STEP-1: DATASET LOADING

```
#IMPORT THE NECESSARY LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#LOAD THE DATASET USING PANDAS
df=pd.read_csv("/content/Sample - Superstore.csv")

#PRINT FIRST 5 ROWS OF THE DATASET
print (df.head(5))
```

```
   Row ID        Order ID  Order Date   Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156  11-08-2016  11-11-2016    Second Class    CG-12520
1       2  CA-2016-152156  11-08-2016  11-11-2016    Second Class    CG-12520
2       3  CA-2016-138688  06-12-2016   6/16/2016    Second Class    DV-13045
3       4  US-2015-108966  10-11-2015  10/18/2015  Standard Class    SO-20335
4       5  US-2015-108966  10-11-2015  10/18/2015  Standard Class    SO-20335

     Customer Name    Segment        Country             City  ...  \
0      Claire Gute   Consumer  United States        Henderson  ...
1      Claire Gute   Consumer  United States        Henderson  ...
2  Darrin Van Huff  Corporate  United States      Los Angeles  ...
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...

  Postal Code  Region       Product ID         Category Sub-Category  \
0       42420   South  FUR-BO-10001798        Furniture    Bookcases
1       42420   South  FUR-CH-10000454        Furniture       Chairs
2       90036    West  OFF-LA-10000240  Office Supplies       Labels
3       33311   South  FUR-TA-10000577        Furniture       Tables
4       33311   South  OFF-ST-10000760  Office Supplies      Storage

                                     Product Name     Sales  Quantity  \
0                 Bush Somerset Collection Bookcase  261.9600         2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2  Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                    Eldon Fold 'N Roll Cart System   22.3680         2

   Discount    Profit
0      0.00   41.9136
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164

[5 rows x 21 columns]
```

```
#PRINT LAST 5 ROWS OF THE DATASET
print(df.tail(5))
```

```
      Row ID        Order ID  Order Date   Ship Date       Ship Mode  \
9989    9990  CA-2014-110422   1/21/2014   1/23/2014    Second Class
9990    9991  CA-2017-121258   2/26/2017  03-03-2017  Standard Class
9991    9992  CA-2017-121258   2/26/2017  03-03-2017  Standard Class
9992    9993  CA-2017-121258   2/26/2017  03-03-2017  Standard Class
9993    9994  CA-2017-119914  05-04-2017  05-09-2017    Second Class

     Customer ID     Customer Name   Segment        Country         City  ...  \
9989    TB-21400  Tom Boeckenhauer  Consumer  United States        Miami  ...
9990    DB-13060       Dave Brooks  Consumer  United States   Costa Mesa  ...
9991    DB-13060       Dave Brooks  Consumer  United States   Costa Mesa  ...
9992    DB-13060       Dave Brooks  Consumer  United States   Costa Mesa  ...
9993    CC-12220      Chris Cortes  Consumer  United States  Westminster  ...

     Postal Code  Region       Product ID         Category Sub-Category  \
9989       33180   South  FUR-FU-10001889        Furniture  Furnishings
9990       92627    West  FUR-FU-10000747        Furniture  Furnishings
9991       92627    West  TEC-PH-10003645       Technology       Phones
9992       92627    West  OFF-PA-10004041  Office Supplies        Paper
9993       92683    West  OFF-AP-10002684  Office Supplies   Appliances

                                      Product Name    Sales  Quantity  \
9989                             Ultra Door Pull Handle   25.248         3
9990  Tenex B1-RE Series Chair Mats for Low Pile Car...   91.960         2
9991                              Aastra 57i VoIP phone  258.576         2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.600         4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou...  243.160         2

      Discount   Profit
9989       0.2   4.1028
9990       0.0  15.6332
9991       0.2  19.3932
9992       0.0  13.3200
9993       0.0  72.9480

[5 rows x 21 columns]
```

```
#Check the following:
#Shape of the dataset
#Column names
#Data types using info()
print(f"1. The shape of the data is: {df.shape}")
print(f"2. The columns of the data are:{df.columns}")
print(f"3. The data structure and data types are described as:")
print(df.info())
```

```
1. The shape of the data is: (9994, 21)
2. The columns of the data are:Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
3. The data structure and data types are described as:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
None
```

STEP-2: DATA CLEANING AND PREPROCESSING

```
#IDENTIFYING MISSING VALUES
print(df.isnull().sum())

#There are no missing values as seen from the output
```

```
Row ID          0
Order ID        0
Order Date      0
Ship Date       0
Ship Mode       0
Customer ID     0
Customer Name   0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Product ID      0
Category        0
Sub-Category    0
Product Name    0
Sales           0
```

```
Quantity        0
Discount        0
Profit          0
dtype: int64
```

```
#CHECK FOR DUPLICATE RECORDS
print(f"The number of duplicate records are: {df.duplicated().sum()}")
```

```
The number of duplicate records are: 0
```

```
#PRINT DATE AND NUMERICAL COLUMNS.
print(f"The date columns are: {df.columns[df.columns.str.contains('Date')]}")
numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
print(f"The numerical columns are:{numerical_columns}")
```

```
The date columns are: Index(['Order Date', 'Ship Date'], dtype='object')
The numerical columns are:Index(['Row ID', 'Postal Code', 'Sales', 'Quantity', 'Discount', 'Profit'], dtype='object')
```

```
#Convert data types where required:
#1.date columns to datetime
#2.Convert numerical columns to int/float (Numerical columns are already in int/float for our data)
df['Order Date'] = pd.to_datetime(df['Order Date'], format='mixed', dayfirst=False)
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='mixed', dayfirst=False)
print(df.head())
```

```
   Row ID        Order ID Order Date  Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
1       2  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
2       3  CA-2016-138688 2016-06-12 2016-06-16    Second Class    DV-13045
3       4  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335
4       5  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335

     Customer Name    Segment        Country             City  ...  \
0      Claire Gute   Consumer  United States        Henderson  ...
1      Claire Gute   Consumer  United States        Henderson  ...
2  Darrin Van Huff  Corporate  United States      Los Angeles  ...
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...

  Postal Code  Region       Product ID         Category Sub-Category  \
0       42420   South  FUR-BO-10001798        Furniture    Bookcases
1       42420   South  FUR-CH-10000454        Furniture       Chairs
2       90036    West  OFF-LA-10000240  Office Supplies       Labels
3       33311   South  FUR-TA-10000577        Furniture       Tables
4       33311   South  OFF-ST-10000760  Office Supplies      Storage

                                     Product Name     Sales  Quantity  \
0                 Bush Somerset Collection Bookcase  261.9600         2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2  Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                    Eldon Fold 'N Roll Cart System   22.3680         2

   Discount    Profit
0      0.00   41.9136
```

```
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164

[5 rows x 21 columns]
```

```
#STANDARDIZE COLUMN NAMES
df.columns = df.columns.str.upper()
print("Updated column names:")
print(df.columns)
```

```
Updated column names:
Index(['ROW ID', 'ORDER ID', 'ORDER DATE', 'SHIP DATE', 'SHIP MODE',
       'CUSTOMER ID', 'CUSTOMER NAME', 'SEGMENT', 'COUNTRY', 'CITY', 'STATE',
       'POSTAL CODE', 'REGION', 'PRODUCT ID', 'CATEGORY', 'SUB-CATEGORY',
       'PRODUCT NAME', 'SALES', 'QUANTITY', 'DISCOUNT', 'PROFIT'],
      dtype='object')
```

STEP-3: EXPLORATORY DATA ANALYSIS

```
##Perform EDA using pandas operations.



#• Identify top or bottom performing categories
 ##
```

```
#Display Summary Statistics
df.describe()
```

Show hidden output

```
#Finding Value counts for categorical columns
print(df['CATEGORY'].value_counts())
print(df['REGION'].value_counts())
print(df['SEGMENT'].value_counts())
print(df['SHIP MODE'].value_counts())
```

Show hidden output

```
#Group-by analysis (e.g., average, total, count)
```

```
category_summary = df.groupby('CATEGORY')[['SALES','PROFIT']].sum().reset_index()
category_summary
```

Show hidden output

Next steps: Generate code with category_summary New interactive sheet

```
region_summary = df.groupby('REGION')[['SALES','PROFIT']].sum().reset_index()
region_summary
```

Show hidden output

Next steps: Generate code with region_summary New interactive sheet

```
#category_analysis = df.groupby('CATEGORY').agg(
    #Sales_Sum=('SALES', 'sum'),
    #Sales_Mean=('SALES', 'mean'),
    #Sales_Count=('SALES', 'count'),
    #Profit_Sum=('PROFIT', 'sum'),
    #Profit_Mean=('PROFIT', 'mean'),
    #Profit_Count=('PROFIT', 'count')
)

#print("Aggregated Sales and Profit by Category:")
#print(category_analysis)
```

Show hidden output

```
# Correlation analysis between numerical columns
df[['SALES','QUANTITY','DISCOUNT','PROFIT']].corr()
```

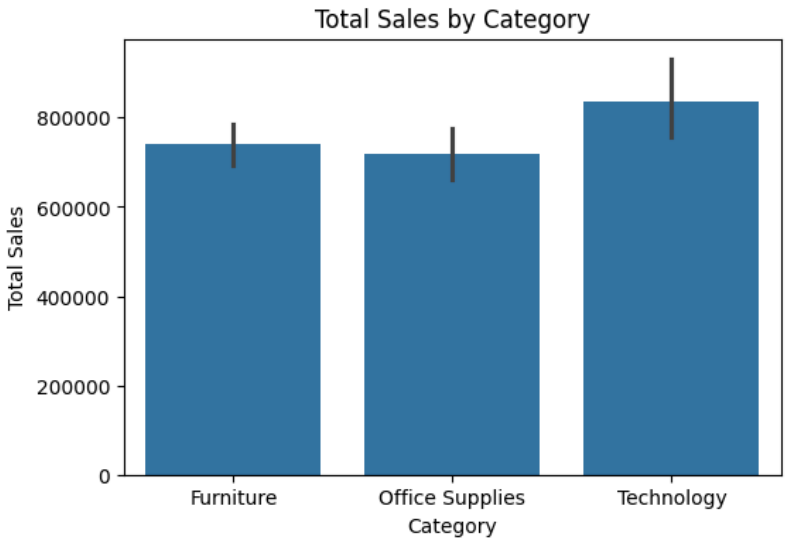
	SALES	QUANTITY	DISCOUNT	PROFIT
SALES	1.000000	0.200795	-0.028190	0.479064
QUANTITY	0.200795	1.000000	0.008623	0.066253
DISCOUNT	-0.028190	0.008623	1.000000	-0.219487
PROFIT	0.479064	0.066253	-0.219487	1.000000

```
# where is top or bottom performing categories??
```

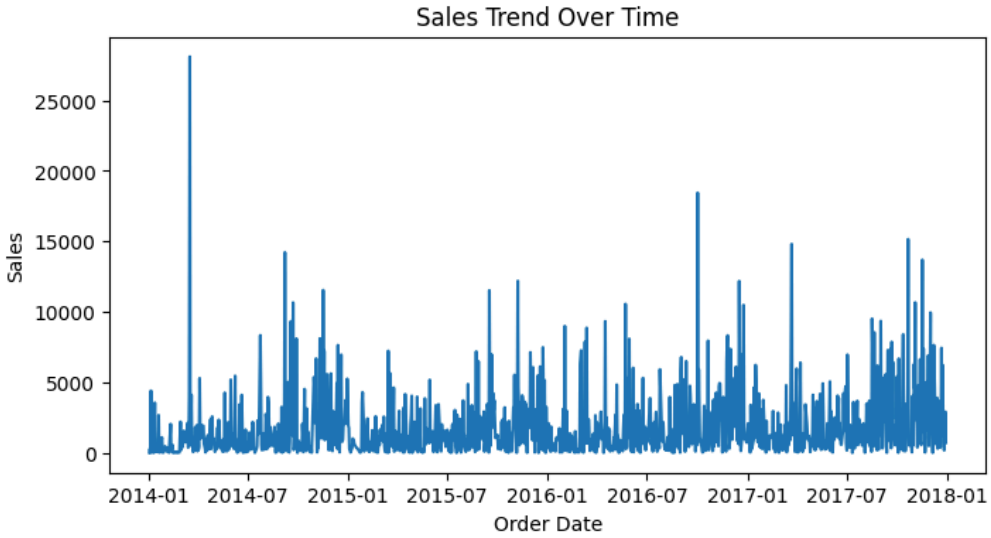
Step 4: Data Visualization

```
#Bar Chart – Category-wise comparison

plt.figure(figsize=(6,4))
sns.barplot(x='CATEGORY', y='SALES', data=df, estimator=sum)
plt.title('Total Sales by Category')
```

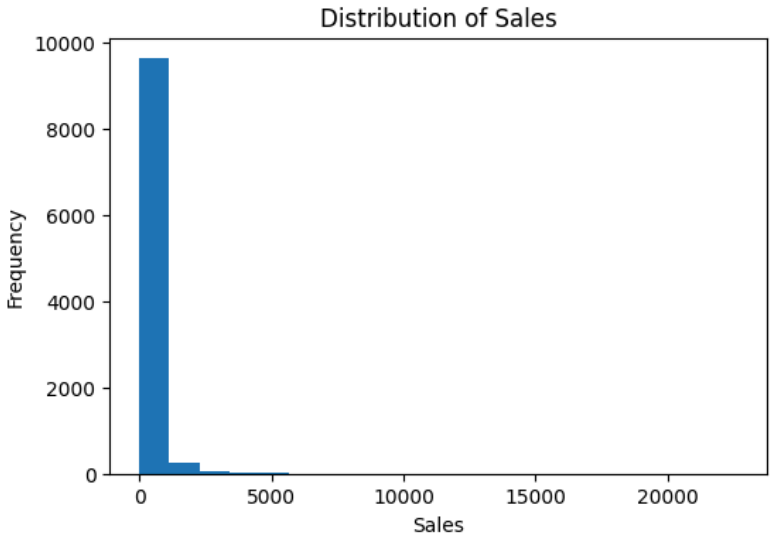
```
plt.xlabel('Category')
plt.ylabel('Total Sales')
plt.show()
```



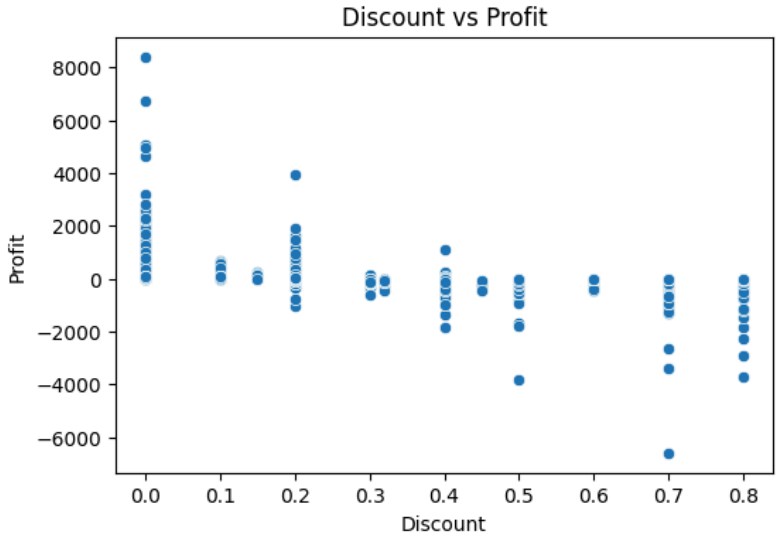

```
#Line Chart – Trend Analysis
df_time = df.groupby('ORDER DATE')['SALES'].sum().reset_index()

plt.figure(figsize=(8,4))
plt.plot(df_time['ORDER DATE'], df_time['SALES'])
plt.title('Sales Trend Over Time')
plt.xlabel('Order Date')
plt.ylabel('Sales')
plt.show()
```




```
# Histogram – Distribution of a numerical column
plt.figure(figsize=(6,4))
plt.hist(df['SALES'], bins=20)
plt.title('Distribution of Sales')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```



```
#Scatter Plot – Relationship between two variables
plt.figure(figsize=(6,4))
sns.scatterplot(x='DISCOUNT', y='PROFIT', data=df)
plt.title('Discount vs Profit')
plt.xlabel('Discount')
plt.ylabel('Profit')
plt.show()
```



Step 5: Insights and Interpretation

1️⃣ Technology category generates the highest sales and profit, indicating strong demand and better margins compared to Furniture and Office Supplies.

2️⃣ The Consumer segment contributes the most to overall sales, showing it is the primary revenue driver for the business.

3️⃣ Higher discounts are often associated with lower or negative profits, suggesting excessive discounting reduces profitability.

4️⃣ Sales show fluctuations over time, indicating seasonal demand patterns rather than steady growth.

5️⃣ The Central region records lower profit compared to sales, highlighting possible cost or pricing inefficiencies in that region.