EDS ACTIVITY NO.1
[SALES DATASET]

- **Name : Vineet Kaldate**
- **Roll no : CS7-87**
- **PRN : 202401110074**
- **Colab link: https://colab.research.google.com/drive/1gtGBmC5LkEG5E1EjnS66rweyEf6fLTDe?usp=sharing**

```
[1] # Upload CSVfile
    from google.colab import files
    uploaded = files.upload()
```

Choose Files sales_data.csv

- **sales_data.csv**(text/csv) - 103583 bytes, last modified: 4/28/2025 - 100% done

Saving sales_data.csv to sales_data.csv

```
import pandas as pd

# Upload and load your CSV
df = pd.read_csv('/content/sales_data.csv')  # <-- adjust the path if needed

df.head()  # Preview
```

	Product_ID	Sale_Date	Sales_Rep	Region	Sales_Amount	Quantity_Sold	Product_Category	Unit_Cost	Unit_Price	Customer_Type	Discount	Payment_Method	Sales_Channel	Region_and_Sal
0	1052	03-02-2023	Bob	North	5053.97	18	Furniture	152.75	267.22	Returning	0.09	Cash	Online	No
1	1093	21-04-2023	Bob	West	4384.02	17	Furniture	3816.39	4209.44	Returning	0.11	Cash	Retail	We
2	1015	21-09-2023	David	South	4631.23	30	Food	261.56	371.40	Returning	0.20	Bank Transfer	Retail	Soutl
3	1072	24-08-2023	Bob	South	2167.94	39	Clothing	4330.03	4467.75	New	0.02	Credit Card	Retail	Sou
4	1061	24-03-2023	Charlie	East	3750.20	13	Electronics	637.37	692.71	New	0.08	Credit Card	Online	East-

[6] 0s

```
#Grain 1: How many rows and columns are there in the dataset?
print(df.shape)
```

```
(1000, 14)
```

```
#Grain 2: What are the column names?
print(df.columns.tolist())
```

```
['Product_ID', 'Sale_Date', 'Sales_Rep', 'Region', 'Sales_Amount', 'Quantity_Sold', 'Product_Category', 'Unit_Cost', 'Unit_Price', 'Customer_Type', 'Discount', 'Payment_Method', 'Sales
```

```
[8] #Grain 3: Show the first 5 rows of the dataset.
    print(df.head())
```

```
   Product_ID   Sale_Date Sales_Rep Region  Sales_Amount  Quantity_Sold  \
0        1052  03-02-2023       Bob  North       5053.97             18
1        1093  21-04-2023       Bob   West       4384.02             17
2        1015  21-09-2023     David  South       4631.23             30
3        1072  24-08-2023       Bob  South       2167.94             39
4        1061  24-03-2023   Charlie   East       3750.20             13

  Product_Category  Unit_Cost  Unit_Price Customer_Type  Discount  \
0        Furniture     152.75      267.22     Returning      0.09
1        Furniture    3816.39     4209.44     Returning      0.11
2             Food     261.56      371.40     Returning      0.20
3         Clothing    4330.03     4467.75           New      0.02
4      Electronics     637.37      692.71           New      0.08

  Payment_Method Sales_Channel Region_and_Sales_Rep
0           Cash        Online           North-Bob
1           Cash        Retail            West-Bob
2  Bank Transfer        Retail          South-David
3    Credit Card        Retail            South-Bob
4    Credit Card        Online         East-Charlie
```

```
[23] #Grain 4: How many unique products were sold?
     print(df['Product_Category'].dropna().nunique())
```

```
4
```

```
[22] #Grain 5: List all unique products.
     print(df['Product_Category'].dropna().unique())
```

```
['Furniture' 'Food' 'Clothing' 'Electronics']
```

```
[25] #Grain 6: What is the total quantity ordered?
     print(df['Quantity_Sold'].sum())
```

```
25355
```

```
[27] #Grain 7: What is the total revenue generated?
     total_revenue = (df['Quantity_Sold'] * df['Unit_Price']).sum()
     print(total_revenue)
```

```
70329940.71
```

[29] 0s

```
#Grain 8: Product with highest quantity sold.
top_product = df.groupby('Product_Category')['Quantity_Sold'].sum().idxmax()
print(top_product)
```

```
Clothing
```

```
[30] #Grain 9: Product with highest sales revenue.
     df['Sales'] = df['Quantity_Sold'] * df['Unit_Price']
     top_sales_product = df.groupby('Product_Category')['Sales_Channel'].sum().idxmax()
     print(top_sales_product)
```

```
Food
```

```
[57] #Grain 21: What are the top 3 cities by total revenue?
     top_3_cities = df.groupby('Region')['Sales'].sum().sort_values(ascending=False).head(3)
     print(top_3_cities)
```

```
Region
North    18208403.85
East     18077401.05
West     17761568.65
Name: Sales, dtype: float64
```

```
#Grain 11:  What is the most expensive product sold (highest price each)?
most_expensive_product = df[['Product_Category', 'Unit_Price']].drop_duplicates().sort_values('Unit_Price', ascending=False).head(1)
print(most_expensive_product)
```

```
    Product_Category  Unit_Price
995             Food     5442.15
```

```
[41] #Grain 12: Total sales for each city.
     print(df.groupby('Region')['Sales'].sum())
```

```
Region
East     18077401.05
North    18208403.85
South    16282567.16
West     17761568.65
Name: Sales, dtype: float64
```

```
[42] #Grain 13: Month with highest sales.
     df['Sale_Date'] = pd.to_datetime(df['Sale_Date'], errors='coerce')  # safely convert
     df['Month'] = df['Sale_Date'].dt.month
     top_month = df.groupby('Month')['Sales'].sum().idxmax()
     print(top_month)
```

```
8.0
```

```
[43] #Grain 14: Total sales month-wise.
     print(df.groupby('Month')['Sales'].sum())
```

```
Month
1.0     2423497.84
2.0     2467048.75
3.0     2486622.26
4.0     1826960.50
5.0     1544951.54
6.0     1811367.95
7.0     1929914.88
8.0     2752057.60
9.0     1503907.67
10.0    2707646.68
11.0    2561514.50
12.0    2363570.63
Name: Sales, dtype: float64
```

```
[44] #Grain 15: Average order value.
     print(df['Sales'].mean())
```

```
70329.94071
```

[46] 0s

```
#Grain 16: Day of week with most sales.
df['Day of Week'] = df['Sale_Date'].dt.day_name()
top_day = df.groupby('Day of Week')['Sales'].sum().idxmax()
print(top_day)
```

```
Wednesday
```

[47] 0s

```
#Grain 17: Time of day with most orders.
df['Hour'] = df['Sale_Date'].dt.hour
top_hour = df['Hour'].value_counts().idxmax()
print(top_hour)
```

```
0.0
```

[48]
```python
#Grain 18: Top 5 products by quantity ordered.
print(df.groupby('Product_Category')['Quantity_Sold'].sum().sort_values(ascending=False).head(5))
```

```
Product_Category
Clothing       6922
Furniture      6729
Electronics    6096
Food           5608
Name: Quantity_Sold, dtype: int64
```

```
[50] #Grain 19: Total number of orders per city.
     print(df['Region'].value_counts())
```

```
Region
North    267
East     263
West     244
South    226
Name: count, dtype: int64
```

+ Code + Text

[53] #Grain 20: Correlation between quantity ordered and price.
0s
```
print(df[['Quantity_Sold', 'Unit_Price']].corr())
```

```
                Quantity_Sold  Unit_Price
Quantity_Sold        1.000000    0.057296
Unit_Price           0.057296    1.000000
```

[50] #Grain 19: Total number of orders per city