RETAIL STORE SALES ANALYSIS

Submitted By:

Nandana.Nv

Submitted To:

Shilpa Miss

CONTENTS

1. INTRODUCTION

This project focuses on cleaning and analyzing the Retail Store Sales dataset to understand customer purchasing behavior and sales performance. The dataset includes information such as transaction details , product categories, payment methods, price per unit,quantity,total spent, time-related features like month and year of purchase and discounts. The primary objective of this study is to identify factors that influence sales, evaluate the performance of various product categories, and understand the impact of factors such as pricing, discounts, and payment methods on total revenue. The analysis and visualization were performed using python along libraries such as pandas for data cleaning, Matplotlib and Seaborn for visualization ,and Numpy for numerical operations.

2. DATASET DESCRIPTION

Transaction ID	Customer	Category	Item	Price Per Unit	Quantity	Total Spent	Payment Method	Location	Transaction Date	Discount Applied
TXN_6867343	CUST_09	Patisserie	Item_10_PAT	18.5	10	185	Digital Wallet	Online	4/8/2024	TRUE
TXN_3731986	CUST_22	Milk Products	Item_17_MILK	29	9	261	Digital Wallet	Online	7/23/2023	TRUE
TXN_9303719	CUST_02	Butchers	Item_12_BUT	21.5	2	43	Credit Card	Online	10/5/2022	FALSE
TXN_9458126	CUST_06	Beverages	Item_16_BEV	27.5	9	247.5	Credit Card	Online	5/7/2022	
TXN_4575373	CUST_05	Food	Item_6_FOOD	12.5	7	87.5	Digital Wallet	Online	10/2/2022	FALSE
TXN_7482416	CUST_09	Patisserie			10	200	Credit Card	Online	11/30/2023	
TXN_3652209	CUST_07	Food	Item_1_FOOD	5	8	40	Credit Card	In-store	6/10/2023	TRUE
TXN_1372952	CUST_21	Furniture		33.5			Digital Wallet	In-store	4/2/2024	TRUE
TXN_9728486	CUST_23	Furniture	Item_16_FUR	27.5	1	27.5	Credit Card	In-store	4/26/2023	FALSE
TXN_2722661	CUST_25	Butchers	Item_22_BUT	36.5	3	109.5	Cash	Online	3/14/2024	FALSE
TXN_8776416	CUST_22	Butchers	Item_3_BUT	8	9	72	Cash	In-store	12/14/2024	TRUE
TXN_5422631	CUST_09	Milk Products			8	52	Digital Wallet	In-store	1/12/2025	TRUE
TXN_5874772	CUST_23	Food	Item_2_FOOD	6.5	7	45.5	Cash	Online	9/9/2023	TRUE
TXN_4413070	CUST_14	Patisserie	Item_24_PAT	39.5	6	237	Digital Wallet	In-store	5/20/2022	FALSE
TXN_2490363	CUST_09	Milk Products	Item_16_MILK	27.5	2	55	Digital Wallet	Online	5/22/2022	
TXN_1809665	CUST_14	Beverages		24.5			Credit Card	In-store	5/11/2022	
TXN_7563311	CUST_23	Patisserie	Item_17_PAT	29	8	232	Cash	Online	11/16/2024	TRUE
TXN_9634894	CUST_15	Milk Products			10	275	Digital Wallet	Online	4/17/2022	
TXN_4396807	CUST_17	Electric household essentials	Item_13_EHE	23	1	23	Digital Wallet	In-store	2/7/2022	FALSE
TXN_4206593	CUST_01	Furniture		35			Digital Wallet	Online	1/13/2025	FALSE
TXN_9939063	CUST_14	Beverages	Item_7_BEV	14	9	126	Digital Wallet	In-store	1/14/2024	
TXN_8685338	CUST_15	Milk Products			3	105	Credit Card	In-store	10/29/2023	
TXN_6547964	CUST_10	Electric household essentials	Item_4_EHE	9.5	7	66.5	Cash	In-store	8/15/2022	
TXN_3314099	CUST_04	Food	Item_10_FOOD	18.5	1	18.5	Cash	Online	9/30/2023	FALSE

Column Name	**Description**	**Data Type**
Transaction ID	Unique ID for each sales transaction	String
Customer ID	Unique Id for each customer	String
Category	Product Category	String
Item	Name of the product	String
Price Per Unit	Price of the single unit of the item	Float
Quantity	Number of items purchased	Integer
Total Spent	Total amount spent(Quantity*Price Per Unit)	Float
Payment Method	Mode of Payment	String
Purchase Mode	Where the purchase was made	String
Transaction Date	Date of the transaction	Date
Discount Applied	Discount applied on the transaction	String

Dataset Source : Kaggle Total

Rows: 12575

Total Columns: 11

3. DATA CLEANING AND PREPROCESSING

3.1 Importing Data

The dataset was imported using pandas in python.

```
import pandas as pd
data=pd.read_csv("retail_store_sales.csv")
data
```

3.2 Checking for Missing Values

We used data**.** isnull ( )**.**any( ) to identify missing values**.**

```
data.isnull().any()
```

	0
Transaction ID	False
Customer ID	False
Category	False
Item	True
Price Per Unit	True
Quantity	True
Total Spent	True
Payment Method	False
Location	False
Transaction Date	False
Discount Applied	True

3.3 Drop Unnecessary Columns

The columns Customer ID and Item were dropped as they were not needed for analysis.

```
data = data.drop(columns=['Customer ID', 'Item'], errors='ignore')
data
```

3.4 Creating New Columns

Year and Month columns were extracted from the Transaction Date column

```
data['Transaction Date'] = pd.to_datetime(data['Transaction Date'])
data['Month'] = data['Transaction Date'].dt.strftime('%B')
data['Year'] = data['Transaction Date'].dt.year
data
```

3.5 Rename Column Names

The columns Total Spent, Price per Unit, and Location were renamed to maintain consistency and improve readability in the dataset.

```
data.rename(columns={
    "Price Per Unit":"Price_Per_unit",
    "Total Spent":"Total_Spent",
     "Location":"Purchase_Mode"
},inplace=True)
data
```

3.6 Handling Missing Values

- **Price Per Unit:** Filled with the mean value

```
data["Price_Per_unit"].fillna(data["Price_Per_unit"].mean(),inplace=True)
data
```

- **Quantity:** Filled missing values with 0

```
data["Quantity"].fillna(0,inplace=True)
data
```

- **Total Spent:** Filled missing values with 0

```
data["Total_Spent"].fillna(0,inplace=True)
data
```

- **Discount Applied:** Filled missing with unknown

```
data["Discount Applied"].fillna("unknown",inplace=True)
data
```

```
data.isnull().any()
```

	0
Transaction ID	False
Category	False
Price_Per_unit	False
Quantity	False
Total_Spent	False
Payment Method	False
Purchase_Mode	False
Transaction Date	False
Discount Applied	False
Month	False
Year	False

3.7 Check Duplicates

```
data.duplicated()
```

	0
0	False
1	False
2	False
3	False
4	False
...	...
12570	False
12571	False
12572	False
12573	False
12574	False

4. EXPLORATORY DATA ANALYSIS(EDA)

- Statistical summary
  - ❖ data.describe ( )

```
data.describe()
```

	Price Per Unit	Quantity	Total Spent
count	11966.000000	11971.000000	11971.000000
mean	23.365912	5.536380	129.652577
std	10.743519	2.857883	94.750697
min	5.000000	1.000000	5.000000
25%	14.000000	3.000000	51.000000
50%	23.000000	6.000000	108.500000
75%	33.500000	8.000000	192.000000
max	41.000000	10.000000	410.000000

  - ❖ data.info( )

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12575 entries, 0 to 12574
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction ID    12575 non-null  object
 1   Customer ID       12575 non-null  object
 2   Category          12575 non-null  object
 3   Item              11362 non-null  object
 4   Price Per Unit    11966 non-null  float64
 5   Quantity          11971 non-null  float64
 6   Total Spent       11971 non-null  float64
 7   Payment Method    12575 non-null  object
 8   Location          12575 non-null  object
 9   Transaction Date  12575 non-null  object
 10  Discount Applied  8376 non-null   object
dtypes: float64(3), object(8)
memory usage: 1.1+ MB
```

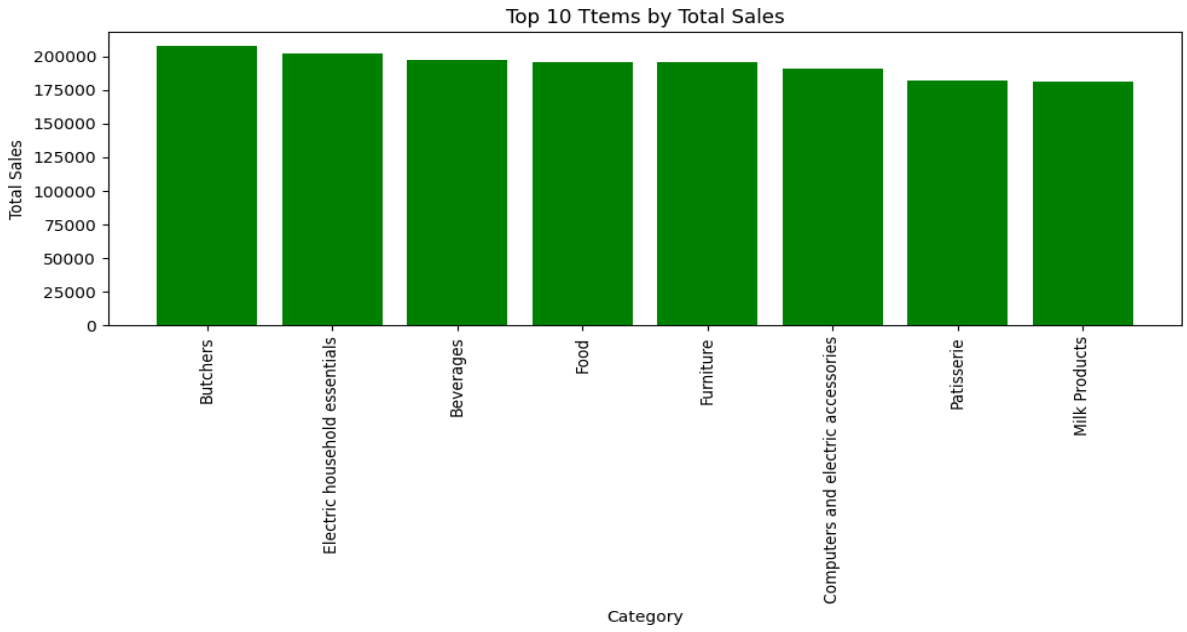
VISUALIZATION

Various visualizations were created using matplotlib and seaborn

4.1 Bar Chart -Top 10 Items by Total Sales

Displays the top 10 items by total sales in the retail store dataset. This visualization helps to identify which product categories contribute the most to overall revenue

```
import matplotlib.pyplot as plt
data["Total_Spent"]=data["Price_Per_unit"]*data["Quantity"]
Total_Spent=data.groupby("Category")["Total_Spent"].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(10,6))
plt.bar(Total_Spent.index,Total_Spent.values,color="green")
plt.title("Top 10 Ttems by Total Sales")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



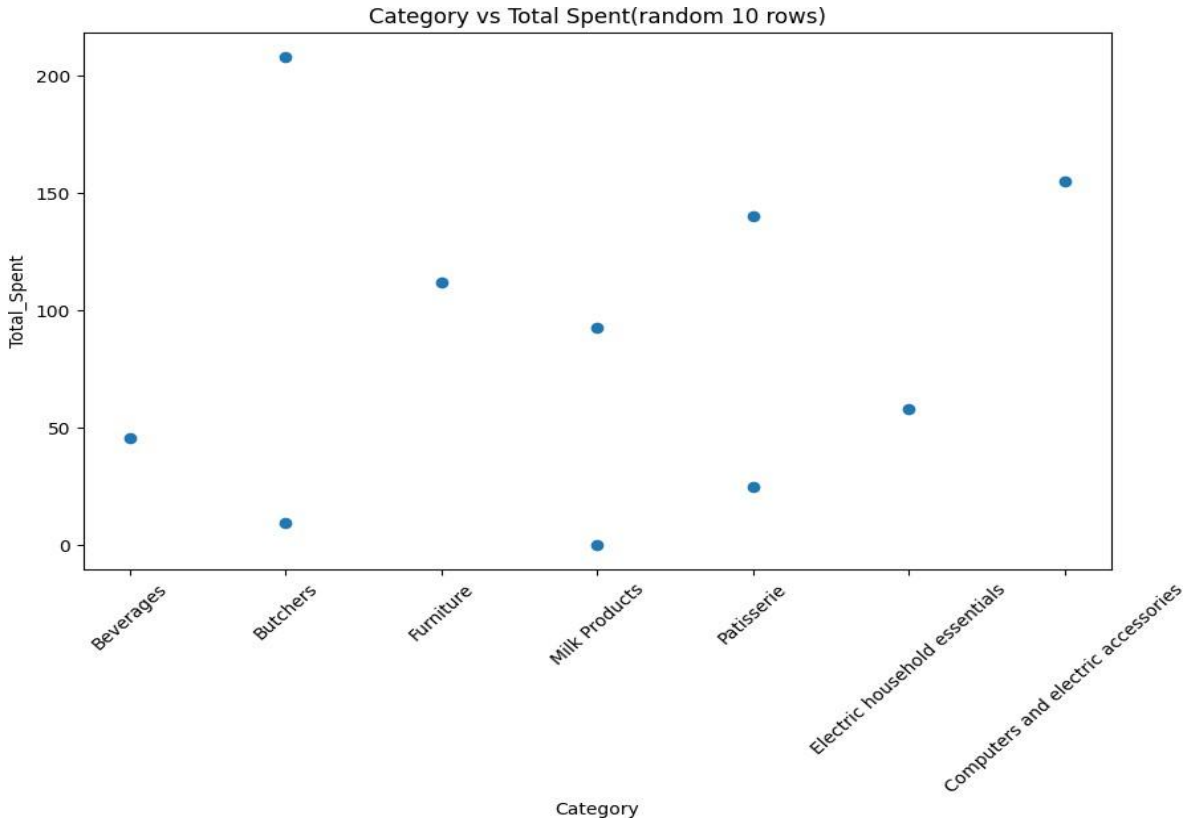
Interpretation

The chart shows the top 10 product categories by total sales. Butchers had the highest sales, followed by Electric household essentials and Beverages. Overall, sales among the top categories are fairly consistent, with only slight differences between them.

4.2 Scatter Plot

Shows the relationship between Category and Total Spent for a random sample of 10 rows from the dataset

```
import matplotlib.pyplot as plt
#randomly Select 10 rows
sample_data=data.sample(10)
plt.figure(figsize=(10,6))
plt.scatter(sample_data["Category"],sample_data["Total_Spent"])
plt.xlabel("Category")
plt.ylabel("Total_Spent")
plt.title("Category vs Total Spent(random 10 rows)")
plt.xticks(rotation=45)
plt.show()
```



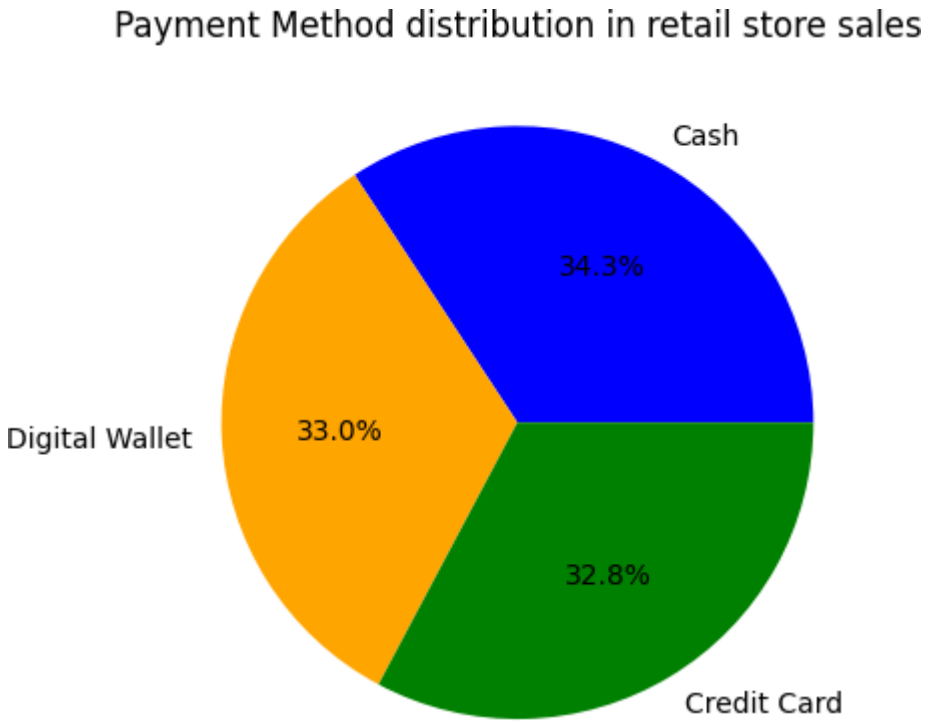
Interpretation

This chart shows the relationship between product categories and total spending for a random sample of 10 transactions. It helps to visualize how spending varies across categories, showing that some categories have much higher or lower spending than others in the sample.

4.3 Pie chart

Shows the percentage distribution of payment methods used in retail store sales.

```
data["Payment Method"].value_counts().plot(
    kind="pie",
    autopct="%1.1f%%",
    colors=["Blue","Orange","green"])
plt.title("Payment Method distribution in retail store sales")
plt.ylabel("")
plt.show()
```



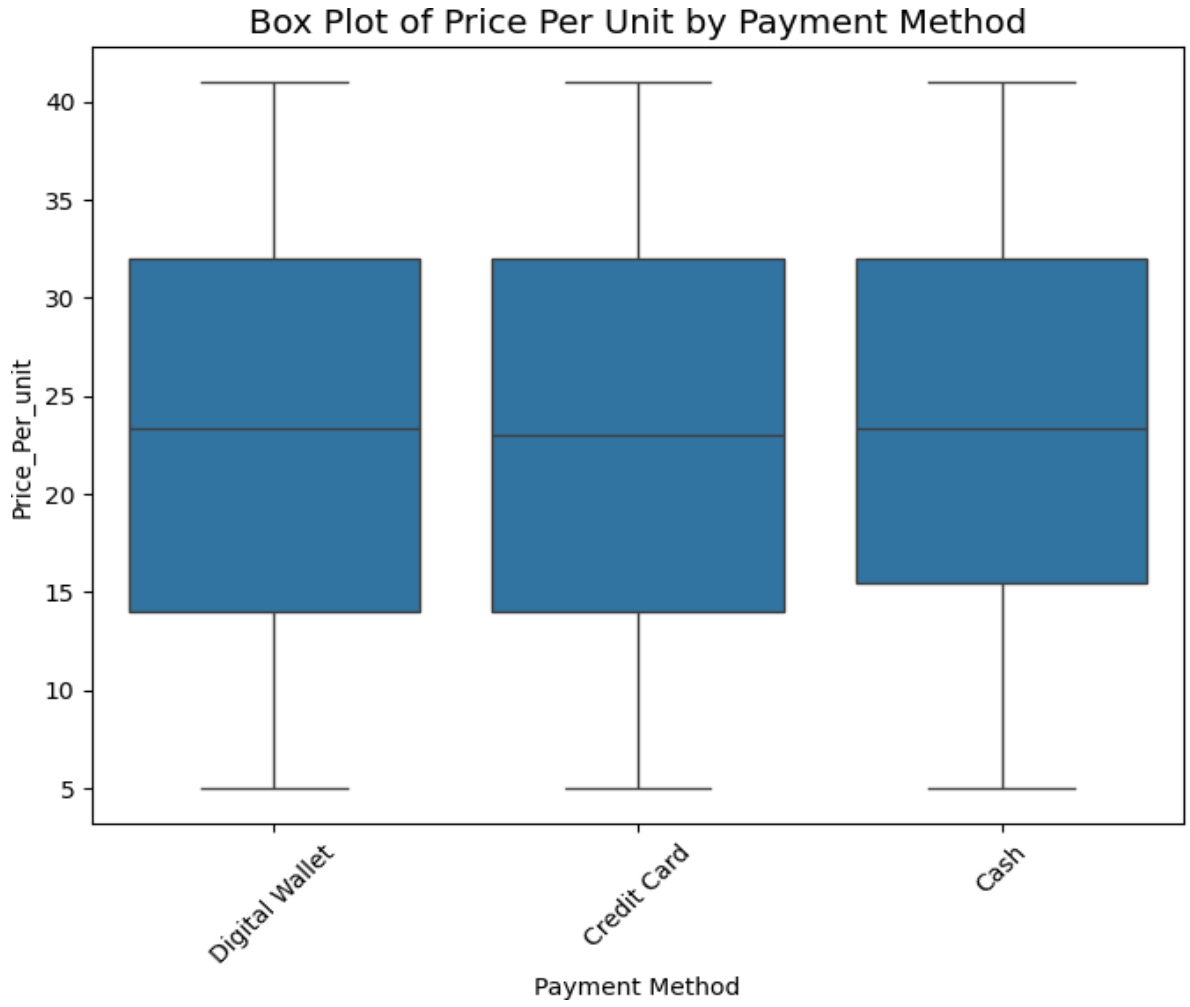
Interpretation

From the above chart 34.3% of customers were prefer to pay with cash, 33% of customers use Digital wallet –almost equal to cash payments and 32.8% customers were used credit cards for payment.

4.4 Box Plot

Box plot showing the distribution of Price per Unit across different Payment Methods

```
plt.figure(figsize=(8, 6))
sns.boxplot(x='Payment Method',y='Price_Per_unit', data=data)
plt.title('Box Plot of Price Per Unit by Payment Method', fontsize=14)
plt.xticks(rotation=45)
plt.show()
```



Interpretation

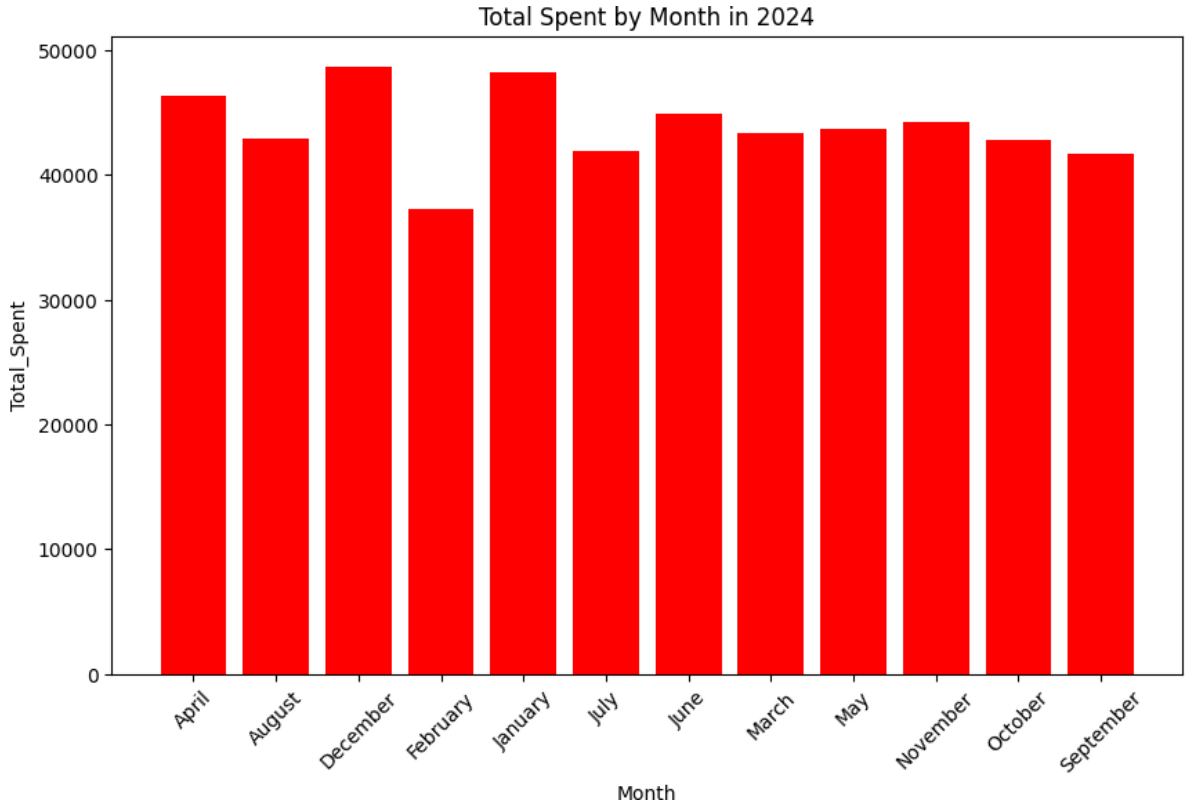
The box plot shows that the median Price per Unit is nearly the same across all payment methods, indicating consistent pricing regardless of how customers pay. The spread and range of prices are also similar for all methods.

4.5 Bar chart

Bar chart showing the total amount spent each month in 2024.

```
data_2024=data[data["Year"]==2024]
data['Month'] = pd.Categorical(data['Month'],
    categories=['January', 'February', 'March', 'April', 'May', 'June',
                'July', 'August', 'September', 'October', 'November', 'December'],
    ordered=True
)
Month_totals=data_2024.groupby("Month")["Total_Spent"].sum().sort_index()
plt.figure(figsize=(10,6))
plt.bar(Month_totals.index,Month_totals.values,color="red")
plt.xlabel("Month")
plt.ylabel("Total_Spent")
plt.title("Total Spent by Month in 2024")
plt.xticks(rotation=45)
plt.show()
```

I



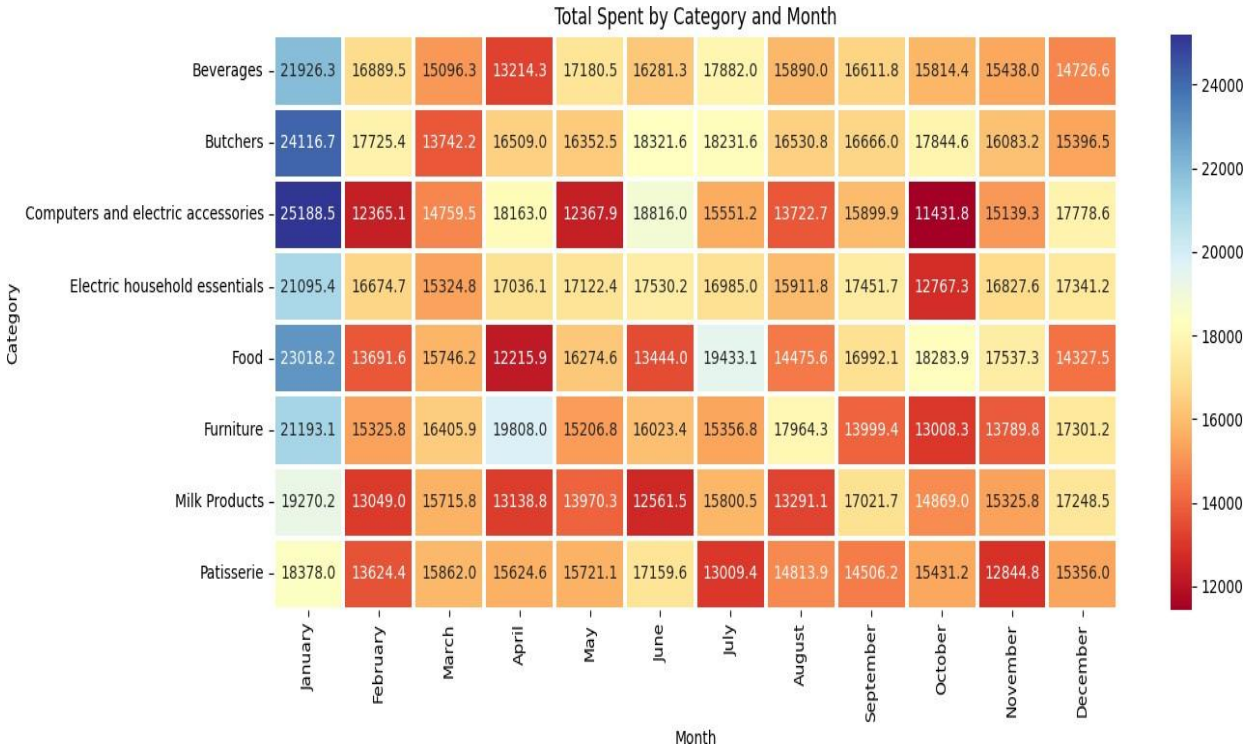
Interpretation

The chart shows that total spending in 2024 remained fairly steady across months, with the highest spending in December and April, and the lowest in February.

4.6 Heat Map

Shows the total amount spent for each category across different months, helping visualize spending patterns and trends over time.

```
data['Month'] = pd.Categorical(data['Month'],
    categories=['January', 'February', 'March', 'April', 'May', 'June',
                'July', 'August', 'September', 'October', 'November', 'December'],
    ordered=True
)
# Create a pivot table: Total Spent by Category and Month
heatmap_data = data.pivot_table(
    index='Category',
    columns='Month',
    values='Total_Spent',
    aggfunc='sum'
)
# Plot the heatmap
plt.figure(figsize=(14, 6))
sns.heatmap(heatmap_data, annot=True, fmt="0.1f",cmap="RdYlBu",linewidths=1.5, linecolor='white')
plt.title("Total Spent by Category and Month")
plt.xlabel("Month")
plt.ylabel("Category")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



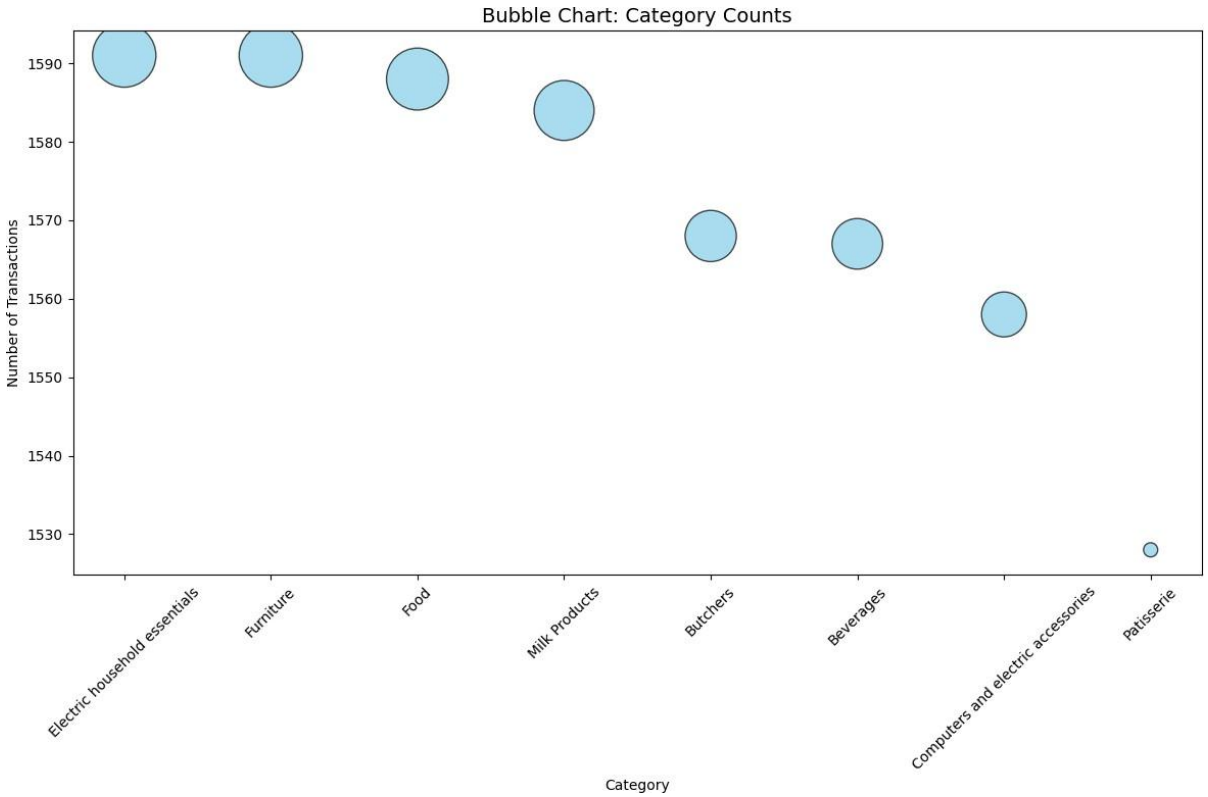
Interpretation

Above chart shows monthly spending across categories. Higher spending is seen in January, especially for Computers and electric accessories and Butchers, while spending generally declines toward December, indicating stronger purchases at the start of the year.

4.7 Bubble Chart

Shows the number of transactions in each category, where the bubble size represents how frequently each category appears in the dataset.

```
category_counts = data['Category'].value_counts().reset_index()
category_counts.columns = ['Category', 'Count']
min_size = 100
max_size = 2000
sizes = (
    (category_counts['Count'] - category_counts['Count'].min()) /
    (category_counts['Count'].max() - category_counts['Count'].min())
) * (max_size - min_size) + min_size
plt.figure(figsize=(12, 8))
scatter = plt.scatter(
    x=category_counts['Category'],
    y=category_counts['Count'],
    s=sizes,
    color='skyblue',
    alpha=0.7,
    edgecolors='k'
)
plt.title('Bubble Chart: Category Counts', fontsize=14)
plt.xlabel('Category')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



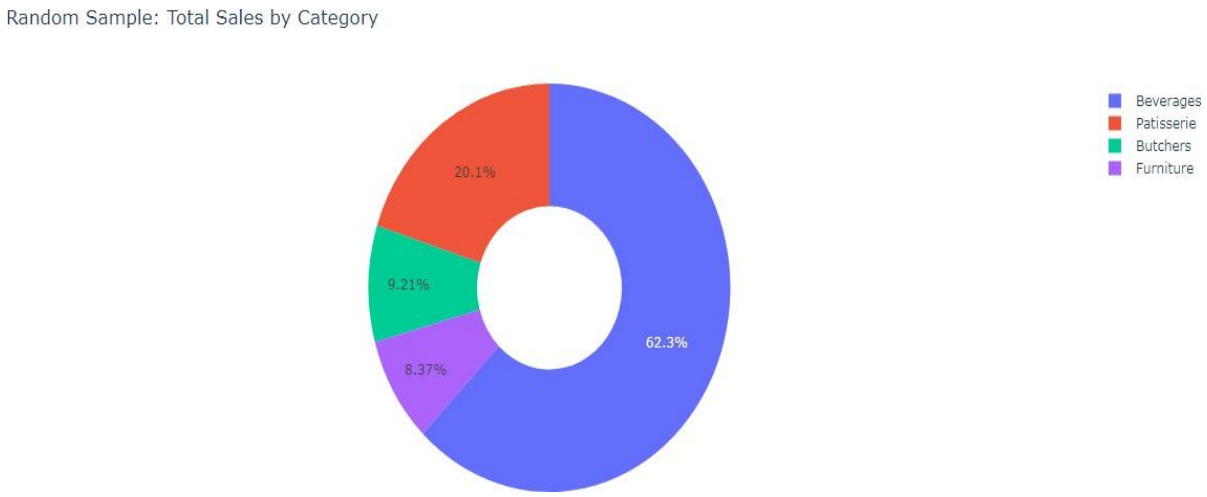
Interpretation

The bubble chart shows that Electric household essentials, Furniture, and Food have the highest number of transactions, while Patisserie has the least, indicating varying popularity among product categories.

4.8 Donut Chart

Donut (pie) chart using a random sample of five records to show the total sales by category, highlighting how spending is distributed among selected categories.

```
import plotly.express as px
data_random = data.sample(n=5, random_state=42)
data_agg = data_random.groupby('Category')['Total_Spent'].sum().reset_index()
fig = px.pie(
    data_agg,
    names='Category',
    values='Total_Spent',
    hole=0.4,
    title='Random Sample: Total Sales by Category'
)
fig.show()
```



Interpretation

Above chart chart shows that Beverages account for the largest share of total sales in the random sample, while Furniture and Butchers contribute smaller portions, indicating uneven sales distribution among categories.

4.9 Line Chart

Shows the monthly trend of total spending in 2023, helping to visualize how spending levels changed throughout the year.

```
data['Transaction Date'] = pd.to_datetime(data['Transaction Date'])
data_2023 = data[data['Transaction Date'].dt.year == 2023]
data_2023['Month'] = data_2023['Transaction Date'].dt.month_name()
monthly_spent = data_2023.groupby('Month')['Total_Spent'].sum().reindex([
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
])
plt.figure(figsize=(10, 6))
plt.plot(monthly_spent.index, monthly_spent.values, color='green', marker='o', linewidth=2)
plt.title('Monthly Variation of Total Spent in 2023', fontsize=14)
plt.xlabel('Month')
plt.ylabel('Total_Spent')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

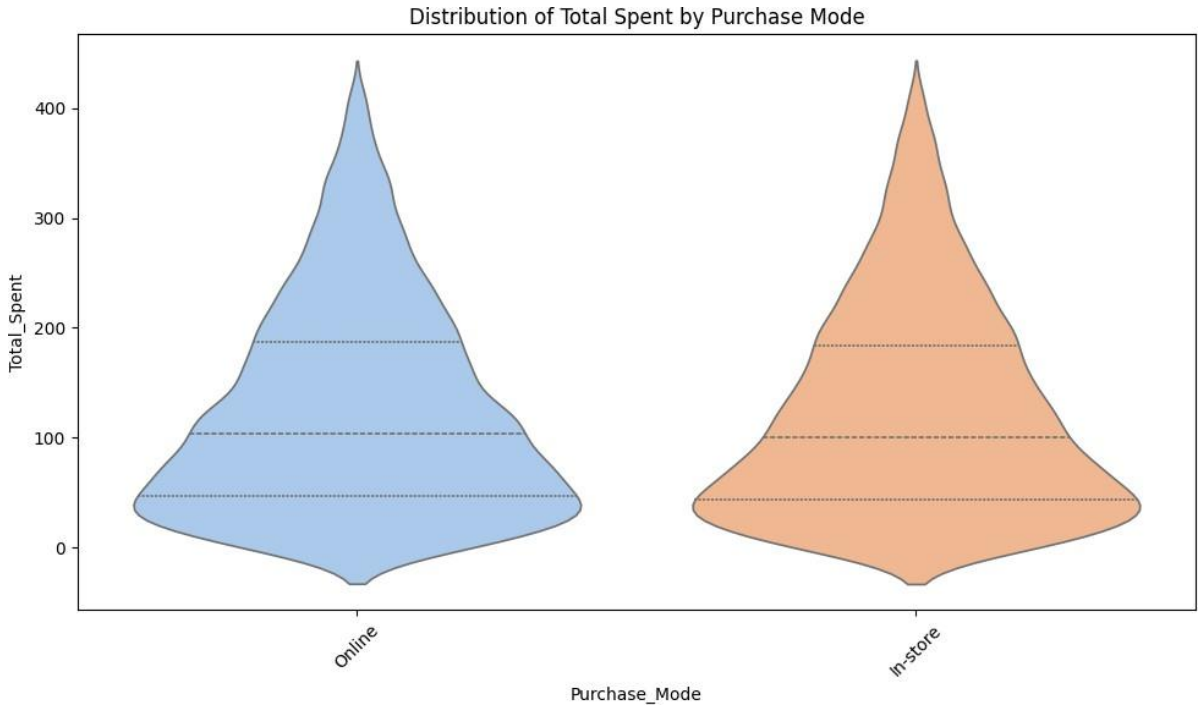
Interpretation

The line chart shows that total spending was highest in January and July, while it dipped around March and November, indicating fluctuating spending patterns across 2023 with peaks at the beginning and mid-year.

4.10 Violin Chart

Shows the distribution of total spending across different purchase modes, highlighting how spending varies and where most transactions are concentrated for each mode.

```
plt.figure(figsize=(10,6))
sns.violinplot(data=data, x='Purchase_Mode', y='Total_Spent', inner='quartile', palette='pastel')
plt.title('Distribution of Total Spent by Purchase Mode')
plt.xlabel('Purchase_Mode')
plt.ylabel('Total_Spent')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



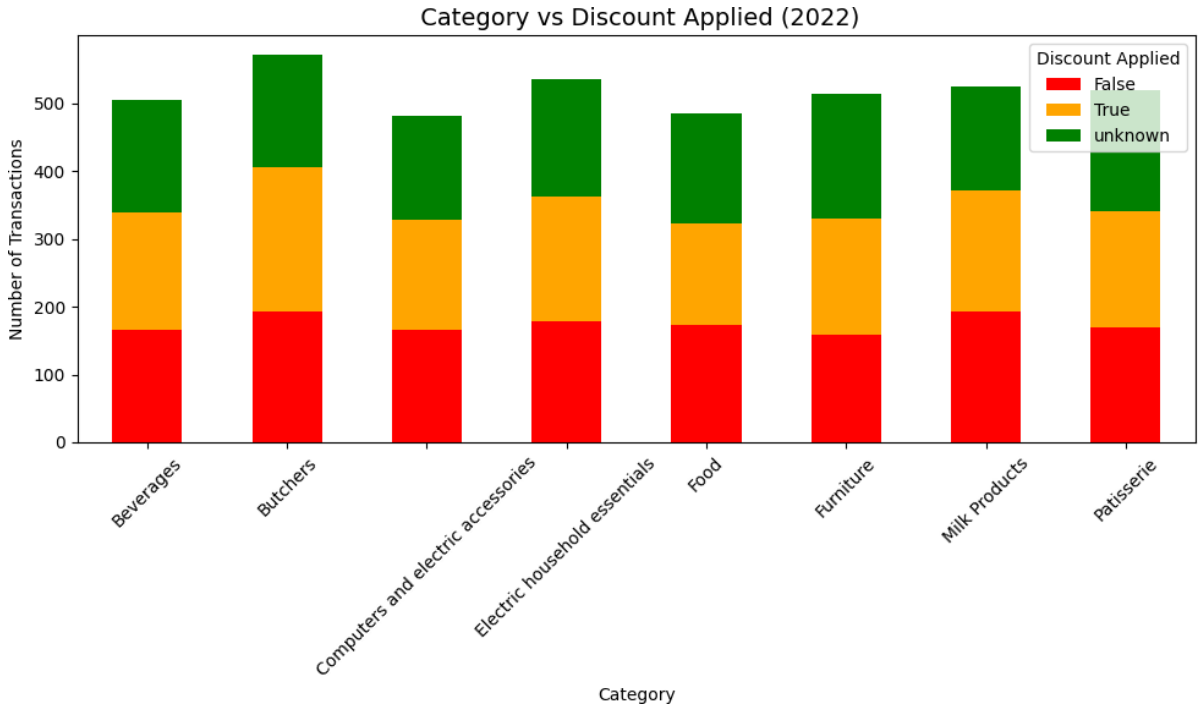
Interpretation

Above chart shows that both online and in-store purchases have similar spending distributions, with most transactions around the lower range but a few high-value purchases in each mode.

4.11 Staked Bar chart

Shows how discounts were applied across different categories in 2022, allowing comparison of the number of discounted and non-discounted transactions per category.

```
data['Transaction Date'] = pd.to_datetime(data['Transaction Date'])
data_2022 = data[data['Transaction Date'].dt.year == 2022]
category_discount_2022 = pd.crosstab(
    data_2022['Category'],
    data_2022['Discount Applied']
)
category_discount_2022.plot(
    kind='bar',
    stacked=True,
    figsize=(10, 6),
    color=['red','orange','green']
)
plt.title(' Category vs Discount Applied (2022)', fontsize=14)
plt.xlabel('Category')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=45)
plt.legend(title='Discount Applied')
plt.tight_layout()
plt.show()
```



Interpretation

The stacked bar chart shows that all categories had a mix of discounted and non-discounted transactions in 2022, with Butchers and Milk Products having slightly more transactions overall compared to other categories.

CONCLUSION

The main aim of this analysis was to explore the Retail Store Sales dataset to understand sales trends, customer behavior, and factors influencing overall business performance.

Through detailed exploratory data analysis, several key insights were discovered**:**

The analysis revealed that Butchers, Electric Household Essentials, and Beverages were the top-performing categories. Payment methods were almost evenly distributed among cash, digital wallets, and credit cards, showing customer flexibility. Spending patterns were relatively steady throughout the year, with peaks in January, April, and December. Discount analysis showed that most categories received a balanced mix of discounted and non-discounted sales, with Butchers and Milk Products showing slightly higher transaction volumes. Overall, sales remained consistent, with minor variations by category and month.

These insights can guide retail managers to optimize inventory for high-selling categories, adjust discount strategies to maintain steady sales, and plan marketing campaigns during low-spending months. Understanding customer payment preferences can also help improve checkout experiences and encourage more digital transactions. Overall, this analysis supports data-driven decisions to boost profitability and customer satisfaction.