UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

GROUP PROJECT BSD2333 DATA WRANGLING

SEMESTER II 2022/2023

TITLE:

TESCO RETAIL TRANSACTION ANALYSIS

PREPARED FOR:

DR. MOHD KHAIRUL BAZLI BIN MOHD AZIZ

GROUP NAME: COLAB

MATRIC ID	NAME	SECTION
SD22004	NUR AINUL NASUHA BINTI MOHD AZREEN	01G
SD22013	AQILAH MAISARAH BINTI AZIZI	01G
SD22012	PUTRI BALQIS BATRISYIA BINTI MOHD RIZAL	01G
SD22023	NUR SYAZWANA BINTI ROSPI	01G
SD22055	AMIRAH AISYAH BINTI ABDULLAH	01G

Table of content

1.0 Synopsis

1.1 Description of the assignment

In today's digital age, Tesco, like many retail businesses, is faced with a massive quantity of transactional data. These datasets are essential for modern retail businesses like Tesco because they offer insightful information that has the potential to completely change how companies interact with their customers. Every transaction is a record of customer behavior, preferences, and trends, rather than just a simple exchange of products for money. Tesco can discover undiscovered opportunities in its transactional data by utilizing machine learning and advanced analytics. From identifying patterns and predicting future trends to personalizing marketing strategies and optimizing inventory management, the possibilities are endless. In this data-driven landscape, Tesco must embrace the power of its transactional data to gain a competitive edge and thrive in a dynamic market.

1.2 Problem to be solved

Retail transaction data often contains inconsistencies and unstructured information, making it a little challenging to get meaningful insights. The problem to be addressed in this project is the management and preparation of Tesco retail transaction data to uncover trends, patterns, and opportunities for the business.

1.3 Questions to be answered

1. What are the top-selling products at Tesco based on the quantity sold and the total amount generated?
2. How effective are discount strategies in influencing customer purchasing behavior and driving product sales?
3. What correlations exist between customer purchasing behavior metrics (frequency, quantity, payment method) and product sales performance?

1.4 Objective

1. To determine the top-selling products at Tesco based on quantity sold and the total amount generated.
2. To analyze the effectiveness of Tesco's discount strategies in influencing customer purchasing behavior and driving product sales performance.
3. To explore patterns in customer purchasing behavior, such as frequency of purchases, quantity bought, and preferred payment methods, and analyze their correlation with product sales performance to better understand consumer preferences and optimize marketing strategies.

1.5 Basic Description of Data

The dataset comprises several attributes, including customer ID, product ID, quantity, price, timestamp, payment method, store location, product category, discount applied, and total amount. This structured dataset provides granular information necessary for conducting a detailed analysis of Tesco's product performance. This dataset consists of 10 columns and 100000 rows.

No	Attributes	Explanation
1	Customer ID	unique identifier for each customer
2	Product ID	unique identifier for each product sold
3	Quantity	number of units purchased in each transaction
4	Price	unit price of the product
5	Timestamp	date and time of the transaction
6	Payment method	the method used for payment (cash, credit card, debit card, Paypal)
7	Store location	geographic location of the store where the transaction occurred
8	Product category	category or type of the product
9	Discount Applied	any discounts or promotions applied to the transaction
10	Total Amount	the total purchase amount including discount

2.0 Packages required

Pandas

Description: Pandas is a powerful data manipulation and analysis library. It provides data structures like Series and DataFrame, which are useful for handling structured data, performing operations like merging, grouping, and aggregation.

Usage: Reading and writing data (CSV, Excel, etc.), data cleaning, transformation, and aggregation.

NumPy

Description: NumPy is a library for numerical computing in Python. It provides support for arrays, matrices, and a large collection of mathematical functions to operate on these data structures.

Usage: Efficient numerical computations, handling large datasets, performing mathematical operations.

Matplotlib

Description: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for generating plots, histograms, bar charts, pie charts, and more.

Usage: Creating a wide range of static visualizations like line plots, scatter plots, and pie charts.

Seaborn

Description: Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Usage: Creating visually appealing and informative statistical plots such as heatmaps, violin plots, and pair plots.

Plotly

Description: Plotly is an interactive graphing library that makes it easy to create interactive plots and dashboards. It supports various types of plots, including scatter plots, bar charts, and pie charts, which can be embedded in web applications.

Usage: Creating interactive and web-based visualizations, dashboards, and complex plots.

SciPy

Description: SciPy is a library used for scientific and technical computing. It builds on NumPy and provides additional functionality for optimization, integration, interpolation, eigenvalue problems, and other advanced computations.

Usage: Performing scientific computations, advanced mathematical functions, and data analysis.

Scikit-learn

Description: Scikit-learn is a machine learning library in Python. It provides simple and efficient tools for data mining and data analysis, including various classification, regression, and clustering algorithms.

Usage: Implementing machine learning models, performing data preprocessing, and evaluating model performance.

Statsmodels

Description: Statsmodels is a library for statistical modeling and econometrics in Python. It provides classes and functions for estimating and testing statistical models.

Usage: Conducting statistical tests, estimating linear models, and performing time series analysis.

ipywidgets

Description: A library for building interactive widgets in Jupyter notebooks. It allows for the creation of dynamic user interfaces that can interact with data and visualizations.

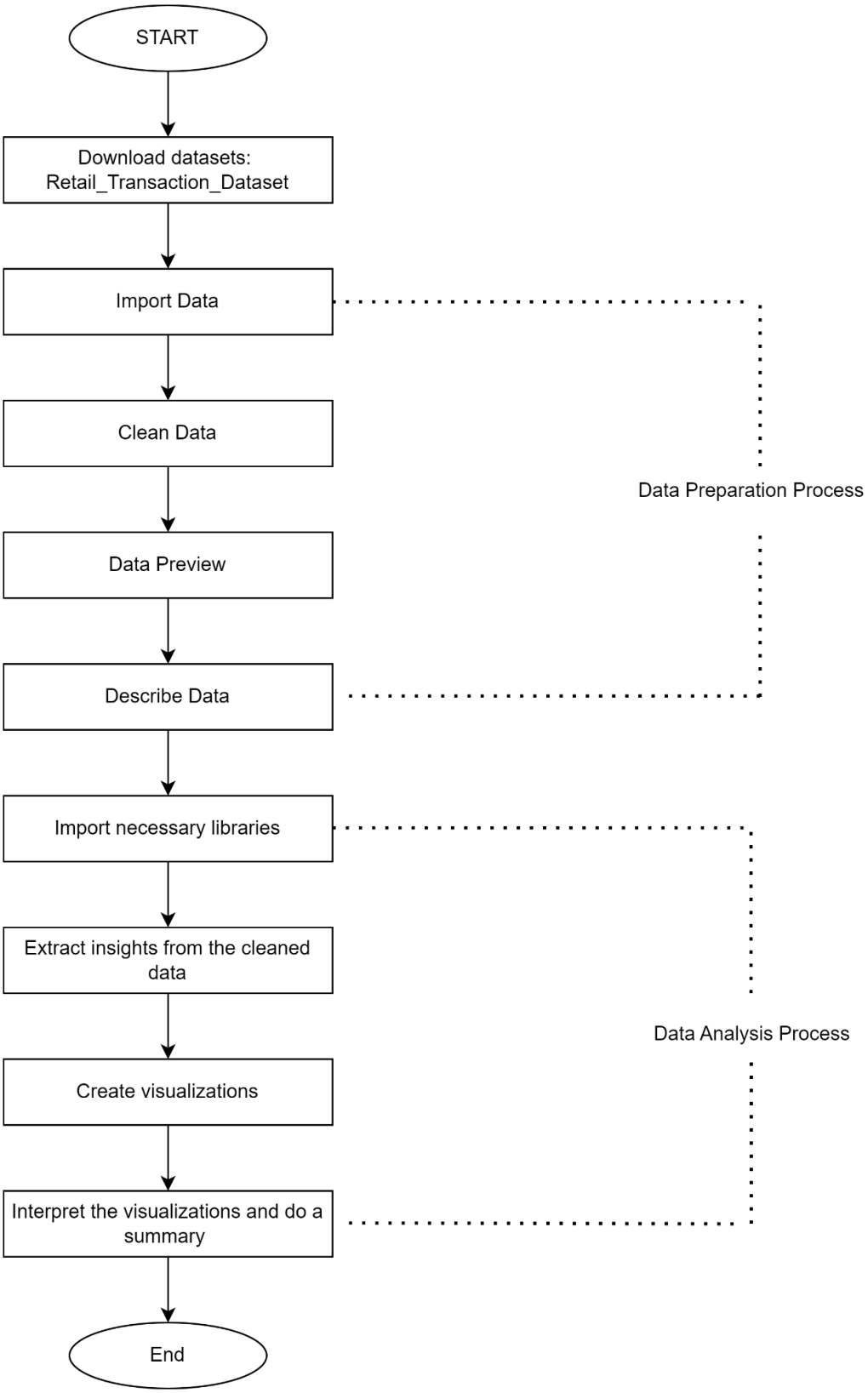
Usage: Creating interactive sliders, dropdowns, and other controls to filter and manipulate data interactively.

IPython.display

Description: An interactive command shell for Python that provides more features compared to the standard Python shell.

Usage: Displaying widgets, clearing output areas in notebooks.

3.0 Data preparation



1) Import the data

```
[ ]  import pandas as pd
     import numpy as np
```

```
[ ]  data = pd.read_csv("/content/Retail_Transaction_Dataset.csv")
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	109318	C	7	80.079844	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	18.677100	455.862764	2023-12-26	12:32:00
1	993229	C	4	75.195229	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.121365	258.306546	2023-08-05	00:00:00
2	579675	A	8	31.528816	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	15.943701	212.015651	2024-03-11	18:51:00
3	799826	D	5	98.880218	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	6.686337	461.343769	2023-10-27	22:00:00
4	121413	A	7	93.188512	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.030096	626.030484	2023-12-22	11:38:00
...	...	...	...	...	...	...	...	...	...	...	...
99995	726461	A	2	56.078258	Credit Card	3632 Darren Station Apt. 553\nEricaborough, RI...	Clothing	18.345145	91.581240	2023-07-17	16:59:00
99996	328056	A	6	88.516406	Credit Card	821 Taylor Shoals\nEvansville, IL 70845	Electronics	3.995541	509.878179	2023-05-30	09:04:00
99997	887304	B	4	72.385564	Credit Card	50653 Kara Lakes\nStephanieborough, RI 94492	Clothing	17.423979	239.092472	2023-08-25	07:59:00
99998	326401	C	5	66.542239	PayPal	18756 Mcfarland Way Suite 866\nBarnettside, PR...	Electronics	14.345018	284.983717	2024-02-05	19:45:00
99999	771566	C	5	38.087766	Debit Card	8046 Hull Drive\nPaulstad, GU 87218	Home Decor	2.966058	184.790305	2024-02-04	11:53:00

100000 rows × 11 columns

The initial stage in the data wrangling process is data import, which involves collecting raw data from multiple sources and bringing it into the workspace for further processing and analysis. For our case study, we import the data from the Kaggle platform. We download the dataset of Retail Transaction Dataset that is available in a compressed format which is a ZIP file. Then, we extract the contents to get the raw data files in CSV format. By using Python's Pandas library and Numpy library, we are able to read the CSV files into data frame objects, which allows us to easily manipulate and analyse the data.

2) Separate the data

```
data['TransactionDate'] = pd.to_datetime(data['TransactionDate'])

# Extract date and time components
data['Date'] = data['TransactionDate'].dt.date
data['Time'] = data['TransactionDate'].dt.time
data.drop(columns=['TransactionDate'], inplace=True)
data
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	109318	C	7	80.079844	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	18.677100	455.862764	2023-12-26	12:32:00
1	993229	C	4	75.195229	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.121365	258.306546	2023-08-05	00:00:00
2	579675	A	8	31.528816	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	15.943701	212.015651	2024-03-11	18:51:00
3	799826	D	5	98.880218	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	6.686337	461.343769	2023-10-27	22:00:00
4	121413	A	7	93.188512	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.030096	626.030484	2023-12-22	11:38:00
...	...	...	...	...	...	...	...	...	...	...	...
99995	726461	A	2	56.078258	Credit Card	3632 Darren Station Apt. 553\nEricaborough, RI...	Clothing	18.345145	91.581240	2023-07-17	16:59:00
99996	328056	A	6	88.516406	Credit Card	821 Taylor Shoals\nEvansville, IL 70845	Electronics	3.995541	509.878179	2023-05-30	09:04:00
99997	887304	B	4	72.385564	Credit Card	50653 Kara Lakes\nStephanieborough, RI 94492	Clothing	17.423979	239.092472	2023-08-25	07:59:00
99998	326401	C	5	66.542239	PayPal	18756 Mcfarland Way Suite 866\nBarnettside, PR...	Electronics	14.345018	284.983717	2024-02-05	19:45:00
99999	771566	C	5	38.087766	Debit Card	8046 Hull Drive\nPaulstad, GU 87218	Home Decor	2.966058	184.790305	2024-02-04	11:53:00

100000 rows × 11 columns

We separate the data of TransactionDate with two different columns to change the data into the correct data type which is Date and Time.

```
[ ]  data.head()
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	109318	C	7	80.079844	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	18.677100	455.862764	2023-12-26	12:32:00
1	993229	C	4	75.195229	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.121365	258.306546	2023-08-05	00:00:00
2	579675	A	8	31.528816	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	15.943701	212.015651	2024-03-11	18:51:00
3	799826	D	5	98.880218	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	6.686337	461.343769	2023-10-27	22:00:00
4	121413	A	7	93.188512	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.030096	626.030484	2023-12-22	11:38:00

The data.head() shows the initial rows of the data frame giving an overview of the contents of the dataset.

```
[ ]  data.tail()
```

```
[9]  data.tail()
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
99995	726461	A	2	56.078258	Credit Card	3632 Darren Station Apt. 553\nEricaborough, RI...	Clothing	18.345145	91.581240	2023-07-17	16:59:00
99996	328056	A	6	88.516406	Credit Card	821 Taylor Shoals\nEvansville, IL 70845	Electronics	3.995541	509.878179	2023-05-30	09:04:00
99997	887304	B	4	72.385564	Credit Card	50653 Kara Lakes\nStephanieborough, RI 94492	Clothing	17.423979	239.092472	2023-08-25	07:59:00
99998	326401	C	5	66.542239	PayPal	18756 Mcfarland Way Suite 866\nBarnettside, PR...	Electronics	14.345018	284.983717	2024-02-05	19:45:00
99999	771566	C	5	38.087766	Debit Card	8046 Hull Drive\nPaulstad, GU 87218	Home Decor	2.966058	184.790305	2024-02-04	11:53:00

The data.tail() shows the five final rows of the data frame that provide an overview of the last records in the dataset.

3) Data Information

```
[ ]  data.info()
```

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 11 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   CustomerID          100000 non-null  int64
 1   ProductID           100000 non-null  object
 2   Quantity            100000 non-null  int64
 3   Price               100000 non-null  float64
 4   PaymentMethod       100000 non-null  object
 5   StoreLocation       100000 non-null  object
 6   ProductCategory     100000 non-null  object
 7   DiscountApplied(%)  100000 non-null  float64
 8   TotalAmount         100000 non-null  float64
 9   Date                100000 non-null  object
 10  Time                100000 non-null  object
dtypes: float64(3), int64(2), object(6)
memory usage: 8.4+ MB
```

We use the data.info() method because it provides an overview of the dataframe's structure and properties. It was displaying the total number of rows and columns in the dataframe. It provides us with a quick view of the size of the dataset which is 10,000 entries in this case study. Then, the number of observations for each column will be shown by the count of non-null values, which is a list of numbers for each column. The data types of each column are displayed that provide us with information about the internal data storage, like the floats, integers, or other types of data. Lastly, the memory usage of the data frame is shown which is 8.4+ MB.

4) Describe the data

```
[ ]  data.describe()
```

	CustomerID	Quantity	Price	DiscountApplied(%)	TotalAmount
count	100000.000000	100000.000000	100000.000000	100000.000000	100000.000000
mean	500463.982180	5.009290	55.067344	10.020155	248.334955
std	288460.917524	2.579808	25.971567	5.779534	184.554792
min	14.000000	1.000000	10.000430	0.000046	8.274825
25%	250693.750000	3.000000	32.549474	5.001013	95.163418
50%	499679.000000	5.000000	55.116789	10.030353	200.368393
75%	751104.750000	7.000000	77.456763	15.018367	362.009980
max	999997.000000	9.000000	99.999284	19.999585	896.141242

This shows the descriptive statistics that summarize the central tendency, dispersion, and also shape of the numerical data in the data frame. It includes the statistics like count, mean, standard deviation, minimum, maximum, and the various percentiles which are 25%, 50%, and 75%.

5) Check null dataset

```
[ ]  data.isnull()
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
99995	False	False	False	False	False	False	False	False	False	False	False
99996	False	False	False	False	False	False	False	False	False	False	False
99997	False	False	False	False	False	False	False	False	False	False	False
99998	False	False	False	False	False	False	False	False	False	False	False
99999	False	False	False	False	False	False	False	False	False	False	False

100000 rows × 11 columns

Data.isnull() is used to identify the missing value within the data frame. It provides a data frame with the same form as the original, but with a boolean value of True or False that identifies whether each cell contains a missing value or not. In this case, True represents that a value is missing (NaN or None), whereas False indicates that a value is present. By doing

this, we can efficiently identify the columns and rows that have the missing values, and then we can choose the best method for dealing with the data like interpolation, imputation, or removal.

6) Check the sum of null value

```
[ ]  data.isnull().sum()
```

```
CustomerID            0
ProductID             0
Quantity              0
Price                 0
PaymentMethod         0
StoreLocation         0
ProductCategory       0
DiscountApplied(%)    0
TotalAmount           0
Date                  0
Time                  0
dtype: int64
```

The .sum() is applied to calculate the sum of True values in each column. Then, the result of data.isnull().sum is a Series that lists the total number of missing values for every data for each column. It is very helpful for quickly identifying the amount of missing data in the dataset and identifying columns that might require attention or more research. It also provides useful information about the quality of the data and influences decisions about data cleaning and preprocessing techniques.

7) **Check missing value**

```
missing_custID = data["CustomerID"].isnull().sum()
missing_custID
```

```
0
```

```
[15] missing_prodID = data["ProductID"].isnull().sum()
     missing_prodID
```

```
0
```

```
[16] missing_quantity = data["Quantity"].isnull().sum()
     missing_quantity
```

```
0
```

```
[17] missing_price = data["Price"].isnull().sum()
     missing_price
```

```
0
```

```
[18] missing_transData = data["Date"].isnull().sum()
     missing_transData
```

```
0
```

```
[19] missing_transData = data["Time"].isnull().sum()
     missing_transData
```

```
0
```

```
[20] missing_payMethod = data["PaymentMethod"].isnull().sum()
     missing_payMethod
```

```
0
```

```
[21] missing_storeLoc = data["StoreLocation"].isnull().sum()
     missing_storeLoc
```

```
0
```

```
[22] missing_prodCategory = data["ProductCategory"].isnull().sum()
     missing_prodCategory
```

```
0
```

```
[23] missing_discount = data["DiscountApplied(%)"].isnull().sum()
     missing_discount
```

```
0
```

```
missing_TotalAmnt = data["TotalAmount"].isnull().sum()
missing_TotalAmnt
```

```
0
```

We check the missing value one-by-one for each column.

8) Count

```
[ ]  data["CustomerID"].value_counts()
```

```
CustomerID
340516    4
717286    4
892820    4
397868    3
78182     3
          ..
645812    1
992548    1
180903    1
570896    1
771566    1
Name: count, Length: 95215, dtype: int64
```

We check the total number of customerID.

9) Find unique value

```
data["CustomerID"].unique()
```

```
array([109318, 993229, 579675, ..., 887304, 326401, 771566])
```

Here, we find all unique values of customer IDs in that column. This is very useful for us to understand the variety of customer IDs in the dataset and identify any patterns or inconsistencies in the data.

```
data["Date"].unique()
```

```
array([datetime.date(2023, 12, 26), datetime.date(2023, 8, 5),
       datetime.date(2024, 3, 11), datetime.date(2023, 10, 27),
       datetime.date(2023, 12, 22), datetime.date(2023, 8, 15),
       datetime.date(2023, 10, 11), datetime.date(2024, 2, 27),
       datetime.date(2023, 11, 5), datetime.date(2023, 9, 25),
       datetime.date(2023, 12, 29), datetime.date(2023, 12, 27),
       datetime.date(2024, 2, 8), datetime.date(2023, 9, 23),
       datetime.date(2024, 2, 28), datetime.date(2024, 2, 1),
       datetime.date(2024, 4, 28), datetime.date(2024, 4, 6),
       datetime.date(2024, 1, 21), datetime.date(2024, 4, 24),
       datetime.date(2023, 11, 28), datetime.date(2023, 10, 22),
       datetime.date(2023, 8, 2), datetime.date(2023, 6, 12),
       datetime.date(2024, 3, 24), datetime.date(2023, 11, 29),
       datetime.date(2024, 3, 17), datetime.date(2023, 5, 3),
       datetime.date(2024, 3, 12), datetime.date(2023, 6, 4),
       datetime.date(2023, 10, 3), datetime.date(2023, 12, 16),
       datetime.date(2023, 6, 20), datetime.date(2023, 12, 23),
       datetime.date(2024, 4, 20), datetime.date(2024, 1, 15),
       datetime.date(2023, 7, 9), datetime.date(2023, 5, 28),
       datetime.date(2023, 5, 25), datetime.date(2023, 10, 8),
       datetime.date(2023, 12, 13), datetime.date(2023, 10, 24),
       datetime.date(2024, 3, 25), datetime.date(2023, 12, 19),
       datetime.date(2023, 8, 12), datetime.date(2024, 1, 6),
       datetime.date(2023, 6, 16), datetime.date(2023, 6, 26),
       datetime.date(2023, 5, 8), datetime.date(2024, 3, 29),
       datetime.date(2023, 11, 27), datetime.date(2024, 4, 26),
       datetime.date(2023, 10, 1), datetime.date(2023, 5, 20),
       datetime.date(2023, 5, 19), datetime.date(2023, 8, 24),
       datetime.date(2024, 3, 15), datetime.date(2024, 4, 16),
       datetime.date(2023, 9, 7), datetime.date(2024, 3, 31),
       datetime.date(2023, 7, 1), datetime.date(2023, 10, 6),
       datetime.date(2023, 12, 24), datetime.date(2024, 2, 25),
       datetime.date(2023, 11, 22), datetime.date(2024, 3, 2),
```

We find all unique values of Date for this column.

```
data["Time"].unique()
```

```
array([datetime.time(12, 32), datetime.time(0, 0), datetime.time(18, 51),
       ..., datetime.time(12, 44), datetime.time(7, 48),
       datetime.time(18, 42)], dtype=object)
```

We find all unique values of Time in that column.

10) Check the total of duplicated data

```
data.duplicated().sum()
```

```
0
```

We use data.duplicated().sum() to find the total duplicate rows in the data frame. It is very helpful in identifying and quantifying the amount of duplication in the dataset.

11) Rounded the value of data

```
data["Price"] = round(data["Price"], 2) #2 decimal places for price
data["DiscountApplied(%)"] = round(data["DiscountApplied(%)"], 0) #no decimal places for discount
data["TotalAmount"] = round(data["TotalAmount"], 2) #2 decimal places for TotalAmount
data
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	109318	C	7	80.08	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	19.0	455.86	2023-12-26	12:32:00
1	993229	C	4	75.20	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.0	258.31	2023-08-05	00:00:00
2	579675	A	8	31.53	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	16.0	212.02	2024-03-11	18:51:00
3	799826	D	5	98.88	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	7.0	461.34	2023-10-27	22:00:00
4	121413	A	7	93.19	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.0	626.03	2023-12-22	11:38:00
...	...	...	...	...	...	...	...	...	...	...	...
99995	726461	A	2	56.08	Credit Card	3632 Darren Station Apt. 553\nEricaborough, RI...	Clothing	18.0	91.58	2023-07-17	16:59:00
99996	328056	A	6	88.52	Credit Card	821 Taylor Shoals\nEvansville, IL 70845	Electronics	4.0	509.88	2023-05-30	09:04:00
99997	887304	B	4	72.39	Credit Card	50653 Kara Lakes\nStephanieborough, RI 94492	Clothing	17.0	239.09	2023-08-25	07:59:00
99998	326401	C	5	66.54	PayPal	18756 Mcfarland Way Suite 866\nBarnettside, PR...	Electronics	14.0	284.98	2024-02-05	19:45:00
99999	771566	C	5	38.09	Debit Card	8046 Hull Drive\nPaulstad, GU 87218	Home Decor	3.0	184.79	2024-02-04	11:53:00

100000 rows × 11 columns

Here, we want to round the value of Price to two decimal places, no decimal places for Discount Applied (%) and also two decimal places for Total Amount. This process assists in maintaining accuracy and consistency in the data, especially when we are working with the numerical values that need to be precise to certain specifications.

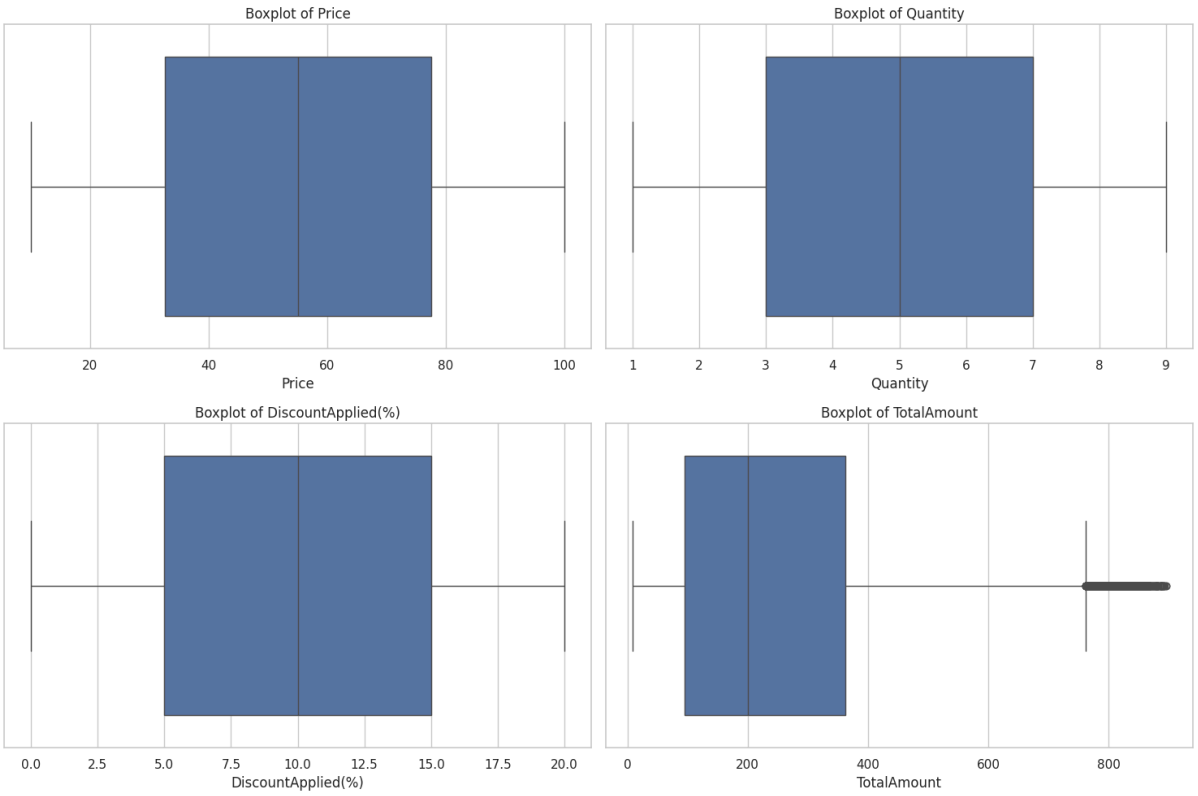
12) Check for the outlier

```
[ ]  !pip install pandas matplotlib seaborn

     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

We install required packages to visualized the outlier.

```
[ ]  # style of the visualization
     sns.set(style="whitegrid")

     #  subplots
     fig, axes = plt.subplots(2, 2, figsize=(15, 10))

     # Boxplot for Price
     sns.boxplot(ax=axes[0, 0], x=data['Price'])
     axes[0, 0].set_title('Boxplot of Price')

     # Boxplot for Quantity
     sns.boxplot(ax=axes[0, 1], x=data['Quantity'])
     axes[0, 1].set_title('Boxplot of Quantity')

     # Boxplot for DiscountApplied(%)
     sns.boxplot(ax=axes[1, 0], x=data['DiscountApplied(%)'])
     axes[1, 0].set_title('Boxplot of DiscountApplied(%)')

     # Boxplot for TotalPrice
     sns.boxplot(ax=axes[1, 1], x=data['TotalAmount'])
     axes[1, 1].set_title('Boxplot of TotalAmount')

     # layout
     plt.tight_layout()
     plt.show()
```

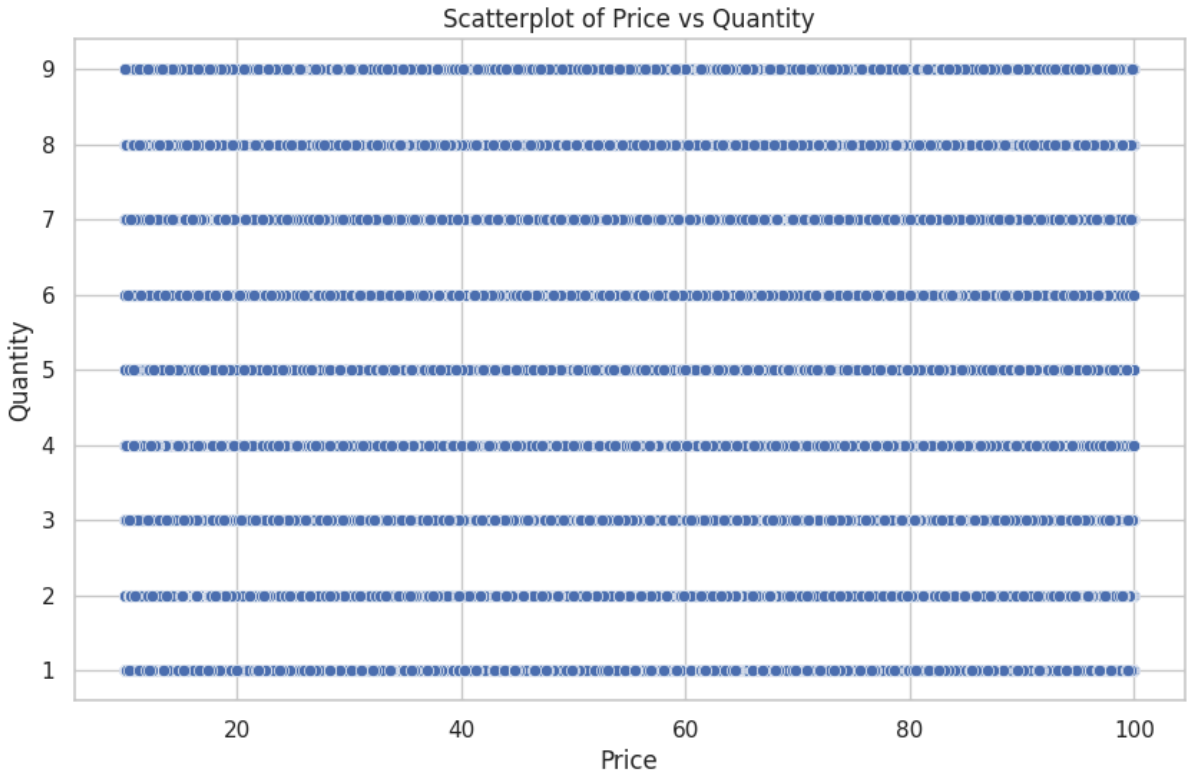
This is the coding to display the boxplot to check for the outlier. This is one of the way to check for the outlier.



From this three boxplot graph, we can see that only TotalAmount data has high outliers. But we didn't remove it since we are not going to do prediction for our project, we only want to see the relationship between the entities.

```
# Scatterplot for Price vs Quantity
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Price', y='Quantity', data=data)
plt.title('Scatterplot of Price vs Quantity')
plt.show()

# Scatterplot for Price vs DiscountApplied(%)
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Price', y='DiscountApplied(%)', data=data)
plt.title('Scatterplot of Price vs DiscountApplied(%)')
plt.show()

# Scatterplot for Price vs TotalPrice
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Price', y='TotalAmount', data=data)
plt.title('Scatterplot of Price vs TotalAmount')
plt.show()
```

Scatter plot is also one of the ways to check for the outlier, we check for Price with Quantity, price with DiscountApplied(%), and Price with TotalAmount.



This relationship between the Price and Quantity does not shows any outlier exist.

same goes to Price vs DiscountApplied(%), it does not seem there is no outlier exist.

Price vs TotalAmount does show there is an outlier exist, and we don't delete it, we just want to check for it existent.

13) Data Preview of Cleaned Data

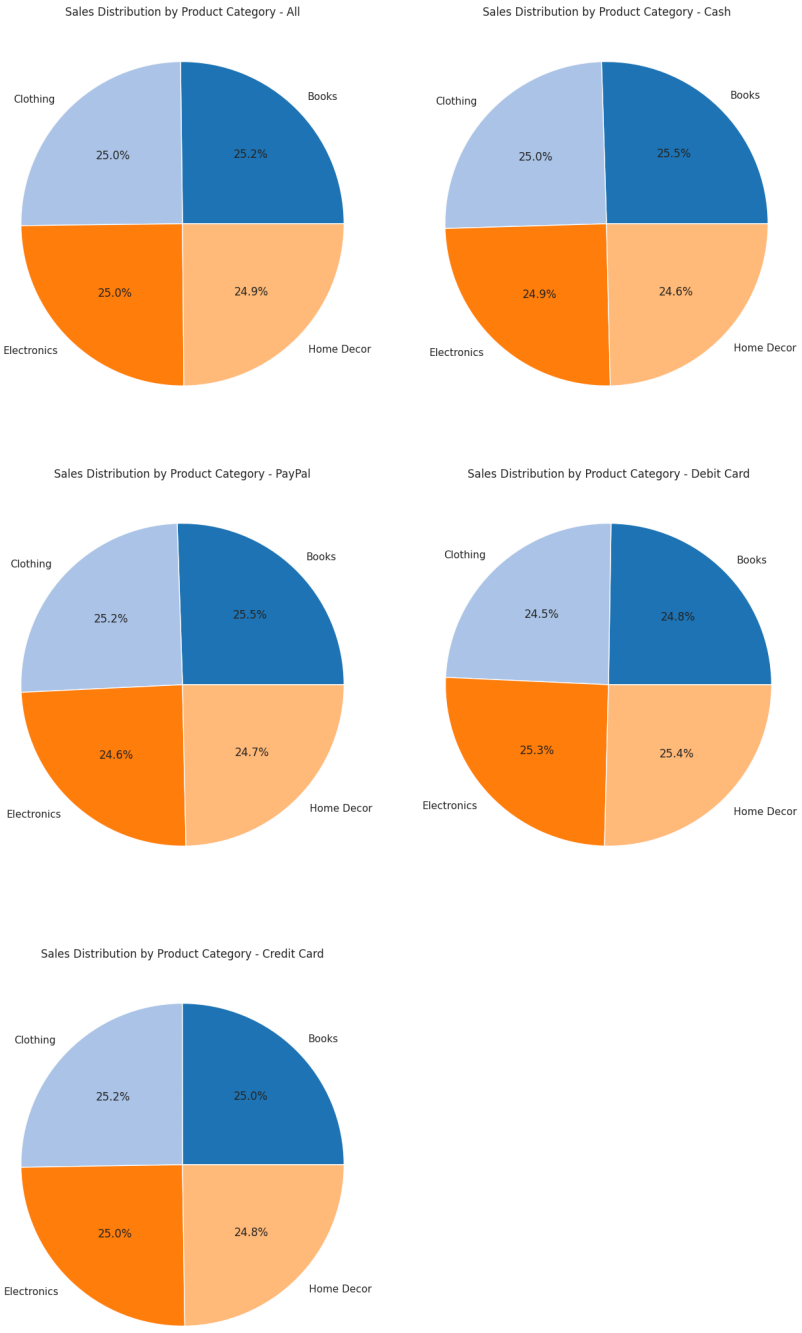
```
[34] data_preview= data.head()
     data_preview
```

	CustomerID	ProductID	Quantity	Price	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)	TotalAmount	Date	Time
0	109318	C	7	80.08	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	19.0	455.86	2023-12-26	12:32:00
1	993229	C	4	75.20	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.0	258.31	2023-08-05	00:00:00
2	579675	A	8	31.53	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	16.0	212.02	2024-03-11	18:51:00
3	799826	D	5	98.88	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	7.0	461.34	2023-10-27	22:00:00
4	121413	A	7	93.19	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.0	626.03	2023-12-22	11:38:00

At the end of this cleaning process, we preview the cleaned version of our data. The data is ready to be visualized.

4.0 Exploratory Data Analysis

4.1 VISUALIZATION 1



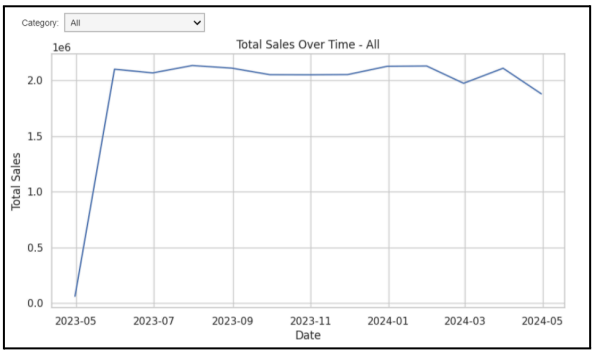
Based on the pie chart visualization of the retail transaction dataset, it shows that the distribution of spending across product categories remains relatively consistent regardless of

the payment method used. Across all payment methods, clothing, books, electronics, and home decor each represent approximately a quarter of the total spending, with only slight variations between categories.

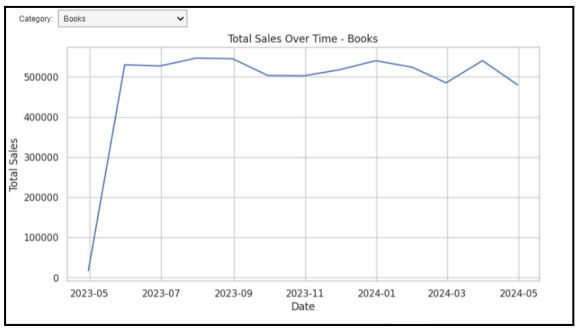
- Consumers use debit cards, they are slightly more likely to spend more on electronics and home decor, and to spend less on books and clothing.
- When it comes to cash expenditures, electronics and home decor are slightly more frequently purchased than books and clothing.
- Credit card payments users reveal a more evenly distributed spending across product categories, with almost equal amounts going towards books, electronics, clothing, and home decor.
- On the other hand, when compared to other payment options, PayPal users typically spend a little larger portion of their budget on clothing.

These little variations imply that consumer preferences, depending on the features and ease of each payment option, may affect purchasing habits. Overall, the pie chart is a useful visual tool for understanding customer behaviour within the retail dataset by clearly illustrating how spending is allocated among product categories and payment methods.

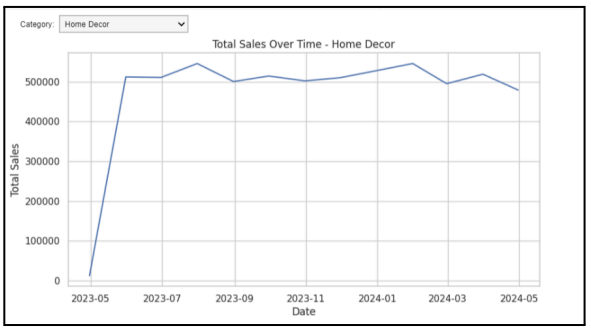
4.2 VISUALIZATION 2



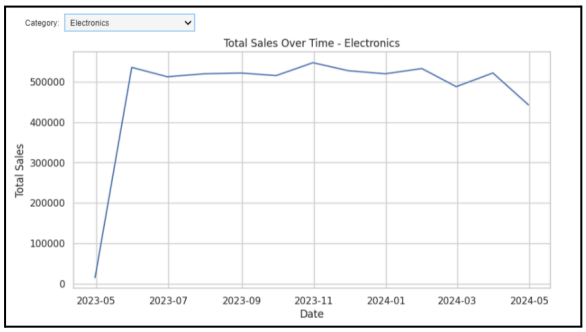
Total Sales Over Time - All



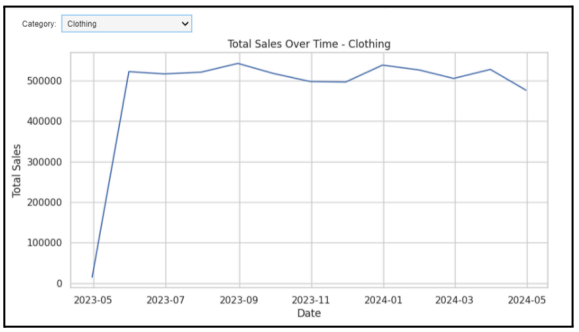
Total Sales Over Time - Books



Total Over Time - Home Decor



Total Sales Over Time - Electronics

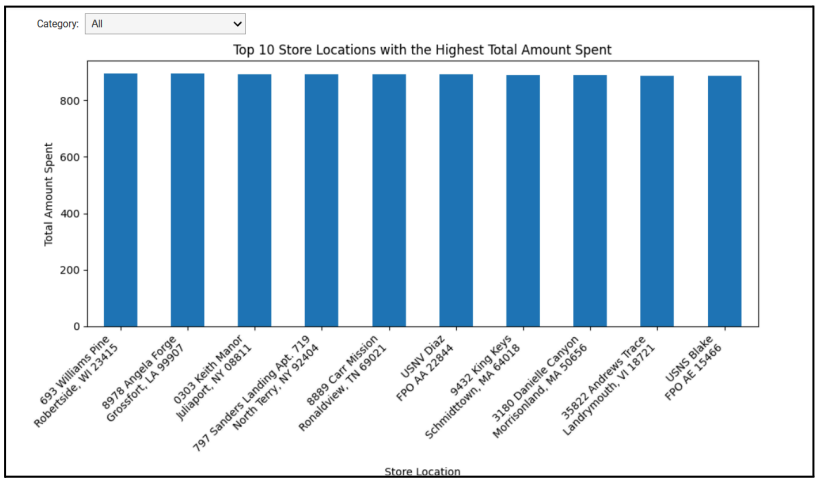


Total Sales Over Time - Clothing

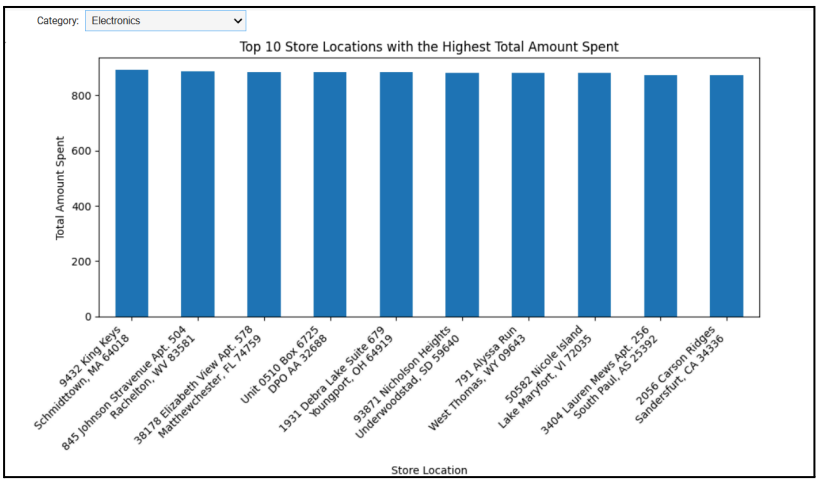
Based on the line chart above, the "Total Sales Over Time" for various product categories, which include All, Books, Home Deco, Electronics, and Clothing, it shows the sales trends for these categories over a given time period. The line chart labeled "All" shows the total sales across all categories, highlighting overall sales patterns like seasonal trends, peak levels, and troughs. Beside that, the individual lines chart for Books, Home Deco, Electronics, and Clothing show sales trends for each category that allows for comparison of the performance. For example, an upward trend in the Electronics line indicates increased

sales in that category, whereas a decline in the Home Decor line chart indicates a decrease in sales. Next, seasonal patterns can be seen in peaks and troughs, such as increased sales in the Clothing category during the winter. In addition, the sudden spikes or drops in any line for any category can indicate significant events, including major sales or product launches. By comparing the slopes and positions of various lines, the relative performance of each category can be determined. A consistent upward or downward trend of sales over time indicates long-term growth or decrease within the category. This line graph is very useful for understanding the sales dynamics in general, planning supplies, adjusting marketing strategies, and making profitable business choices based on the performance of various product categories over time.

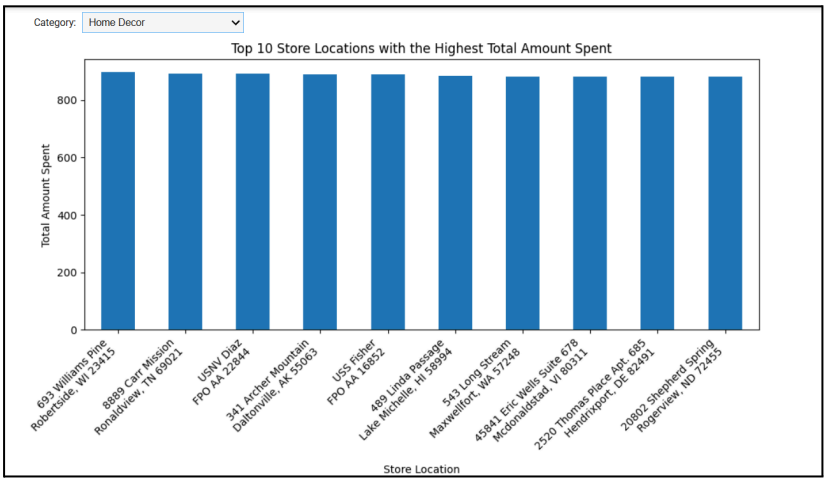
4.3 VISUALIZATION 3



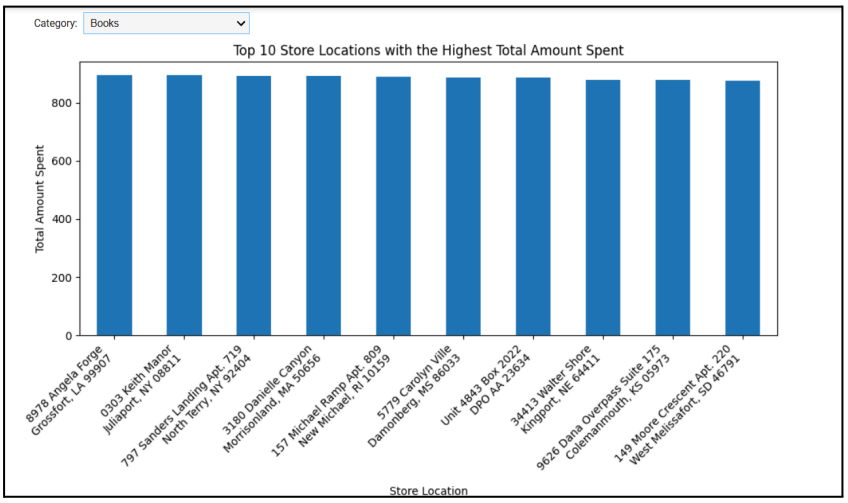
Top 10 Store Locations with Highest Total Amount Spent - All



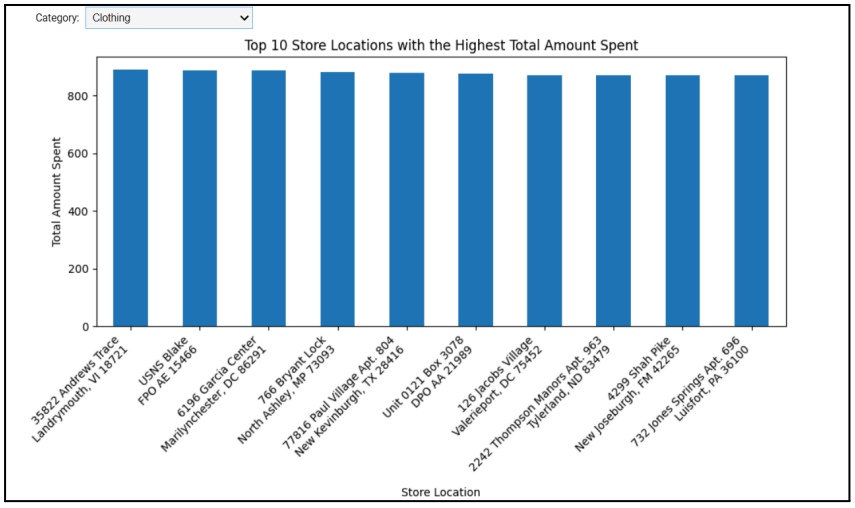
Top 10 Store Locations with Highest Total Amount Spent - Electronics



Top 10 Store Locations with Highest Total Amount Spent - Home Decor



Top 10 Store Locations with Highest Total Amount Spent - Books



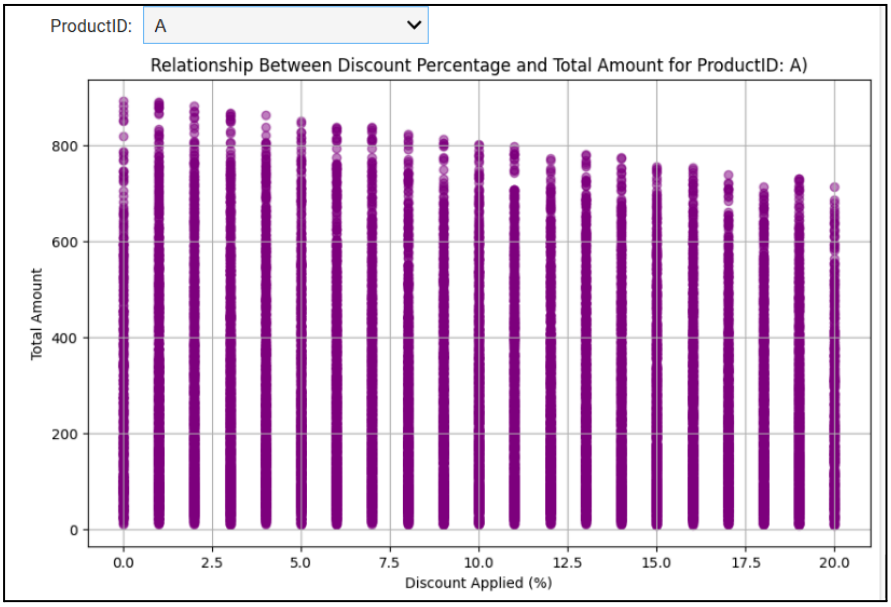
Top 10 Store Locations with Highest Total Amount Spent - Clothing

The bar charts above list the top 10 store locations with the highest total spending across various product categories. Each bar represents a different store location, with the amount spent being close to 900 units, indicating high and consistent spending across these top locations. The first chart highlights the top 10 stores with the highest sales in all product categories, with the store at 3693 Williams Pine Robertside, WI 23415 leading the list. The close totals among these stores suggest that they might be the primary shopping options in their respective areas, driving more customers to make purchases there. This lack of competition likely contributes to the high and consistent sales volumes. These stores appear to be strategically located in regions with limited alternatives, ensuring a steady flow of customers and reflecting a large and loyal customer base that relies on these stores for a wide

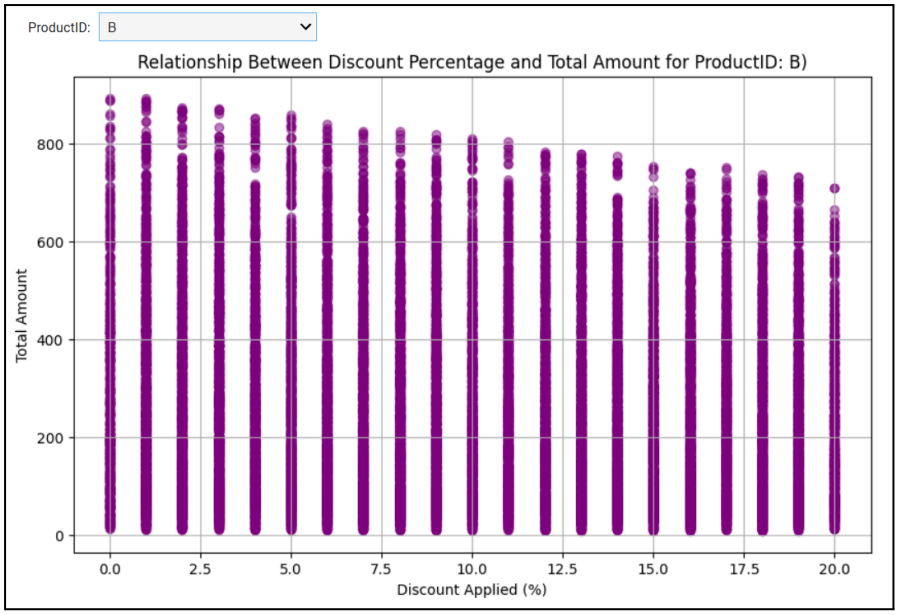
range of products. This information is valuable for understanding market dynamics and can inform future decisions on store locations and marketing strategies.

Interestingly, each product category has a different store leading the top, showcasing the diversity in store performance across different product lines. In the electronics category, the store at 9432 King Keys, Schmidtown, MA 64018, leads the sales. For home decor, the top performer is the store at 693 Williams Pine, Robertside, WI 23415. The store at 8978 Angela Forge, Grossfort, LA 99907, has the highest sales in the books category. In the clothing category, the leading store is located at 35822 Andrews Trace, Landrymouth, VI 18721. Each of these stores excels in a specific category, showcasing their strong market presence and the diverse product demands across different locations.

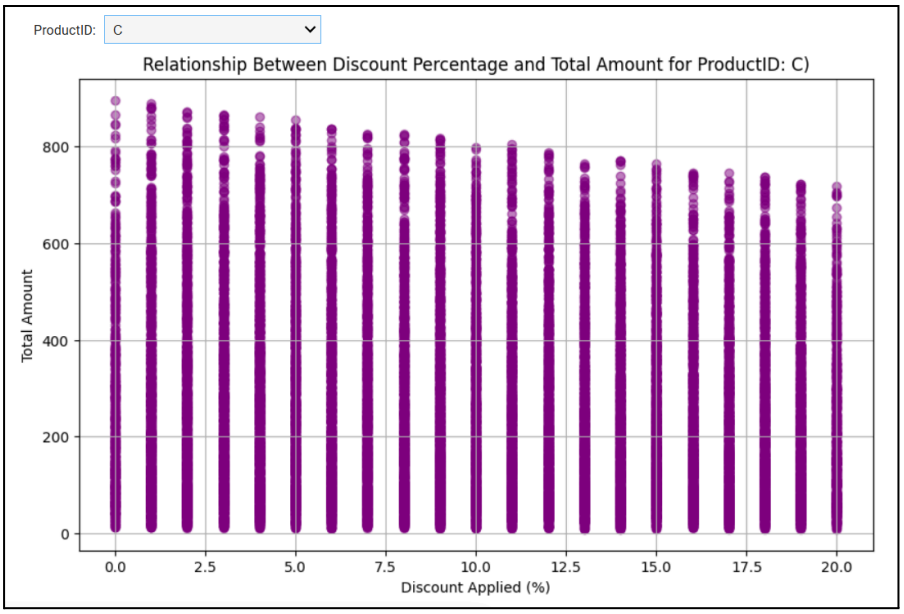
4.4 VISUALIZATION 4



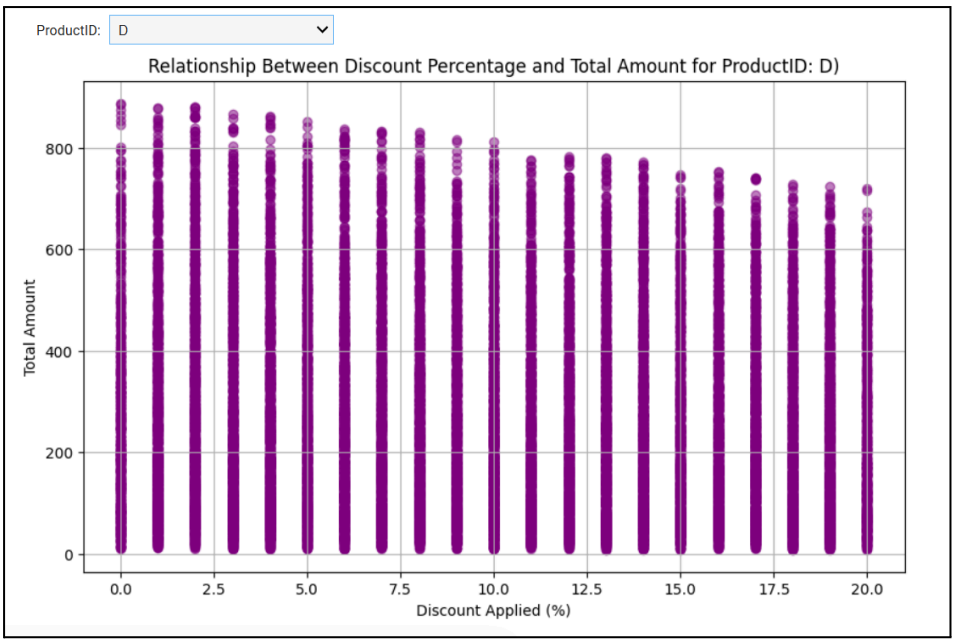
Relationship between Discount Percentage and Total Amount for ProductID - A



Relationship between Discount Percentage and Total Amount for ProductID - B



Relationship between Discount Percentage and Total Amount for ProductID - C



Relationship between Discount Percentage and Total Amount for ProductID - D

This scatter plot shows the relationship between discount applied which is in percentage and total amount spent for 4 different products according to its ProductID, A,B,C and D. In all these graphs for all ProductID, the discount applies at regular intervals ranging from 0% to 20% in increments of 2.5%. The x-axis represents the discount applied (%) while y-axis represents the total amount spent. For all products, the total amount ranges widely from near 0 to over 800. The vertical scatter plot at each discount level indicates that these are fixed discount points rather than continuous values. Each dot on the graph represents a transaction where a specific discount percentage was applied and the corresponding total amount spent.

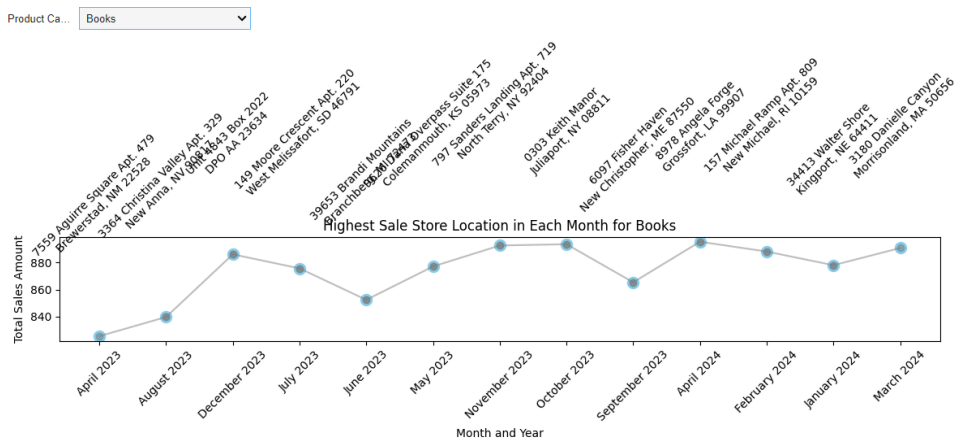
The fixed discount intervals shows that the structured discounting strategy is evident in both graphs, reflecting a systematic approach to apply discount. There is a significant variability in the total amount spent across all discount levels for all products, indicating diverse purchasing behaviours among customers. But, in all the graphs do not exhibit a straightforward linear relationship between the discount percentage and the total amount spent.

For Product C, there is no apparent trend showing that higher discounts which lead to a consistently higher or lower total amount. The total amounts are widely spread at each discount level. For product A, there is a slight trend suggesting that higher discounts,

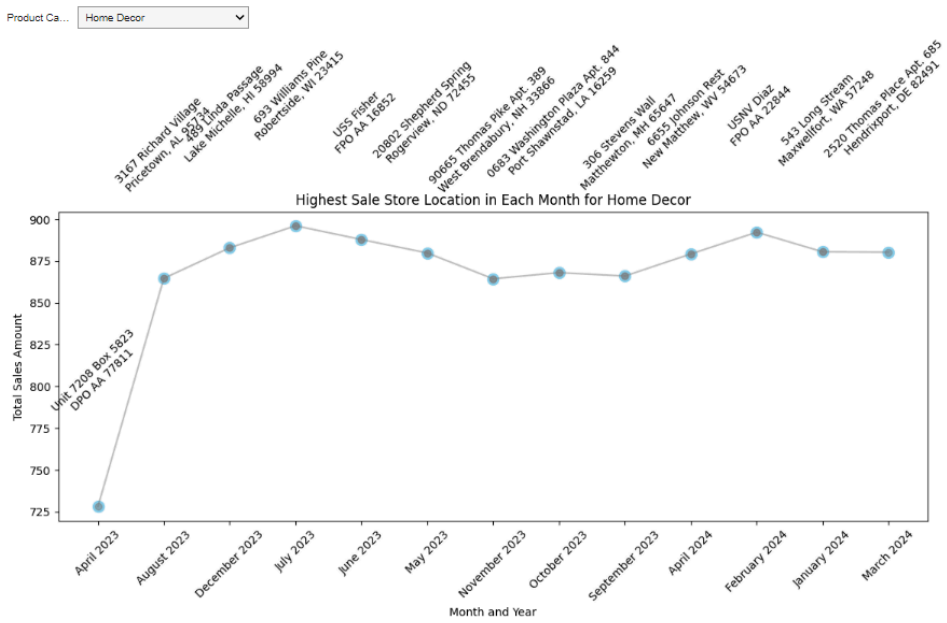
especially from 10% to 20%, are associated with slightly lower total amounts. This indicates that for Product A, increasing discounts might be leading to less total spending or smaller transactions. For Product D, there is a noticeable downward trend in the total amount as the discount percentage increases. Higher discounts tend to be associated with lower total amounts spent, though there is still significant variability at each discount level. Lastly, for product B there is an inverse relationship between the discount percentage and the total amount for ProductID B, meaning that higher discounts generally lead to lower total amounts spent.

4.5 VISUALIZATION 5

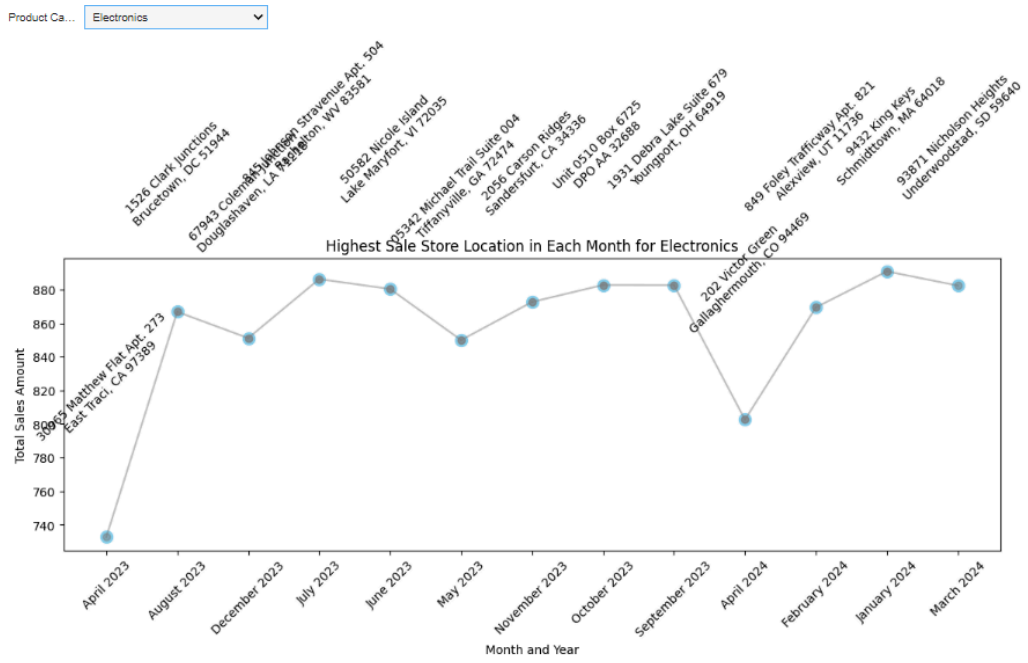
Store Location with The Highest Total Sales Amount in Each Month for All Category



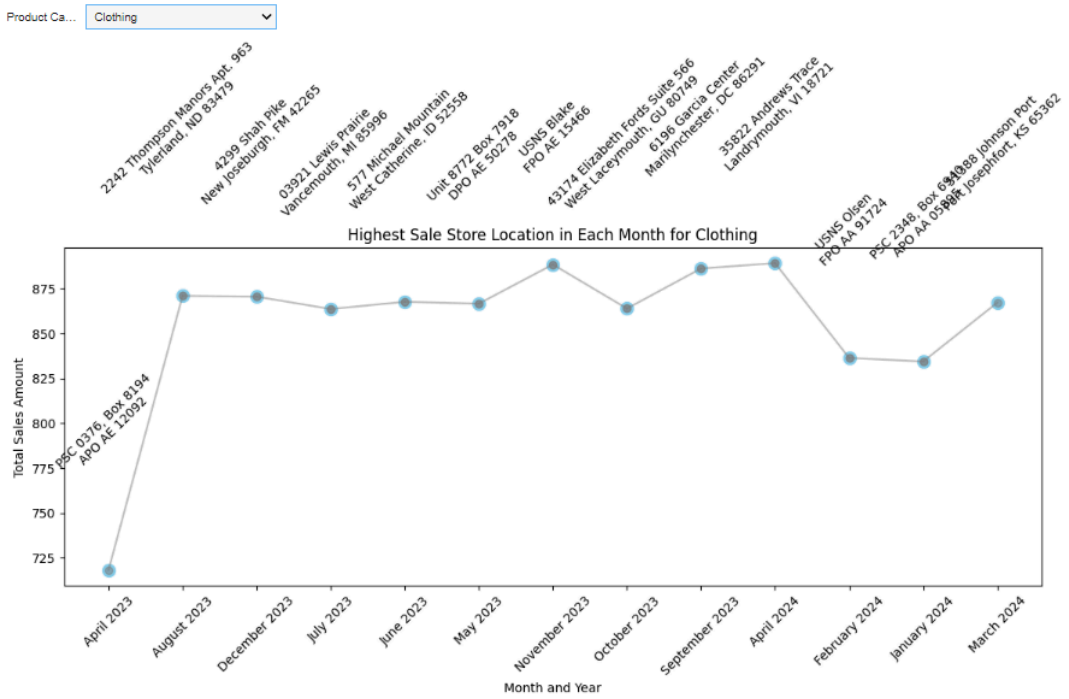
Store Location with The Highest Total Sales Amount in Each Month for Books Category



Store Location with The Highest Total Sales Amount in Each Month for Home Decor Category



Store Location with The Highest Total Sales Amount in Each Month for Electronics Category



Store Location with The Highest Total Sales Amount in Each Month for Clothing Category

The line graphs above show the store location with the highest total sales amount in each month across all categories; all, books, electronics, home decor, clothing . The data is recorded in the time period from April 2023 until April 2024. Each point represents a different store location, with the amount spent close to 900 units, indicating high and consistent spending.

From the first visualization, it shows the store location with the highest total sales amount in each month for all categories. For the first quarter 2024, January, February and March, there is a significant value of total sales amount, which is above 880 units. There is no big difference for the time period (first quarter 2024). Store location 8978 Angela Forge Grossfort,LA 99907 has the highest total sales amount which is in April 2024.

For books category, store located at 7559 Aguirre Square apt. 479 Brewerstad, NM 22528 recorded the lowest total sales amount which is less than 840. From May 2023 to June 2023, the total sales amount is decreasing. This may be caused by educational institutions

having their holiday due to summer. That explains why the total sales amount dropped because summer starts in June.

The third visualization shows the highest sale store location in each month for home decor category. From April 2023 until July 2023, the total sales amount is increasing from month to month. Store located at Unit 7208 Box 5823 DPO AA 77811 recorded the lowest total sales amount. The highest total sales amount recorded is in July 2023 at 693 Williams Pine Roberside, WI 23415. This may be caused by Independance Day being in July, therefore, there would be many sales and promotions for home decor in conjunction with the USA Independence Day.

From the fourth visualization, the highest total sales amount for the electronics category is in January 2024 at 9432 King Keys Schmidttown, MA 64018. Many people may buy new electronics since new year means the latest phones and there would be a lot of sales and promotion.

For the highest sale store location in each month for clothing category, 35822 Andrews Trace Landrymouth, VI 18721 recorded the highest sale. April marks the beginning of the spring season, as the weather changes, people tend to buy new clothes for the upcoming season. This may be the reason why in April 2024, the total sales amount recorded the highest.

5.0 Summary

Retail businesses are confronted with vast amounts of transactional data in today's digital landscape, offering valuable insights into consumer behavior and market trends. This project focused on examining Tesco's transaction data to identify important patterns and trends for business optimization. Despite facing challenges such as data inconsistencies and unstructured information, our project effectively prepared the data, enabling the extraction of meaningful insights. We began by importing raw data from various sources into our workspace, utilizing Python's Pandas and Numpy libraries to read the CSV files containing the retail transaction data. After separating the TransactionDate data into two columns to ensure the correct data type, we generated descriptive statistics to summarize the central tendency, dispersion, and shape of the numerical data. We then systematically checked for missing values, identified and quantified duplicate rows, and rounded numeric values for accuracy. Utilizing box plots and scatter plots, we visualized outliers in the data, with a particular focus on Total Amount. Although outliers were observed, they were retained as our project did not involve predictive analysis. Finally, we previewed the cleaned data, ensuring it was ready for visualization and analysis, laying the foundation for extracting valuable insights and trends to optimize business strategies. By analyzing this data, we uncovered essential insights such as identifying best-selling products, evaluating discount strategies' impact on consumer behavior, and understanding purchasing patterns among customers.

From visualization, we could get a lot of meaningful insights. For example, our pie chart visualization provided a clear snapshot of spending distribution across different product categories, revealing consistent patterns regardless of the payment method used. We observed that certain payment methods influenced spending habits, with customers using debit cards showing preferences for electronics and home decor, while cash payments were more inclined towards electronics and home decor. Credit card users demonstrated a more balanced spending pattern across various product categories, with PayPal users showing a preference for clothing purchases. These findings underscore the importance of understanding consumer payment preferences in tailoring effective marketing strategies.

Furthermore, our line chart visualizations tracked total sales over time across different product categories, offering insights into seasonal trends, peak periods, and potential market fluctuations. For instance, we noted increased sales in the clothing category during the winter months, indicating a correlation between seasonal changes and consumer purchasing behavior. Additionally, sudden spikes or drops in sales volumes could signal significant

events such as promotional campaigns or product launches, highlighting the need for agile marketing strategies to capitalize on market opportunities.

Our bar chart visualizations identified the top-performing store locations based on total sales across various product categories, shedding light on regional market dynamics and customer preferences. Interestingly, while certain stores dominated specific product categories, indicating their niche market presence, others excelled in overall sales, suggesting a broader appeal to diverse consumer segments. These insights can inform strategic decisions regarding store expansions, inventory management, and targeted marketing efforts tailored to local consumer preferences.

Moreover, our scatter plot visualizations explored the relationship between discount percentages and total spending for different product categories, revealing nuanced consumer behavior patterns. While some products exhibited a clear inverse relationship between discounts and total spending, indicating price sensitivity among consumers, others displayed more varied responses, highlighting the need for dynamic pricing strategies tailored to specific product categories and consumer segments.

Lastly, our line graph visualizations highlighted the store locations with the highest total sales amounts in each month across all product categories, offering insights into regional sales trends and market demand fluctuations. By identifying peak sales periods and correlating them with seasonal events or promotional activities, retailers can optimize inventory stocking levels, staffing requirements, and marketing campaigns to maximize sales opportunities and enhance overall customer satisfaction.

In conclusion, the combination of these visualizations provides a comprehensive understanding of Tesco's transaction data landscape, empowering retailers to make data-driven decisions to optimize inventory management, tailor marketing strategies, and enhance overall customer satisfaction. By leveraging these insights, Tesco can gain a competitive edge in the dynamic retail market, ensuring long-term success and sustainability in today's digital age.

6.0 References

Assosia. (2024, February 29). *Retail Analysis | Data Harnessing & Insight Reporting | Assosia*. https://www.assosia.com/product/retail-analysis

Hickins, M. (2023, March 17). *What is Retail analytics? The ultimate guide*. https://www.oracle.com/retail/what-is-retail-analytics/

Majumder, P. (2023, September 25). *Guide to Create Interactive Plots with Plotly Python*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/interactive-plots-in-python-with-plotly-a-complete-guide/

Unproven. (n.d.). *How do I generate an interactive plot in Python? : r/learnpython*. https://www.reddit.com/r/learnpython/comments/17u3z37/how_do_i_generate_an_interactive_plot_in_python/

7.0 Appendix

Google drive:

-Dataset Retail Transaction

-Link Google Docs

https://drive.google.com/drive/folders/17IJgmT7oh_9Ym8HUH9CqXVGDWcxlp0uE?usp=sharing

-Link Google Colab

https://colab.research.google.com/drive/1DrSyH53JX2_khCmTIEUScCx9J2Fi9n0B?usp=sharing