

NAME:**SAMYAK SANJAY SHINGANKAR**

ROLL NO.:**CS4-01**

PRN:**202401040092**

DATASET:**SALES DATASET**

```
import pandas as pd
import numpy as np
```

Load the dataset

```
df = pd.read_csv("/sales_data.csv")
```

1. Total sales amount

```
total_sales = np.sum(df["Sales_Amount"].values)
print("1. Total Sales Amount:", total_sales)
```

```
➦ 1. Total Sales Amount: 5019265.2299999995
```

2. Average discount given

```
average_discount = np.mean(df["Discount"].values)
print("2. Average Discount:", average_discount)
```

```
➦ 2. Average Discount: 0.15239
```

3. Maximum sales amount

```
max_sales = np.max(df["Sales_Amount"].values)
print("3. Max Sales Amount:", max_sales)
```

```
➦ 3. Max Sales Amount: 9989.04
```

4. Minimum sales amount

```
min_sales = np.min(df["Sales_Amount"].values)
print("4. Min Sales Amount:", min_sales)
```

```
➦ 4. Min Sales Amount: 100.12
```

5. Total revenue (Unit Price * Quantity Sold)

```
revenue = np.multiply(df["Unit_Price"].values, df["Quantity_Sold"].values)
print("5. Total Revenue:", np.sum(revenue))
```

```
➦ 5. Total Revenue: 70329940.71
```

6. Average revenue per sale

```
average_revenue = np.mean(revenue)
print("6. Average Revenue per Sale:", average_revenue)
```

```
➦ 6. Average Revenue per Sale: 70329.94071
```

7. Correlation between Unit Price and Unit Cost

```
correlation = np.corrcoef(df["Unit_Price"].values, df["Unit_Cost"].values)[0, 1]
print("7. Correlation (Price vs Cost):", correlation)
```

```
➦ 7. Correlation (Price vs Cost): 0.9950555602792607
```

8. Count of unique products sold

```
unique_products = df["Product_ID"].nunique()
print("8. Unique Products Sold:", unique_products)
```

8. Unique Products Sold: 100

9. Total quantity sold

```
total_quantity = np.sum(df["Quantity_Sold"].values)
print("9. Total Quantity Sold:", total_quantity)
```

9. Total Quantity Sold: 25355

10. Sales grouped by region

```
sales_by_region = df.groupby("Region")["Sales_Amount"].sum()
print("10. Sales by Region:\n", sales_by_region)
```

10. Sales by Region:

Region	
East	1259792.93
North	1369612.51
South	1154250.86
West	1235608.93

Name: Sales_Amount, dtype: float64

11. Most sold product (by quantity)

```
most_sold_product = df.groupby("Product_ID")["Quantity_Sold"].sum().idxmax()
print("11. Most Sold Product:", most_sold_product)
```

11. Most Sold Product: 1090

12. Average cost per unit

```
average_cost = np.mean(df["Unit_Cost"].values)
print("12. Average Unit Cost:", average_cost)
```

12. Average Unit Cost: 2475.3045500000003

13. Highest discount given

```
max_discount = np.max(df["Discount"].values)
print("13. Max Discount:", max_discount)
```

13. Max Discount: 0.3

14. Number of sales transactions

```
total_transactions = df.shape[0]
print("14. Total Transactions:", total_transactions)
```

14. Total Transactions: 1000

15. Profit per transaction (Sales_Amount - (Unit_Cost * Quantity))

```
cost_total = np.multiply(df["Unit_Cost"].values, df["Quantity_Sold"].values)
profit = df["Sales_Amount"].values - cost_total
print("15. Total Profit:", np.sum(profit))
```

15. Total Profit: -58822828.41

16. Average profit

```
print("16. Average Profit:", np.mean(profit))
```

16. Average Profit: -58822.828409999995

17. Standard deviation of sales amount

```
sales_std = np.std(df["Sales_Amount"].values)
print("17. Sales Amount Standard Deviation:", sales_std)
```

```
↵ 17. Sales Amount Standard Deviation: 2845.3663745785966
```

18. Top 3 regions by profit

```
df['Profit'] = profit
region_profit = df.groupby("Region")['Profit'].sum()
top_regions = region_profit.nlargest(3)
print("18. Top 3 Regions by Profit:\n", top_regions)
```

```
↵ 18. Top 3 Regions by Profit:
   Region
South  -13608579.40
West   -14869867.95
East   -15167050.92
Name: Profit, dtype: float64
```

19. Average quantity per transaction

```
avg_quantity = np.mean(df["Quantity_Sold"].values)
print("19. Average Quantity per Transaction:", avg_quantity)
```

```
↵ 19. Average Quantity per Transaction: 25.355
```

20. Percentage of transactions with discount

```
discount_pct = (df["Discount"].values > 0).mean() * 100
print("20. Percentage of Transactions with Discount:", discount_pct, "%")
```

```
↵ 20. Percentage of Transactions with Discount: 98.4 %
```