NAME :- Sanskar Ramesh Meharkar

ROLL NO :- CS7-88

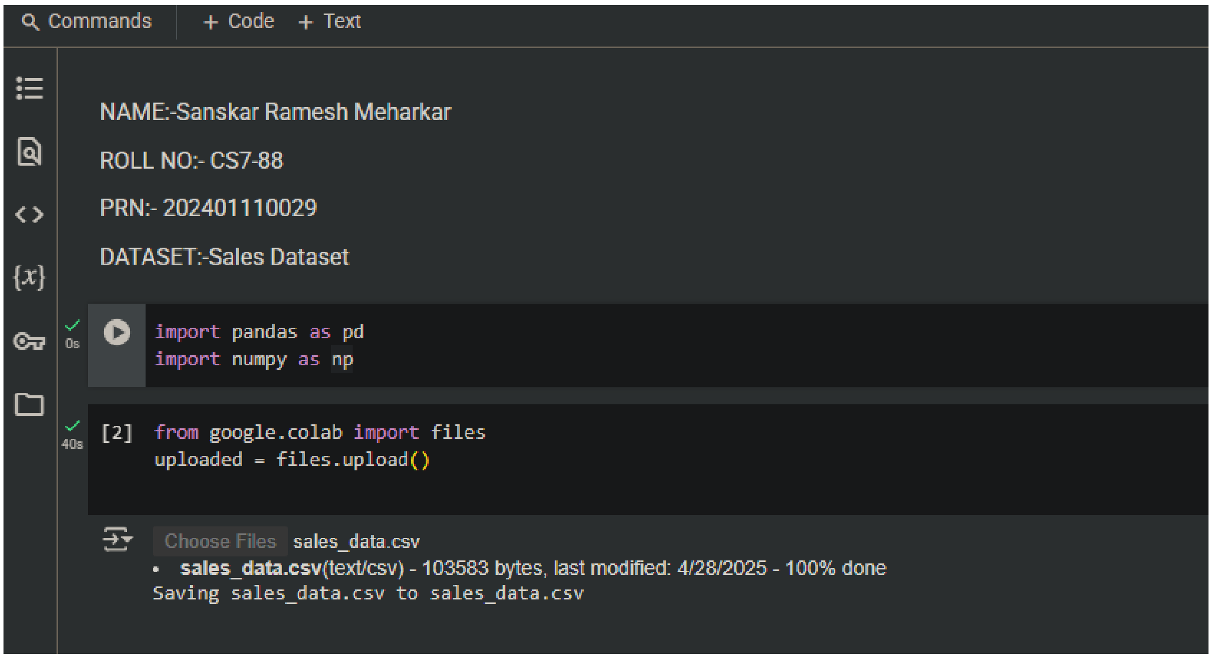
BATCH :- CS74

DIVISION :-CS7

PRN :-202401110029

COLAB LINK:-
https://colab.research.google.com/drive/1RogAuQwRwkd7ZMdEr7HsQ2Bmot3AKrFT?usp=sharing

DATASET LINK :-
https://drive.google.com/drive/folders/1gLamhhVACL8XxdmUeMcd9oGWyMvwSpqh



NAME:-Sanskar Ramesh Meharkar

ROLL NO:- CS7-88

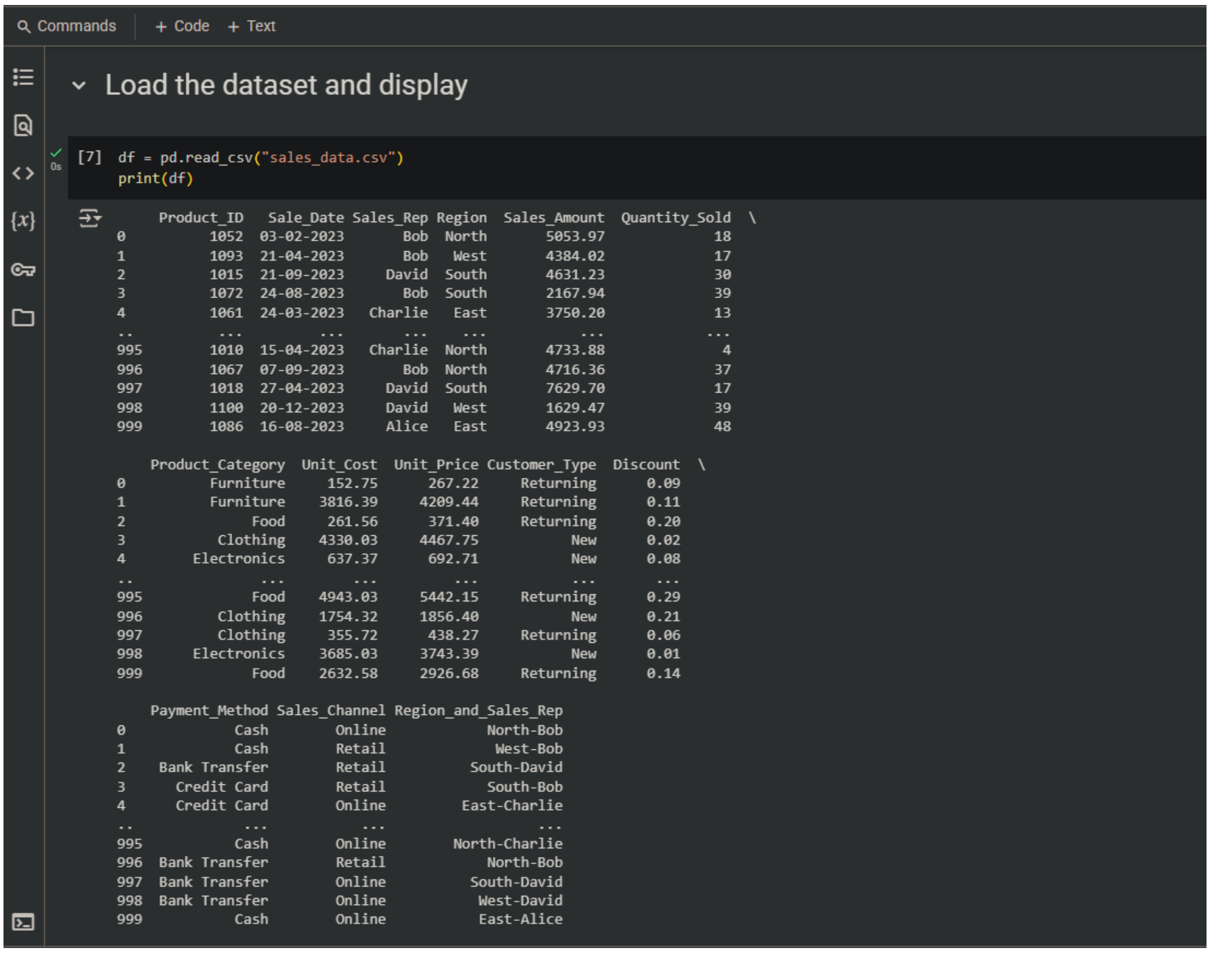
PRN:- 202401110029

DATASET:-Sales Dataset

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

```
[2] from google.colab import files
    uploaded = files.upload()
```

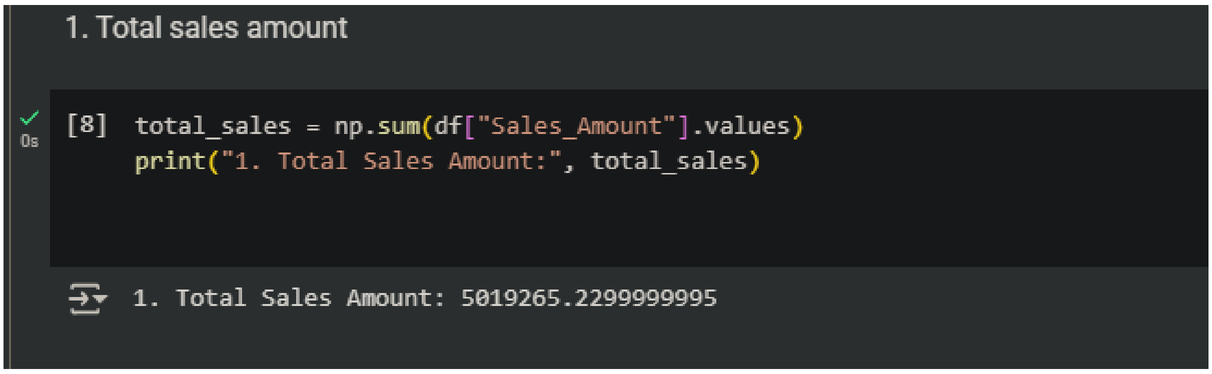
```
sales_data.csv
• sales_data.csv(text/csv) - 103583 bytes, last modified: 4/28/2025 - 100% done
Saving sales_data.csv to sales_data.csv
```



Load the dataset and display

```
[7] df = pd.read_csv("sales_data.csv")
    print(df)
```

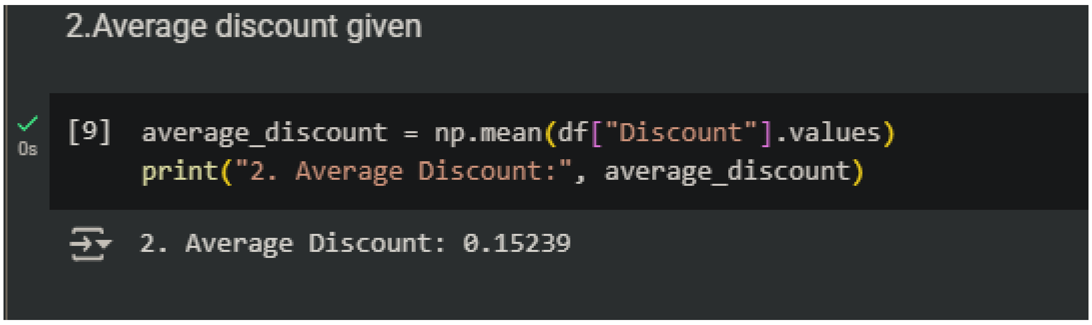
```
     Product_ID  Sale_Date Sales_Rep Region  Sales_Amount  Quantity_Sold  \
0          1052 03-02-2023       Bob  North       5053.97             18
1          1093 21-04-2023       Bob   West       4384.02             17
2          1015 21-09-2023     David  South       4631.23             30
3          1072 24-08-2023       Bob  South       2167.94             39
4          1061 24-03-2023   Charlie   East       3750.20             13
..          ...        ...       ...    ...           ...            ...
995        1010 15-04-2023   Charlie  North       4733.88              4
996        1067 07-09-2023       Bob  North       4716.36             37
997        1018 27-04-2023     David  South       7629.70             17
998        1100 20-12-2023     David   West       1629.47             39
999        1086 16-08-2023     Alice   East       4923.93             48

    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
..               ...        ...         ...           ...       ...
995             Food    4943.03     5442.15     Returning      0.29
996         Clothing    1754.32     1856.40           New      0.21
997         Clothing     355.72      438.27     Returning      0.06
998      Electronics    3685.03     3743.39           New      0.01
999             Food    2632.58     2926.68     Returning      0.14

    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
..             ...           ...                  ...
995           Cash        Online        North-Charlie
996  Bank Transfer        Retail            North-Bob
997  Bank Transfer        Online          South-David
998  Bank Transfer        Online           West-David
999           Cash        Online           East-Alice
```



1. Total sales amount

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

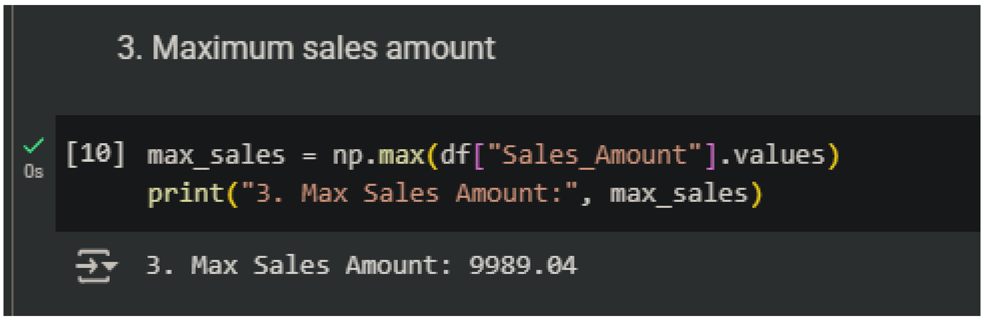
```
1. Total Sales Amount: 5019265.2299999995
```



2.Average discount given

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

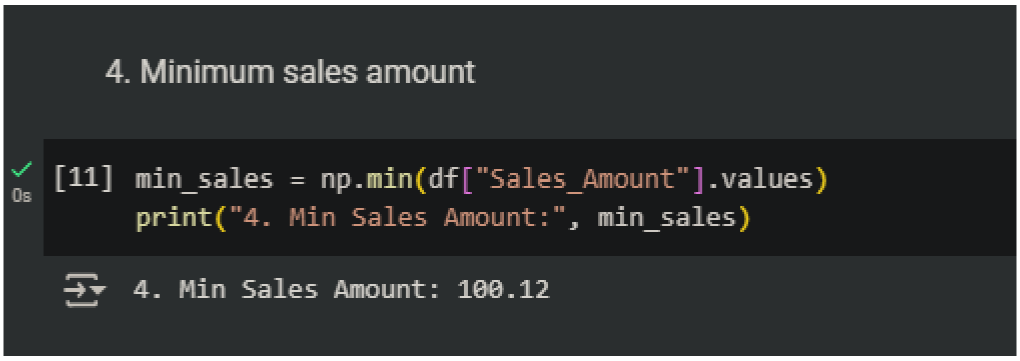
```
2. Average Discount: 0.15239
```



3. Maximum sales amount

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

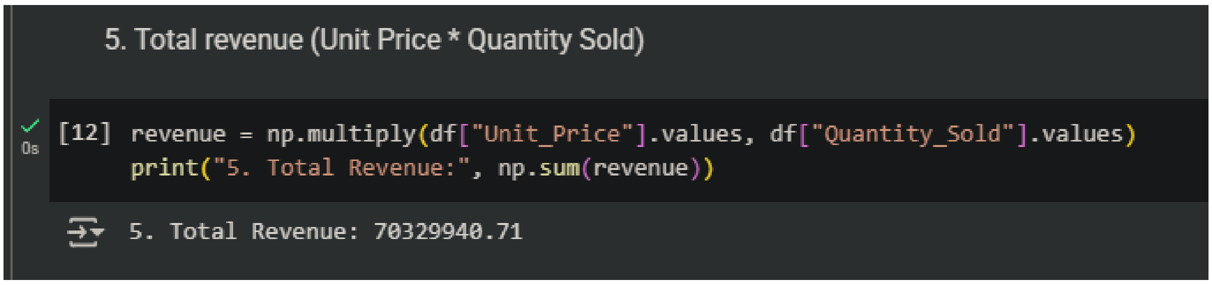
```
3. Max Sales Amount: 9989.04
```



4. Minimum sales amount

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

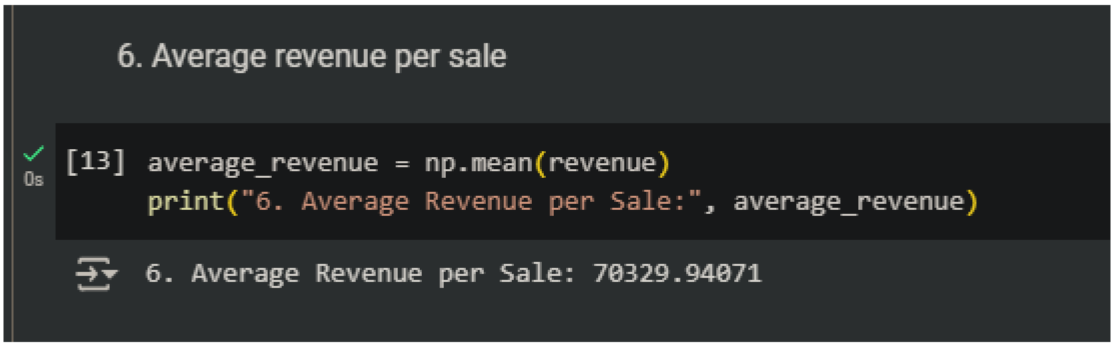
```
4. Min Sales Amount: 100.12
```



5. Total revenue (Unit Price * Quantity Sold)

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

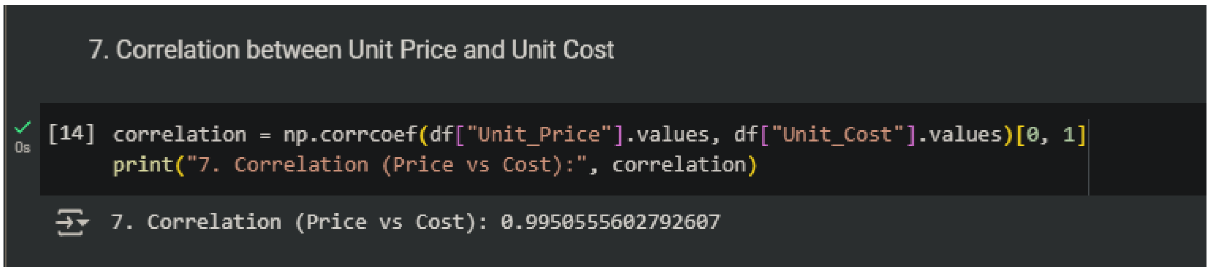
```
5. Total Revenue: 70329940.71
```



6. Average revenue per sale

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

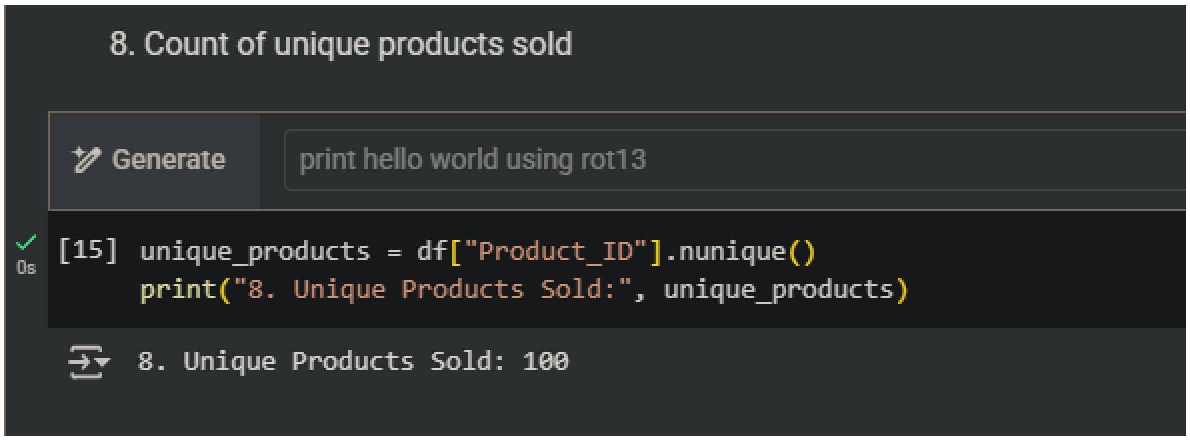
```
6. Average Revenue per Sale: 70329.94071
```



7. Correlation between Unit Price and Unit Cost

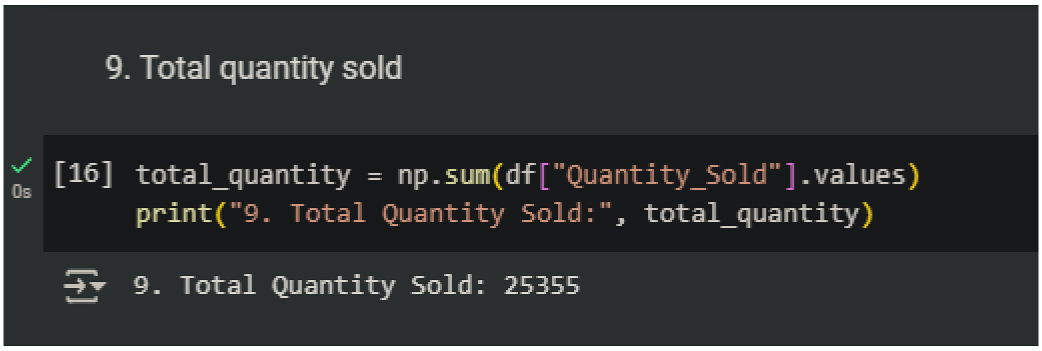
```
[14] correlation = np.corrcoef(df["Unit_Price"].values, df["Unit_Cost"].values)[0, 1]
     print("7. Correlation (Price vs Cost):", correlation)
```

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


8. Count of unique products sold

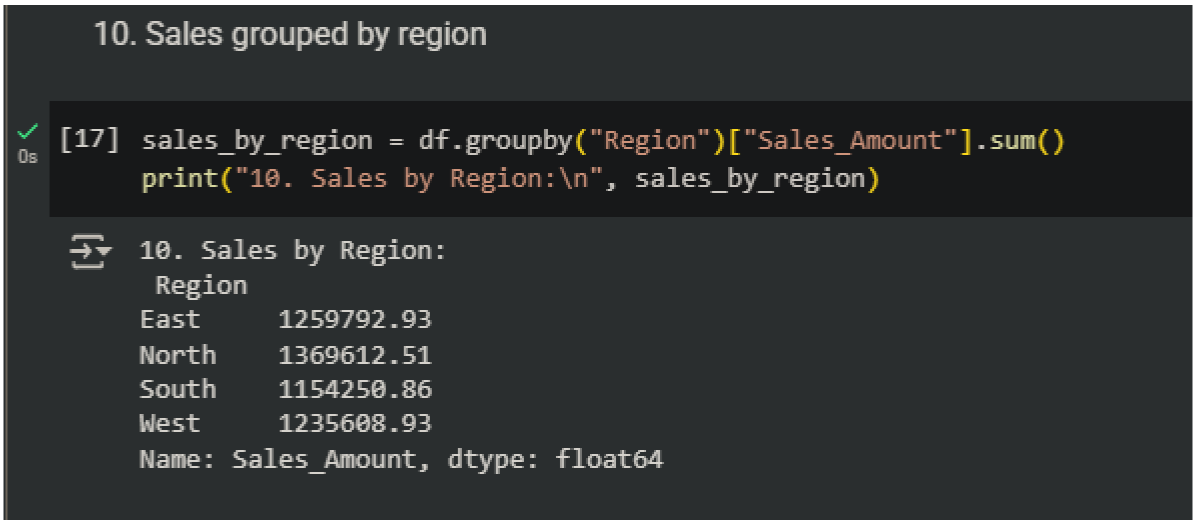
```
[15] unique_products = df["Product_ID"].nunique()
     print("8. Unique Products Sold:", unique_products)
```

```
8. Unique Products Sold: 100
```


9. Total quantity sold

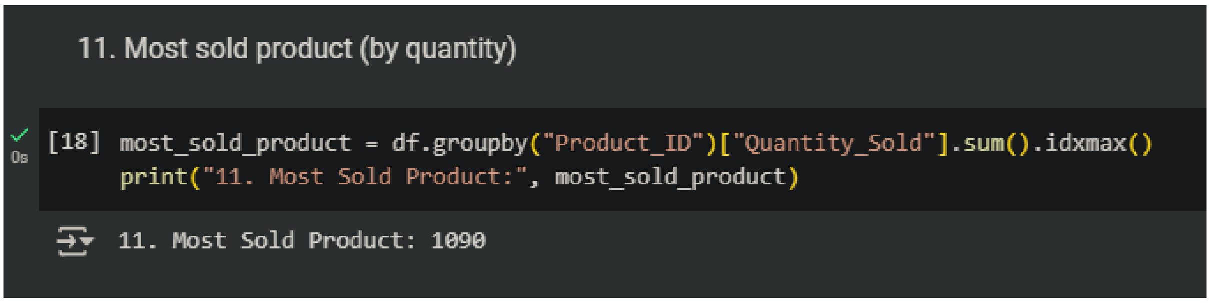
```
[16] total_quantity = np.sum(df["Quantity_Sold"].values)
     print("9. Total Quantity Sold:", total_quantity)
```

```
9. Total Quantity Sold: 25355
```


10. Sales grouped by region

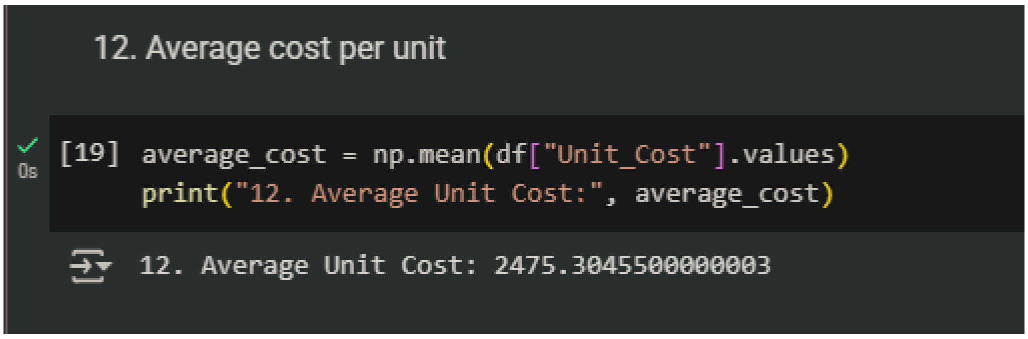
```
[17] sales_by_region = df.groupby("Region")["Sales_Amount"].sum()
     print("10. Sales by Region:\n", sales_by_region)
```

```
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)

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

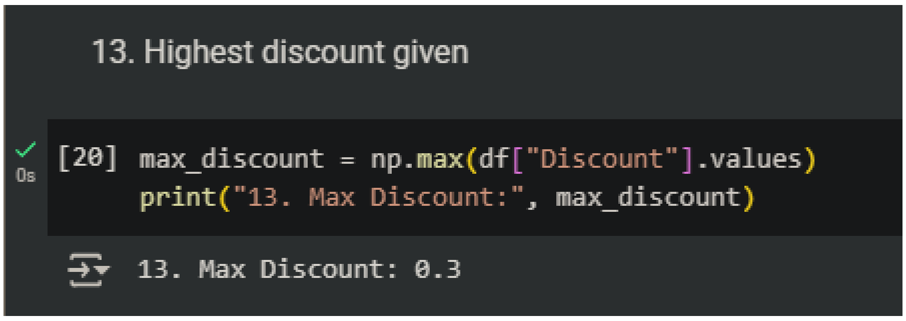
```
11. Most Sold Product: 1090
```



12. Average cost per unit

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

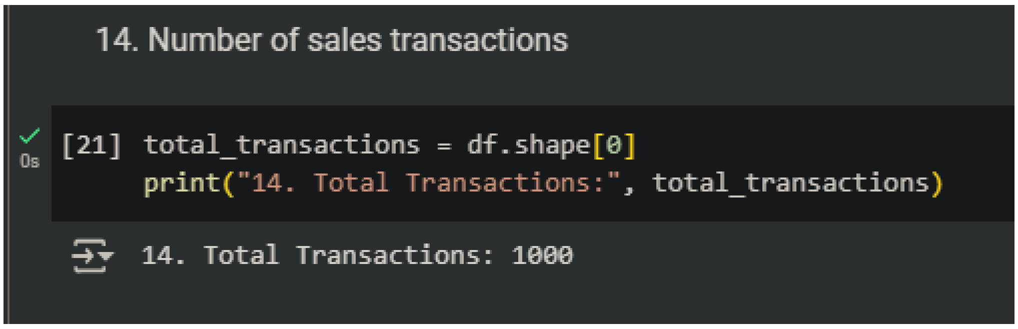
```
12. Average Unit Cost: 2475.3045500000003
```



13. Highest discount given

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

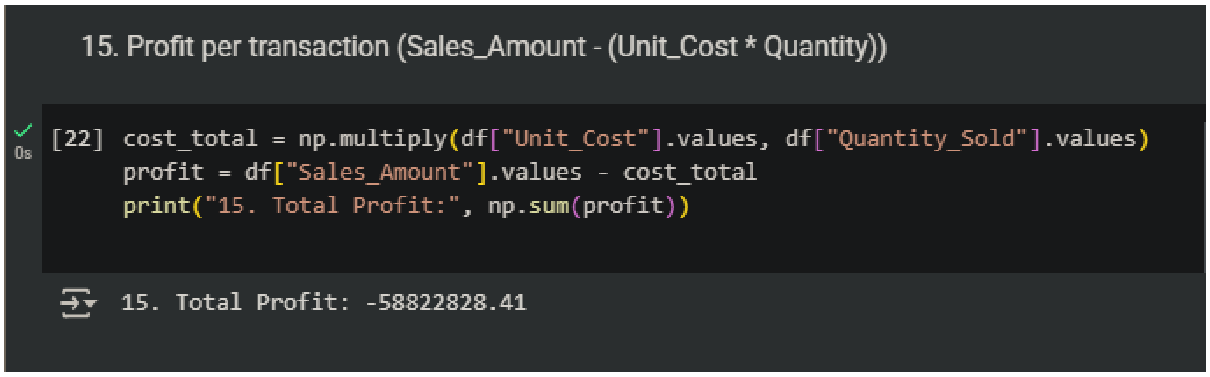
```
13. Max Discount: 0.3
```



14. Number of sales transactions

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

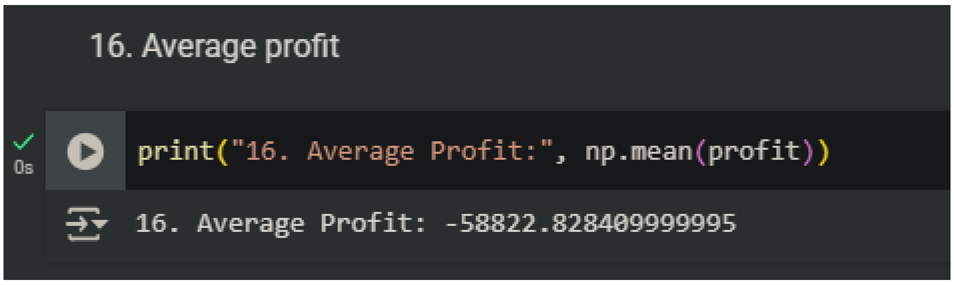
```
14. Total Transactions: 1000
```



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

```
[22] 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))
```

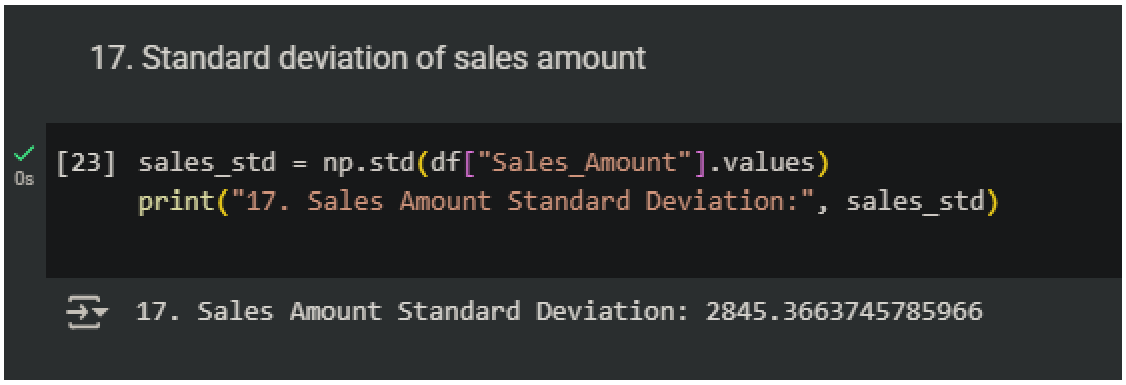
```
15. Total Profit: -58822828.41
```



16. Average profit

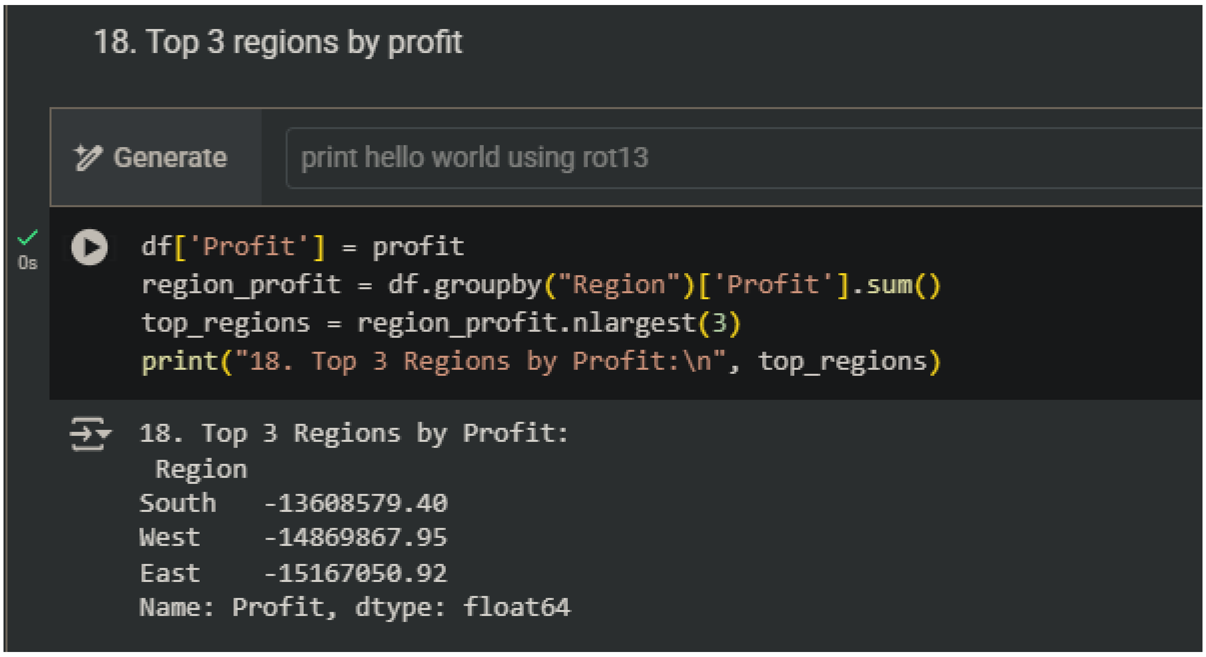
```
print("16. Average Profit:", np.mean(profit))
```

```
16. Average Profit: -58822.828409999995
```


17. Standard deviation of sales amount

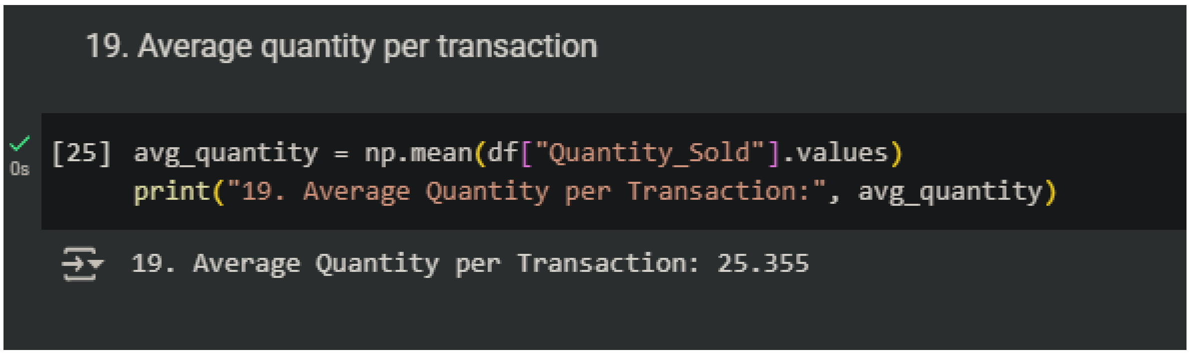
```
[23] sales_std = np.std(df["Sales_Amount"].values)
     print("17. Sales Amount Standard Deviation:", sales_std)
```

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


18. Top 3 regions by profit

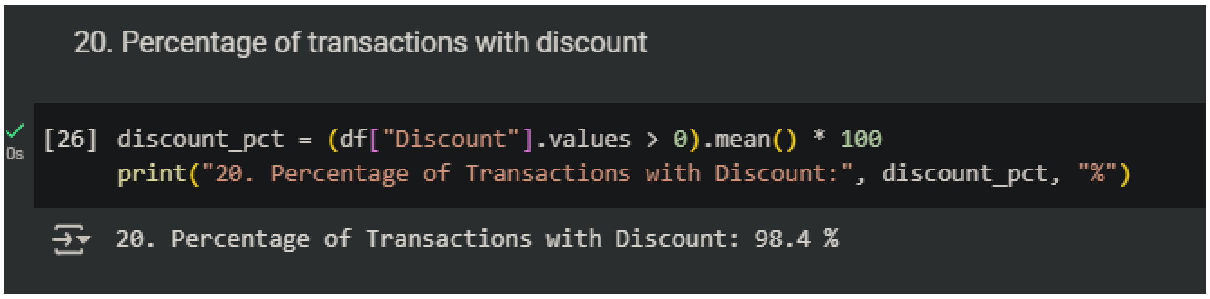
```
df['Profit'] = profit
region_profit = df.groupby("Region")['Profit'].sum()
top_regions = region_profit.nlargest(3)
print("18. Top 3 Regions by Profit:\n", top_regions)
```

```
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

```
[25] 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

```
[26] 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 %
```