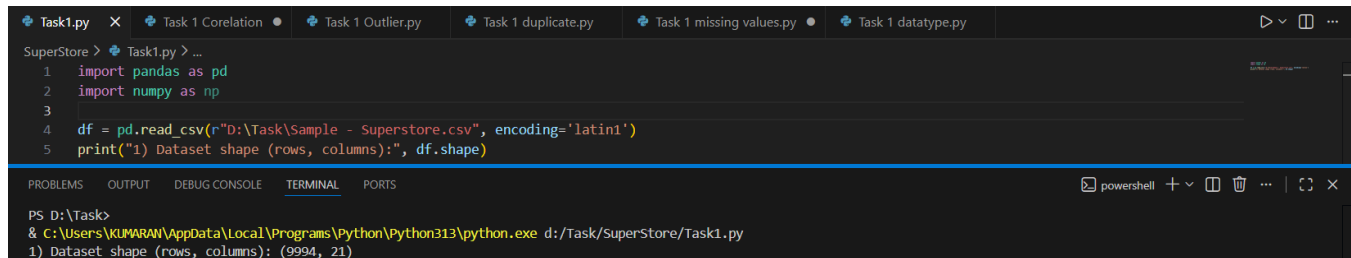


DATA PROFILING REPORT : SAMPLE SUPERSTORE DATASET

Dataset: Sample_Superstore.csv

1. Dataset Structure

- The Superstore dataset contains 9,994 rows and 21 columns.



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar displays several tabs: 'Task1.py', 'Task 1 Correlation', 'Task 1 Outlier.py', 'Task 1 duplicate.py', 'Task 1 missing values.py', and 'Task 1 datatype.py'. The active tab is 'Task1.py', which contains the following Python code:

```
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"D:\Task\Sample - Superstore.csv", encoding='latin1')
5 print("1) Dataset shape (rows, columns):", df.shape)
```

The bottom panel of the notebook shows the 'TERMINAL' output, which displays the command prompt and the execution of the script:

```
PS D:\Task>
& C:\Users\KUMARAW\AppData\Local\Programs\Python\Python313\python.exe d:/Task/SuperStore/Task1.py
1) Dataset shape (rows, columns): (9994, 21)
```

2. Column Names & Data Types

- Row ID (int)
- Order ID (object)
- Order Date (object)
- Ship Date (object)
- Ship Mode (object)
- Customer ID (object)
- Customer Name (object)
- Segment (object)
- Country (object)
- City (object)
- State (object)
- Postal Code (int)
- Region (object)
- Product ID (object)
- Category (object)
- Sub-Category (object)
- Product Name (object)
- Sales (float)
- Quantity (int)
- Discount (float)
- Profit (float)

Task1.py Task 1 Correlation Task 1 Outlier.py Task 1 duplicate.py Task 1 missing values.py Task 1 datatype.py X

SuperStore > Task 1 datatype.py > ...

```
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"D:\Task\Sample - Superstore.csv", encoding='latin1')
5 print("\n2) Columns and Data Types:")
6 print(df.dtypes)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

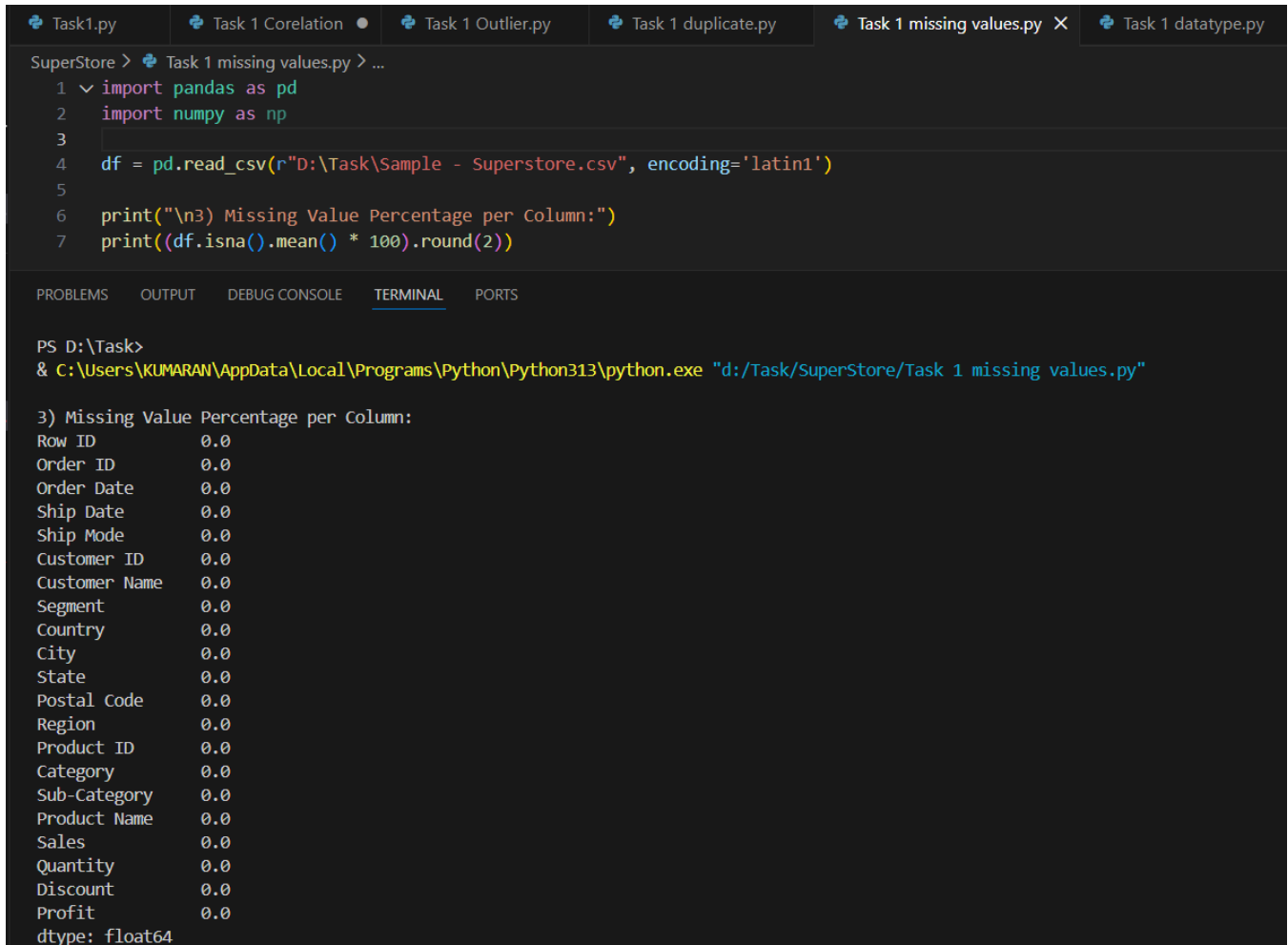
PS D:\Task>
& C:\Users\KUMARAN\AppData\Local\Programs\Python\Python313\python.exe "d:/Task/SuperStore/Task 1 datatype.py"

2) Columns and Data Types:

Row ID	int64
Order ID	object
Order Date	object
Ship Date	object
Ship Mode	object
Customer ID	object
Customer Name	object
Segment	object
Country	object
City	object
State	object
Postal Code	int64
Region	object
Product ID	object
Category	object
Sub-Category	object
Product Name	object
Sales	float64
Quantity	int64
Discount	float64
Profit	float64
dtype:	object

3. Missing Value Analysis

- The dataset contains 0% missing values across all columns.
- No missing values found in key analytical columns such as Sales, Profit, Quantity, Discount, Category, Segment, Region.



The screenshot displays a Jupyter Notebook environment with several tabs at the top: Task1.py, Task 1 Corelation, Task 1 Outlier.py, Task 1 duplicate.py, Task 1 missing values.py (active), and Task 1 datatype.py. The active tab shows a Python script in a code editor. Below the code editor is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (active), and PORTS. The terminal shows the execution of the script, including the command prompt, the command to run the script, and the output of the missing value analysis.

```
SuperStore > Task 1 missing values.py > ...
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"D:\Task\Sample - Superstore.csv", encoding='latin1')
5
6 print("\n3) Missing Value Percentage per Column:")
7 print((df.isna().mean() * 100).round(2))
```

PS D:\Task>
& C:\Users\KUMARAN\AppData\Local\Programs\Python\Python313\python.exe "d:/Task/SuperStore/Task 1 missing values.py"

3) Missing Value Percentage per Column:

Row ID	0.0
Order ID	0.0
Order Date	0.0
Ship Date	0.0
Ship Mode	0.0
Customer ID	0.0
Customer Name	0.0
Segment	0.0
Country	0.0
City	0.0
State	0.0
Postal Code	0.0
Region	0.0
Product ID	0.0
Category	0.0
Sub-Category	0.0
Product Name	0.0
Sales	0.0
Quantity	0.0
Discount	0.0
Profit	0.0

dtype: float64

4. Duplicate Records

- Duplicate rows detected: 0
- Final dataset contains only unique order-line entries.

```
Task1.py Task 1 Corelation Task 1 Outlier.py Task 1 duplicate.py X Task 1 missing values.py Task 1 datatype.py
SuperStore > Task 1 duplicate.py > ...
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"D:\Task\Sample - Superstore.csv", encoding='latin1')
5
6 print("\n4) Duplicate Rows Count:", df.duplicated().sum())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Task>
& C:\Users\KUMARAN\AppData\Local\Programs\Python\Python313\python.exe "d:/Task/SuperStore/Task 1 missing values.py"

3) Missing Value Percentage per Column:
Row ID      0.0
Order ID    0.0
Order Date  0.0
Ship Date   0.0
Ship Mode   0.0
Customer ID 0.0
Customer Name 0.0
Segment     0.0
Country     0.0
City        0.0
State       0.0
Postal Code 0.0
Region      0.0
Product ID  0.0
Category    0.0
Sub-Category 0.0
Product Name 0.0
Sales       0.0
Quantity    0.0
Discount    0.0
Profit      0.0
dtype: float64
```

5. Outlier Analysis

- Sales Outliers: 1,167
- Profit Outliers: 1,881

```
Task1.py Task 1 Correlation Task 1 Outlier.py X Task 1 duplicate.py Task 1 missing values.py Task 1 datatype.py
SuperStore > Task 1 Outlier.py > ...
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"D:\Task\Sample - Superstore.csv", encoding='latin1')
5 import pandas as pd
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Task>
& C:\Users\KUMARAN\AppData\Local\Programs\Python\Python313\python.exe "d:/Task/SuperStore/Task 1 Outlier.py"
Sales Outliers Count: 1167
Sales Lower Bound: -271.71000000000004
Sales Upper Bound: 498.93

Profit Outliers Count: 1881
Profit Lower Bound: -39.724125
Profit Upper Bound: 70.816875
```

6. Relationship Analysis (Correlation)

Discount vs Profit (−0.21 correlation)

- Small negative correlation confirms that increasing discount generally reduces profitability.

Sales vs Quantity (+0.20 correlation)

- Positive but weak relationship: increasing quantity does not always mean higher sales value.

```
Task1.py Task 1 Correlation X Task 1 Outlier.py Task 1 duplicate.py Task 1 missing values.py Task 1 datatype.py
SuperStore > Task 1 Correlation > ...
6 corr_discount_profit = df['Discount'].corr(df['Profit'])
7 corr_sales_quantity = df['Sales'].corr(df['Quantity'])
8
9 print("\n Correlation Checks:")
10 print(f" Discount vs Profit: {corr_discount_profit:.4f}")
11 print(f" Sales vs Quantity: {corr_sales_quantity:.4f}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Task>
& C:\Users\KUMARAN\AppData\Local\Programs\Python\Python313\python.exe "d:/Task/SuperStore/Task 1 Correlation"

Correlation Checks:
Discount vs Profit: -0.2195
Sales vs Quantity: 0.2008
```

7. Summary of Insights

- Dataset is clean with no missing values or duplicates.
- Sales & Profit contain heavy outliers.
- Discounting reduces profitability.
- Quantity has weak influence on Sales.
- Profit volatility is high.