5/15/2012		59601 Belkin F9M820V08 8 Outlet Surge	42.98	15



```
# Importing the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]

```
# Loading the dataset
df = pd.read_csv('/content/orders1.csv')
```

STEP 1:

IMPORTING NECESSARY LIBRARIES:

- **PANDAS** – Pandas is a powerful data manipulation and analysis library in Python.
- **MATPLOTLIB** - Matplotlib is a plotting library in Python that produces high-quality figures and plots.
- **SEABORN** - provides a high-level interface for drawing attractive and informative statistical graphics.

1. Compute the total revenue generated by the online store for each month in the dataset.

1. Compute the total revenue generated by the online store for each month in the dataset.

```
✓ 0s ▶ # Calculate revenue for each order
      df['revenue'] = df['quantity'] * df['product_price']

      # Aggregate revenue by month
      revenue_by_month = df.groupby(df['order_date'].dt.to_period('M'))['revenue'].sum()

      # Display total revenue generated by the online store for each month
      print(revenue_by_month)
```

```
➡ order_date
2010-02    1177.78
2010-04     30.94
2010-05   5418.71
2010-07     11.36
2010-08    208.56
2010-12   6011.76
2011-03     30.98
2011-04     64.35
2011-05   1443.84
2011-07   1734.44
2011-10    217.98
2011-11   2052.76
2011-12     17.60
2012-02     87.84
2012-04     49.02
2012-05   1695.14
2012-10    233.32
2012-12   5171.66
2013-01     67.53
2013-02     42.81
2013-03     30.98
2013-05     90.92
2013-07   1451.64
2013-10   2654.57
Freq: M, Name: revenue, dtype: float64
```

2. Compute the total revenue generated by each product in the dataset.

```
✓ 0s ▶ # Calculating the revenue for each order
df['total_revenue'] = df['quantity'] * df['product_price']

# Aggregate revenue by product
revenue_by_product = df.groupby('product_name')['total_revenue'].sum().sort_values(ascending=False)

# Displaying total revenue generated by each product
print(revenue_by_product)
```

product_name	
Global Troy™ Executive Leather Low-Back Tilter	6011.76
Office Star Flex Back Scooter Chair with White Frame	2441.56
DAX Natural Wood-Tone Poster Frame	2303.76
Riverside Furniture Stanwyck Manor Table Series	2294.80
Accessory34	2063.76
Xerox 194	2052.76
V70	1647.92
Canon S750 Color Inkjet Printer	1451.64
Kensington 6 Outlet MasterPiece® HOMEOFFICE Power Control Center	1443.84
Howard Miller 12-3/4 Diameter Accuwave DS™ Wall Clock	1259.04
StarTAC 7797	1159.90
Hewlett Packard 6S Scientific Calculator	1156.11
Space Solutions Commercial Steel Shelving	840.45
Sauder Forest Hills Library, Woodland Oak Finish	704.90
Belkin F9M820V08 8 Outlet Surge	644.70
Tyvek® Top-Opening Peel & Seal® Envelopes, Gray	359.40
Advantus Plastic Paper Clips	355.00
Xerox 1933	233.32
Park Ridge™ Embossed Executive Business Envelopes	217.98
Avery 485	213.01
DAX Two-Tone Rosewood/Black Document Frame, Desktop, 5 x 7	208.56
Xerox 1930	116.64
Fiskars® Softgrip Scissors	87.84
Staples Highlighting Markers	82.28
Fellowes Twister Kit, Gray/Clear, 3/pkg	72.36
Dixon My First Ticonderoga Pencil, #2	64.35
Xerox 197	61.96
Xerox 1968	46.76
GBC Laser Imprintable Binding System Covers, Desert Sand	42.81
Acme® Preferred Stainless Steel Scissors	39.76
Avery 05222 Permanent Self-Adhesive File Folder Labels for Typewriters, on Rolls, White, 250/Roll	37.17
Hoover Commercial Soft Guard Upright Vacuum And Disposable Filtration Bags	31.08
Grip Seal Envelopes	30.94
Master Giant Foot® Doorstop, Safety Yellow	30.36
Xerox 21	25.92
Xerox 213	25.92
Newell 321	22.96
Xerox 1992	17.94
Staples Gold Paper Clips	17.88
Newell 310	17.60
Anderson Hickey Conga Table Tops & Accessories	15.23
OIC Colored Binder Clips, Assorted Sizes	14.32
Newell 351	13.12
SANFORD Liquid Accent™ Tank-Style Highlighters	11.36
Avery 49	5.76
Name: total_revenue, dtype: float64	

3. Compute the total revenue generated by each customer in the dataset.

```
3. Compute the total revenue generated by each customer in the dataset
```

```
[12] # Calculate revenue for each order
      df['total_revenue'] = df['quantity'] * df['product_price']

      # Aggregate revenue by customer
      revenue_by_customer = df.groupby('customer_id')['total_revenue'].sum().sort_values(ascending=False)

      # Display total revenue generated by each customer
      print(revenue_by_customer)
```

```
customer_id
5      6128.40
3      5250.20
18     3886.79
6      3615.80
21     3149.52
9      2294.80
14     1335.36
16     1220.59
20      886.66
8       704.90
15      493.41
19      233.82
7       233.32
10      217.98
11      208.56
12       87.84
17       42.78
2         5.76
Name: total_revenue, dtype: float64
```

4. Identify the top 10 customers by revenue generated.

```
# Calculate revenue for each order
df['total_revenue'] = df['quantity'] * df['product_price']

# Aggregate revenue by customer
revenue_by_customer = df.groupby('customer_id')['total_revenue'].sum().sort_values(ascending=False)

# Select top 10 customers by revenue
top_10_customers = revenue_by_customer.head(10)

# Display top 10 customers by revenue
print(top_10_customers)
```

```
customer_id
5      6128.40
3      5250.20
18     3886.79
6      3615.80
21     3149.52
9      2294.80
14     1335.36
16     1220.59
20      886.66
8       704.90
Name: total_revenue, dtype: float64
```

PROFILES FOR VERIFICATION PURPOSE:

1. GITHUB PROFILE: <https://github.com/abinandhan05/TANX.FI-INFRASTRUCTURE-TASK>
2. LINKEDIN PROFILE: <https://www.linkedin.com/in/t-s-s-abinandhan-kumar/>

THANKS FOR GIVING ME THIS OPPORTUNITY TANX.FI!!!

TANX.FI