MCS 7103: Machine Learning

Emmanuel Ejeu | 2024/HD05/21922U | 2400721922 | MCSC

**Assignment 1: Exploratory Data Analysis Process**

Based on last week's lecture, your assignment is to perform Exploratory Data Analysis Process (DAP) on your dataset and write a report that is at least 3 pages long. You can write as many pages as you need. The report needs to be clear and follow a step-by-step process.

1. Present your question before and after, then the answers. The answers can be before the next step (Data Wrangling), every process or after the full process. The question that helped you select the dataset should differ from the questions you are using for the rest of the process. I have general questions to guide you through your analysis.
2. Perform in-depth data wrangling.
3. Provide a well-detailed Exploratory Data Analysis (EDA).
4. Draw some conclusions based on the performed EDA.
5. Communicate the findings in a detailed report.

If you don't have one, you must create a GitHub account and a public repository to upload your code. Everything in your report should be visible in the code. In addition to the code, add the report to the repository as well.

Please provide your name, registration number, and GitHub repository in this Excel sheet.

The assignment deadline is next Wednesday, the 11th, midnight(00:00). Keep this in mind, don't wait for the final day, the assignment is not not easy.

Note: Please refrain from using AI-generated text as it can be identified.

https://github.com/Ejeus/EDA-Retail-Sales-Data-with-Seasonal-Trends-Marketing.git

SOLUTIONS

**Title**

Retail Sales Data with Seasonal Trends & Marketing

**Link of Dataset**

<https://www.kaggle.com/datasets/abdullah0a/retail-sales-data-with-seasonal-trends-and-marketing/data>

**Citation**

Abdullah. (2024). Retail Sales Data with Seasonal Trends & Marketing [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/9349466

**Other Reference used for learning**

https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/

**About Dataset**

This dataset provides detailed insights into retail sales, featuring a range of factors that influence sales performance. It includes records on sales revenue, units sold, discount percentages, marketing spend, and the impact of seasonal trends and holidays.

**Key Features of Dataset**

The dataset provided is a retail sales dataset, containing 30,000 rows and 11 columns. Below are the key features in the dataset;

1. **Sales Revenue (USD):** Total revenue generated from sales.
2. **Units Sold:** Quantity of items sold.
3. **Discount Percentage:** The percentage discount applied to products.
4. **Marketing Spend (USD):** Budget allocated to marketing efforts.
5. **Store ID:** Identifier for the retail store.
6. **Product Category:** The category to which the product belongs (e.g., Electronics, Clothing).
7. **Date:** The date when the sale occurred.
8. **Store Location:** Geographic location of the store.
9. **Day of the Week:** Day when the sale took place.
10. **Holiday Effect:** Indicator of whether the sale happened during a holiday period.

**Use Cases of the Dataset**

1. **Predictive Modeling:** Build models to forecast future sales based on historical data.
2. **Marketing Analysis:** Evaluate the effectiveness of marketing spend and discount strategies.
3. **Seasonal Trend Analysis:** Examine how different seasons and holidays impact sales.
4. **Revenue Optimization:** Identify strategies to optimize pricing and marketing for increased revenue.

**Question 1**

Main Research Question

What factors drive the sales performance of stores in different geographic locations?

Follow-up questions

1. What is the relationship between marketing spend, discount percentage, and units sold?
2. Are there specific product categories or store locations that perform better?
3. Does the day of the week or holiday effect influence sales performance?
4. What is the distribution of units sold, sales revenue, discount percentage, marketing spend, sales by day of the week, top store location, and sales by product category?
5. Marketing Spend vs. Units Sold: Does increased marketing lead to more units sold?
6. Discount Percentage vs. Sales Revenue: What impact does discounting have on sales revenue?
7. Holiday Effect vs. Sales Revenue: Do holidays drive higher sales?
8. Marketing Spend vs. Sales Revenue: Does increased marketing lead to more units revenue?
9. Sales Revenue by Day of the Week: Which days of the week generate the highest revenue?
10. Top Sales Revenue by Store Location: Which locations bring what revenue threshold?
11. Sales Revenue by Product Category: What are the most sold products?
12. Correlation Matrix (Heatmap): How do different factors relate to each other?
13. Heatmap of Sales Revenue by Day of the Week and Product Category?

**Question 2**

Performing in-depth Data wrangling

Data wrangling, also known as data cleaning or data preprocessing, is the process of transforming raw data into a clean and usable format for analysis. This typically involves handling missing values, correcting data types, removing duplicates, encoding categorical variables, and addressing other inconsistencies.

There was no data inconsistency as shown by the results below

Store ID 0

Product ID 0

Date 0

Units Sold 0

Sales Revenue (USD) 0

Discount Percentage 0

Marketing Spend (USD) 0

Store Location 0

Product Category 0

Day of the Week 0

Holiday Effect 0

dtype: int64

**Information About Data**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 30000 entries, 0 to 29999

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Store ID 30000 non-null object

1 Product ID 30000 non-null int64

2 Date 30000 non-null object

3 Units Sold 30000 non-null int64

4 Sales Revenue (USD) 30000 non-null float64

5 Discount Percentage 30000 non-null int64

6 Marketing Spend (USD) 30000 non-null int64

7 Store Location 30000 non-null object

8 Product Category 30000 non-null object

9 Day of the Week 30000 non-null object

10 Holiday Effect 30000 non-null bool

dtypes: bool(1), float64(1), int64(4), object(5)

memory usage: 2.3+ MB

None

**Statistics Summary of Data**

Product ID Units Sold Sales Revenue (USD) Discount Percentage \

count 3.000000e+04 30000.000000 30000.000000 30000.000000

mean 4.461294e+07 6.161967 2749.509593 2.973833

std 2.779759e+07 3.323929 2568.639288 5.974530

min 3.636541e+06 0.000000 0.000000 0.000000

25% 2.228600e+07 4.000000 882.592500 0.000000

50% 4.002449e+07 6.000000 1902.420000 0.000000

75% 6.559352e+07 8.000000 3863.920000 0.000000

max 9.628253e+07 56.000000 27165.880000 20.000000

Marketing Spend (USD)

count 30000.000000

mean 49.944033

std 64.401655

min 0.000000

25% 0.000000

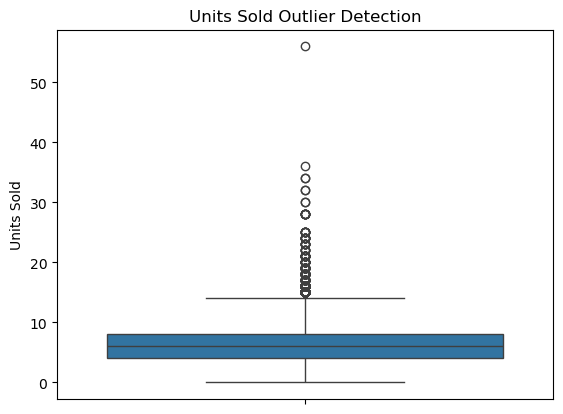
50% 1.000000

75% 100.000000

max 199.000000

|  |  |
| --- | --- |
| **Check for Duplication:**  Store ID 1  Product ID 42  Date 731  Units Sold 32  Sales Revenue (USD) 2545  Discount Percentage 5  Marketing Spend (USD) 200  Store Location 243  Product Category 4  Day of the Week 7  Holiday Effect 2  dtype: int64 | **Data Types:**  Store ID object  Product ID int64  Date datetime64[ns]  Units Sold int64  Sales Revenue (USD) float64  Discount Percentage int64  Marketing Spend (USD) int64  Store Location object  Product Category object  Day of the Week object  Holiday Effect bool  dtype: object |

Units Sold Outlier Detection (Boxplot)



A boxplot that visualizes potential outliers in the "Units Sold" column.

Boxplots show the distribution of data and identify outliers. Any points outside the whiskers of the boxplot are considered potential outliers. This plot is useful for identifying unusually large or small sales volumes that might need further investigation (e.g., bulk sales or data entry errors).

If many outliers are detected, this could suggest unusual events like promotional sales or anomalies in the data that need attention.

**Procedure followed is as below and is reflected in the code written**

Load the Data: First, load the CSV file and inspect its structure.

Check for Missing Values: Identify and handle any missing values in the dataset. You can either drop rows or columns with missing data, or fill missing values using imputation techniques

Check for Duplicates: Check if there are any duplicate rows and remove them.

Handle Data Types: Ensure all columns have the correct data types, especially date columns and numerical fields. Convert columns to the appropriate data types if necessary.

Encode Categorical Variables: If there are categorical columns such as 'Store Location' and 'Product Category', convert them to numerical form using one-hot encoding or label encoding.

Outlier Detection and Removal: Check for outliers in numerical data (such as 'Units Sold', 'Sales Revenue') that could skew analysis. This can be handled by capping, removing, or transforming the data.

Feature Engineering (Optional): Create new columns based on existing data that could be useful for analysis. For example, you might extract features from the 'Date' column (e.g., year, month, or day). (I included this in the code but did not use it)

Final Clean Dataset: Once the data is wrangled, I now had a clean and ready-to-analyze dataset. This was saved for use.

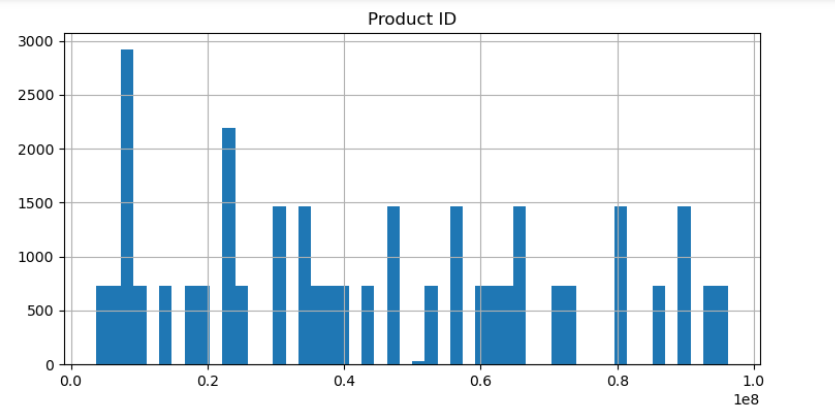
Question 3

1. Univariate Analysis

The graphs provide insights into among others units sold, sales revenue, discount percentage, marketing spend, sales by day of the week, top store location, and sales by product category, aiding decision-making in marketing, inventory, and resource allocation.

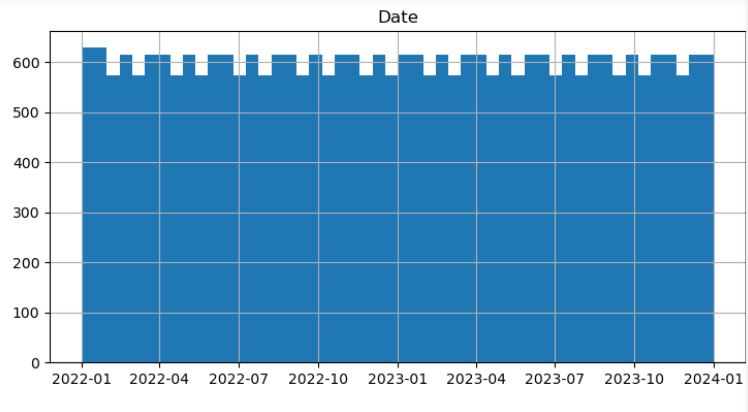
**Product ID**

Shows revenue distribution across product categories. Wider IQRs suggest variability in sales. Identifies high-revenue product categories for focused marketing.



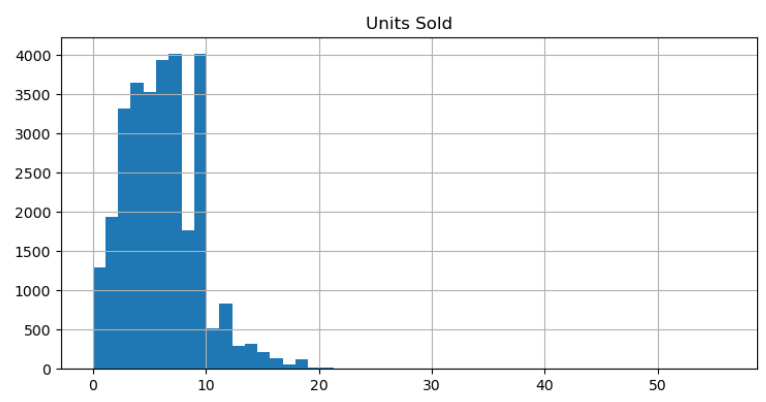
**Sales by Date**

Compares sales revenue across days. Higher medians on certain days suggest peak sales. This helps identifies high-revenue days for optimizing staffing or promotions.



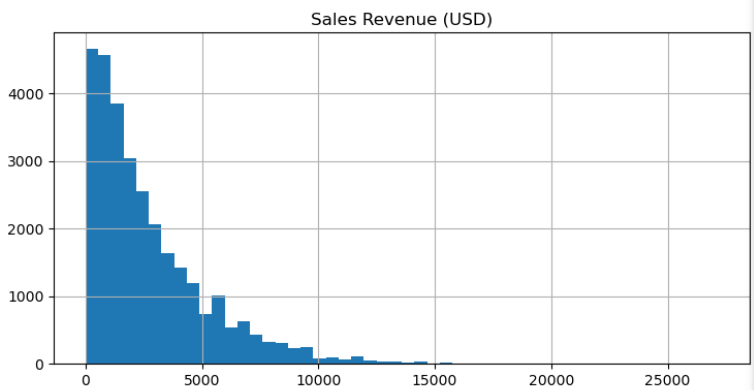
**Units sold**

Shows the frequency of various quantities sold. Right-skewed distribution suggests most sales involve smaller quantities. It also helps identify common sales volumes for inventory planning.



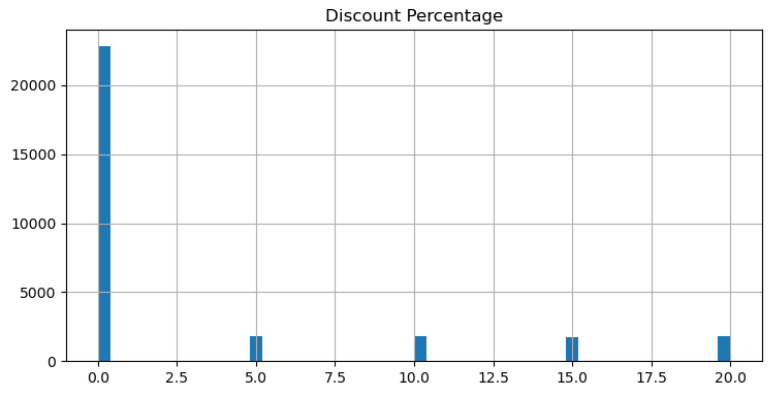
**Sales Revenue**

Visualizes revenue distribution across transactions. Outliers may indicate high-revenue transactions. This graph highlights the typical revenue range and rare high-value sales.



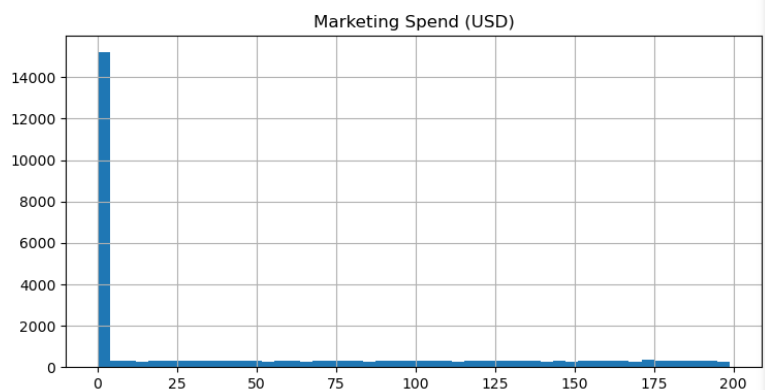
**Discount percentage**

Displays the distribution of applied discounts. Skewed data may show infrequent high discounts. This graph reveals how often discounts are used and their extent.



**Marketing Spend**

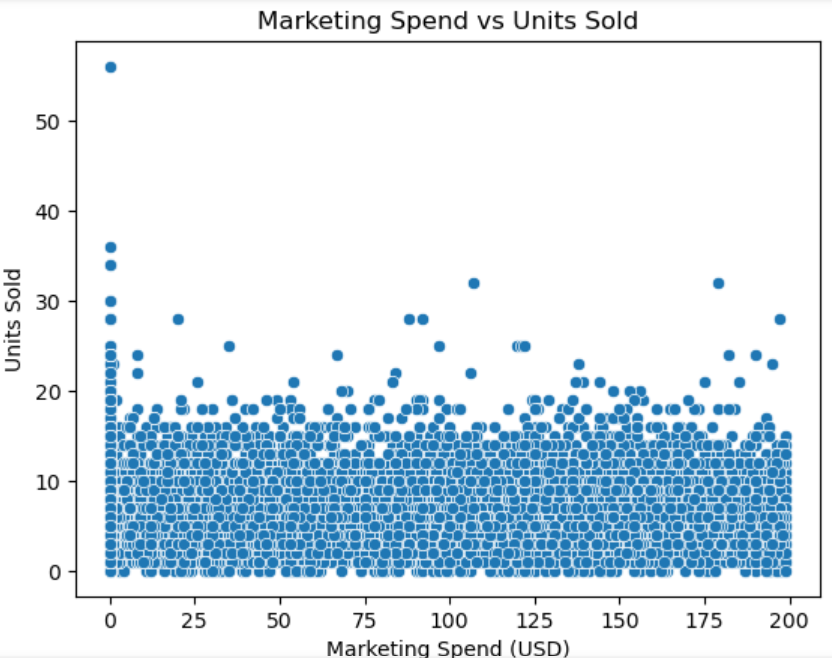
Shows the distribution of marketing expenditure. A right-skew indicates that high spending is rare. This helps assess typical marketing budgets for transactions.



1. Bivariate Analysis

Here, we shall investigate two variables for example;

1. Marketing Spend vs. Units Sold: Does increased marketing lead to more units sold?
2. Discount Percentage vs. Sales Revenue: What impact does discounting have on sales revenue?
3. Holiday Effect vs. Sales Revenue: Do holidays drive higher sales?
4. Marketing Spend vs. Sales Revenue: Does increased marketing lead to more units revenue?
5. Sales Revenue by Day of the Week: Which days of the week generate the highest revenue?
6. Top Sales Revenue by Store Location: Which locations bring what revenue threshold?
7. Sales Revenue by Product Category: What are the most sold products?
8. Correlation Matrix (Heatmap): How do different factors relate to each other
9. Heatmap of Sales Revenue by Day of the Week and Product Category

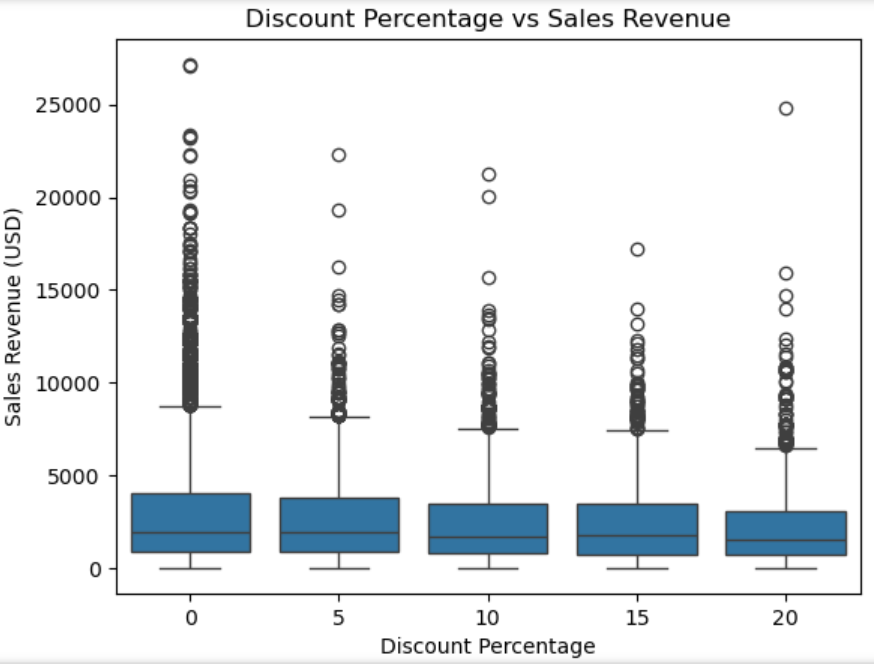
**Marketing Spend vs. Units Sold (Scatterplot)**

Above is a scatterplot showing the relationship between "Marketing Spend (USD)" and "Units Sold."

Scatterplots help visualize correlations between two variables. In this case, the graph seeks to determine if increased marketing expenditure leads to higher sales. Each point on the plot represents a sale, with "Marketing Spend" on the x-axis and "Units Sold" on the y-axis.

A positive correlation would indicate that higher marketing spend leads to higher units sold. If the data points are scattered without a clear pattern, it suggests that marketing spend alone does not directly influence the number of units sold.

**Impact of Discounts on Sales Revenue (Scatterplot)**

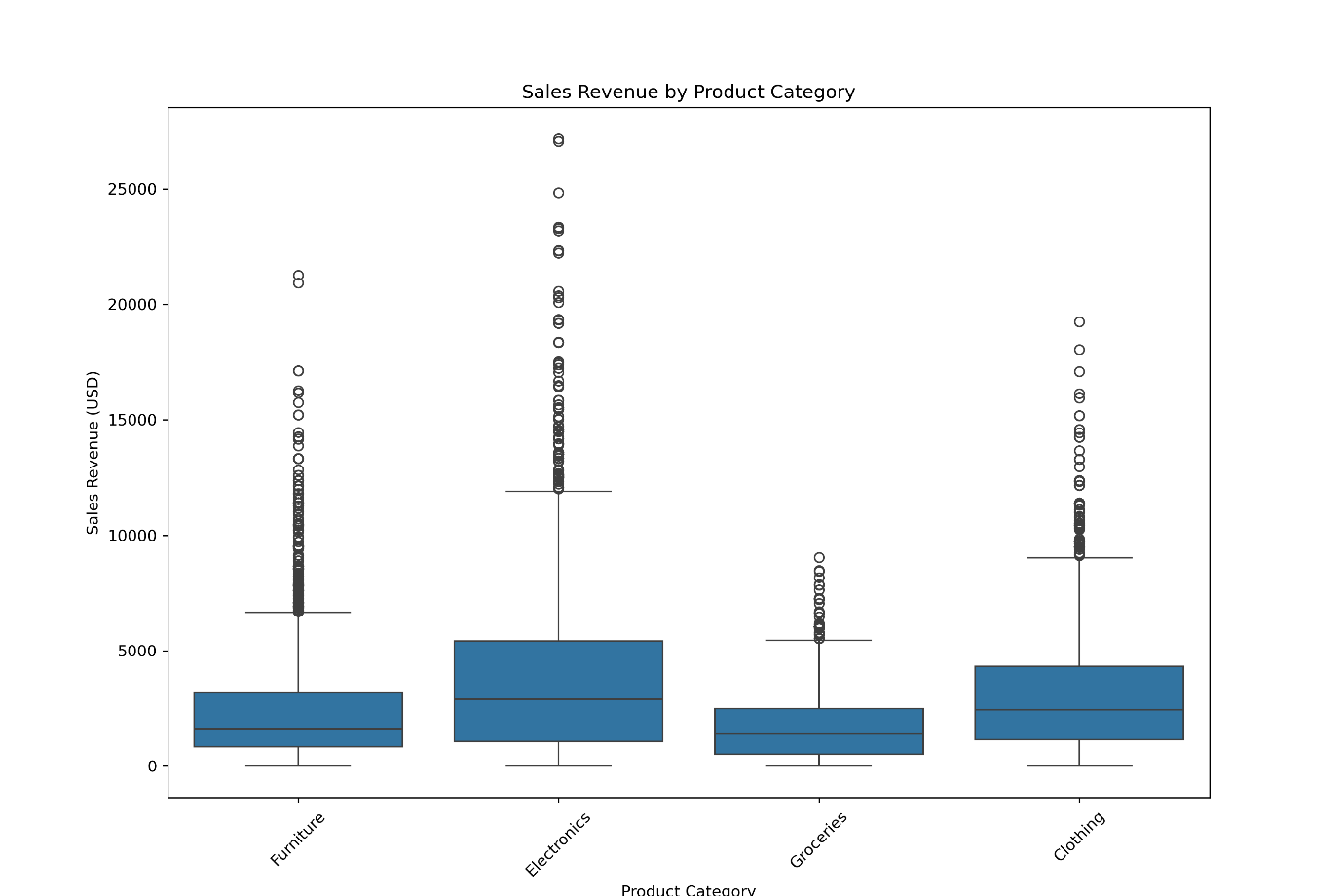


A scatterplot showing the relationship between "Discount Percentage" and "Sales Revenue (USD)."

This plot examines whether offering discounts increases sales revenue. Each point represents a sale, with discount percentage on the x-axis and revenue on the y-axis.

If higher discounts correspond to higher sales revenue, this suggests that discounts are effective in boosting sales. If there's no clear trend, it might indicate that discounts do not always lead to higher revenue.

**Sales Revenue by Product Category (Boxplot)**

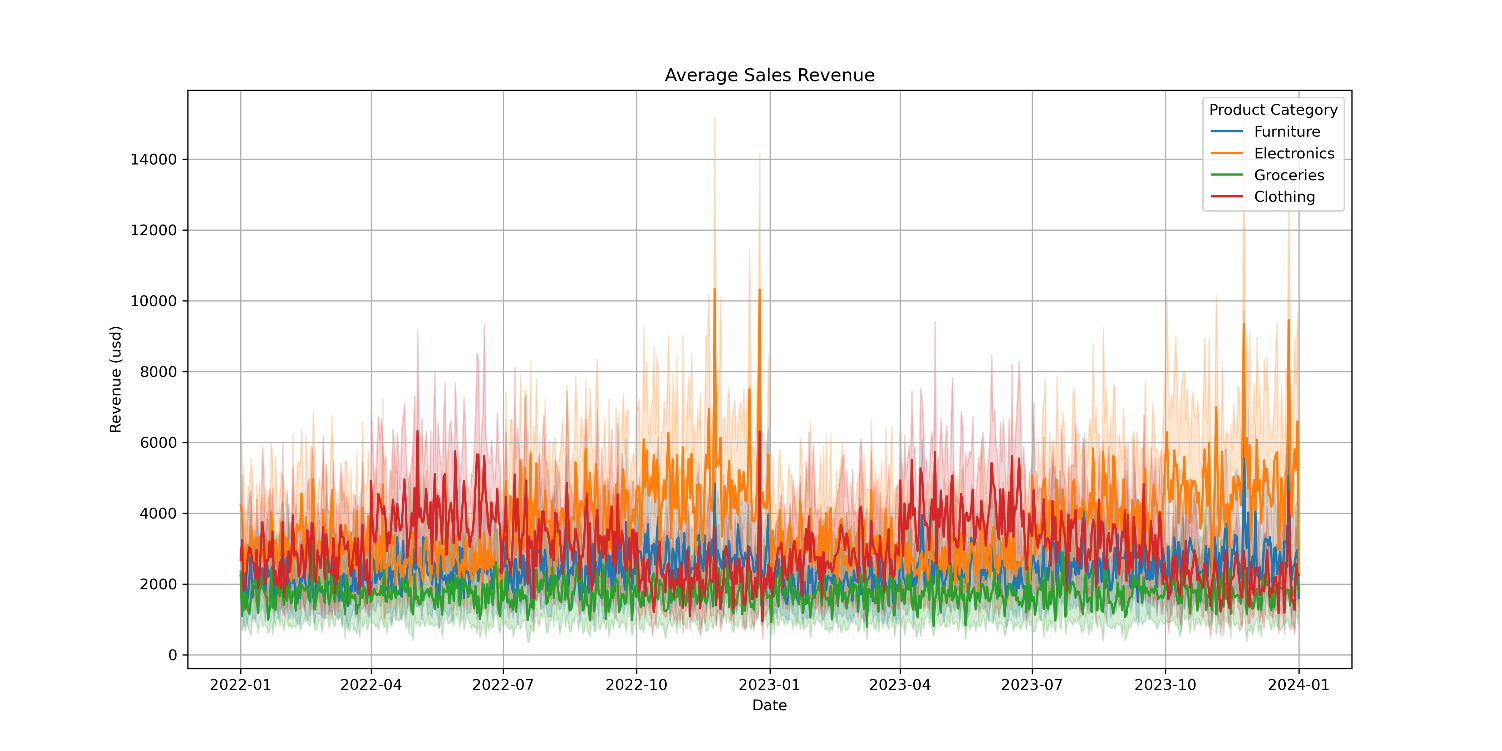


A boxplot comparing the distribution of sales revenue across different product categories.

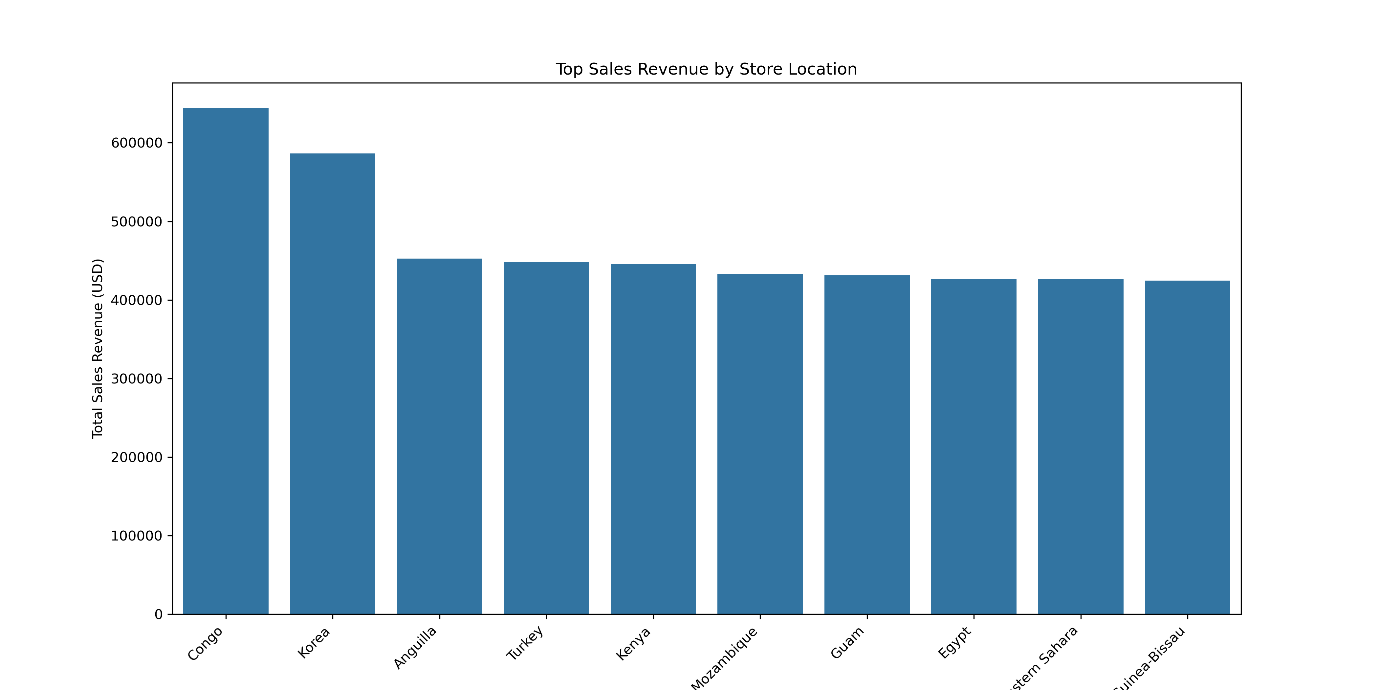
The boxplot helps to compare how various product categories perform in terms of revenue. A wider box or higher median suggests that a particular product category contributes more to total sales revenue.

Identifying which product categories generate the most revenue can guide inventory decisions and promotional strategies.

Below is a lineplot that shows the average sales revenue at different dates for the different product categories



**Top Sales Revenue by Store Location (Barplot)**

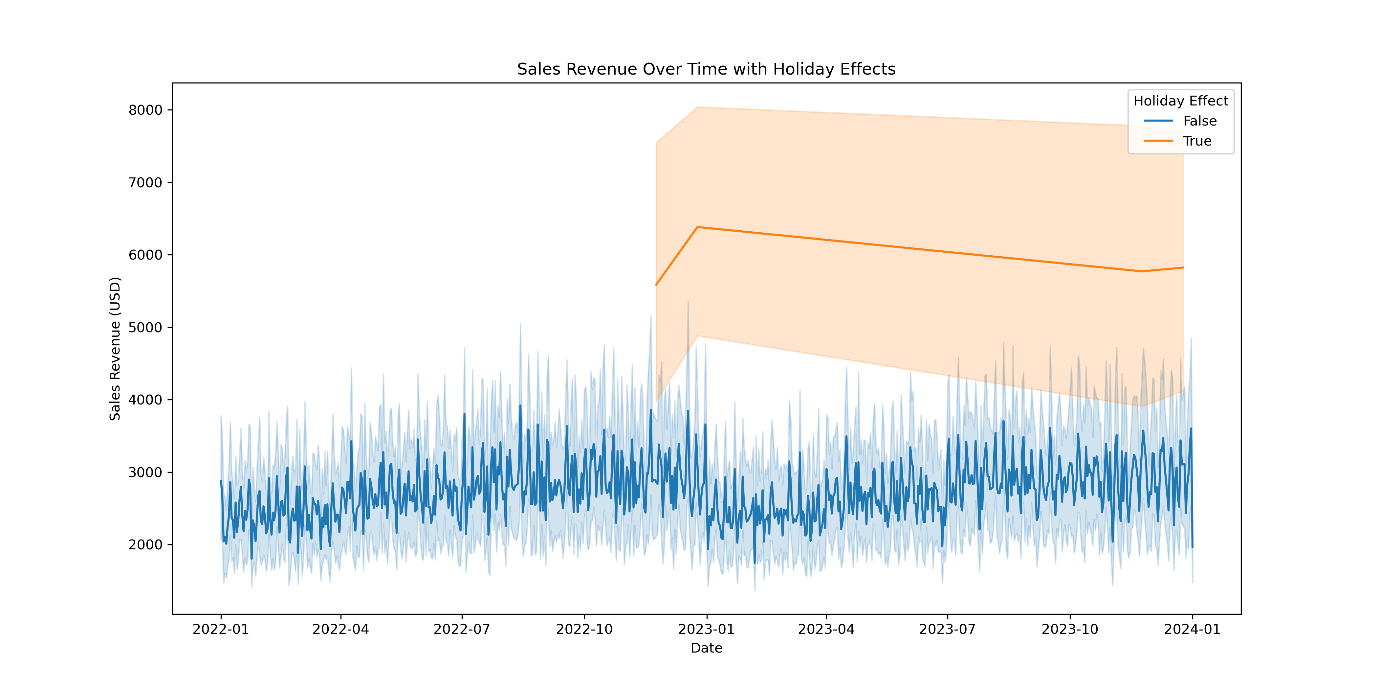


A bar plot showing the top 10 store locations by total sales revenue.

This graph ranks store locations based on the revenue they generate. The longer the bar, the higher the revenue for that store location.

This plot provides insights into the top-performing store locations, which could be crucial for business decisions, such as allocating resources, staffing, or inventory management.

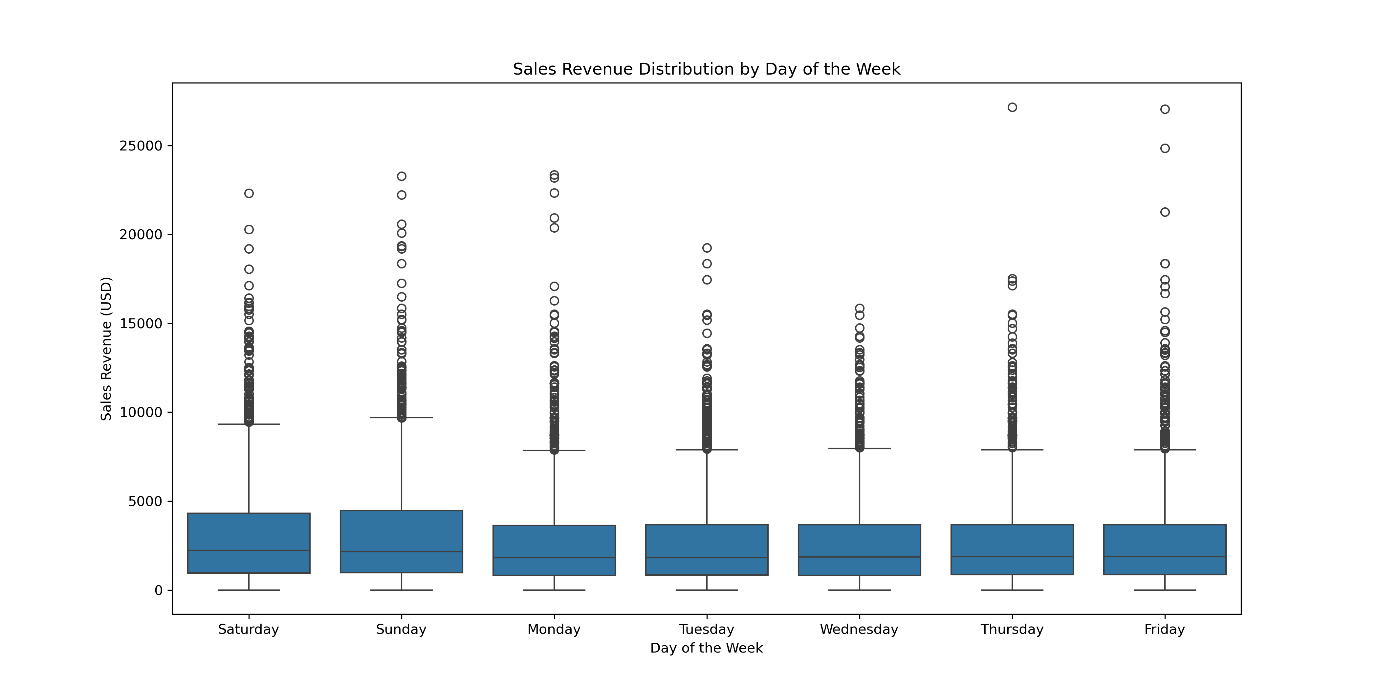
**Sales Revenue Over Time with Holiday Effects (Line Plot)**

Above is a line plot showing the trend of "Sales Revenue (USD)" over time, with "Holiday Effect" highlighted.

This plot provides insights into how sales revenue fluctuates over time, particularly during holiday periods. Holidays are often associated with sales spikes due to promotional events or increased customer spending.

A visible spike in sales during holiday periods would suggest that holidays drive higher revenue. This insight could be useful for planning future promotions or inventory stocking.

**Sales Revenue by Day of the Week (Boxplot)**

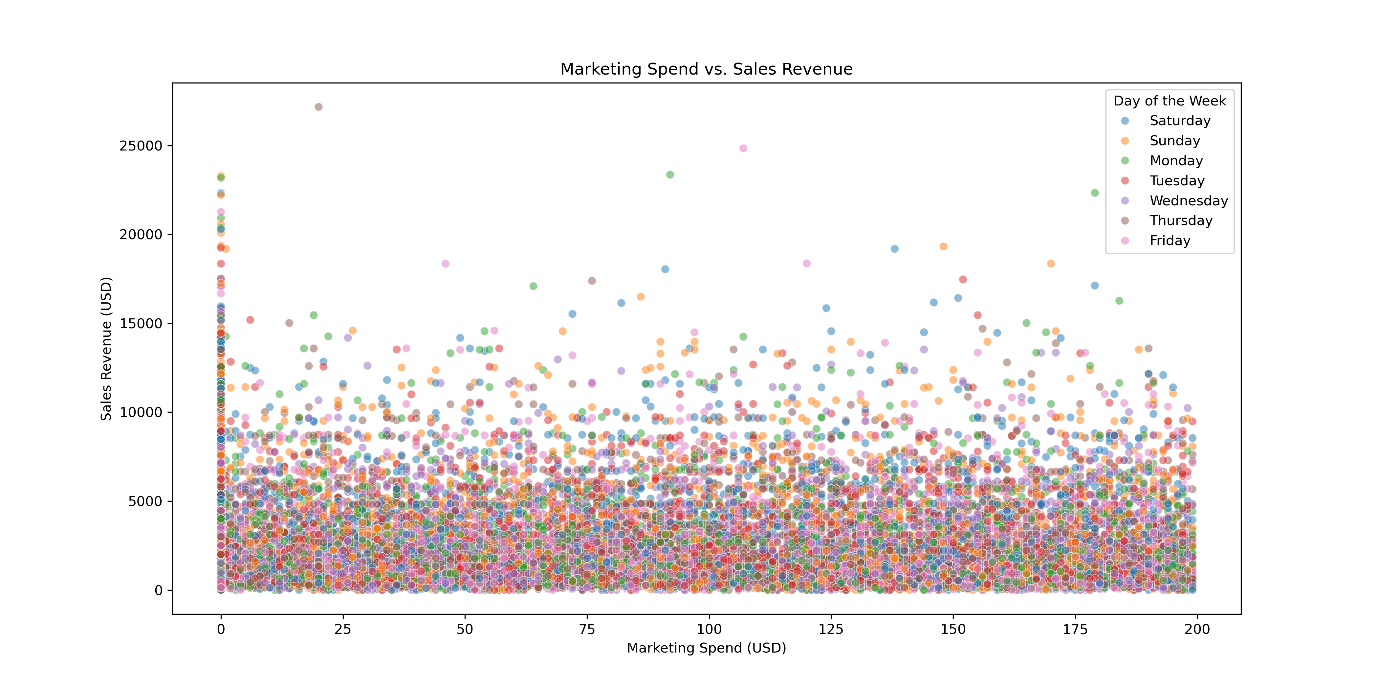


Above is a boxplot showing the distribution of sales revenue across different days of the week.

This boxplot will help visualize how sales revenue varies by day. The median (center line) and spread (whiskers) give insights into which days generate higher or lower revenue.

If weekends or certain weekdays have significantly higher revenue, it could suggest customer behavior trends, such as increased shopping on weekends or during certain promotions.

**Marketing Spend vs. Sales Revenue (Scatterplot)**



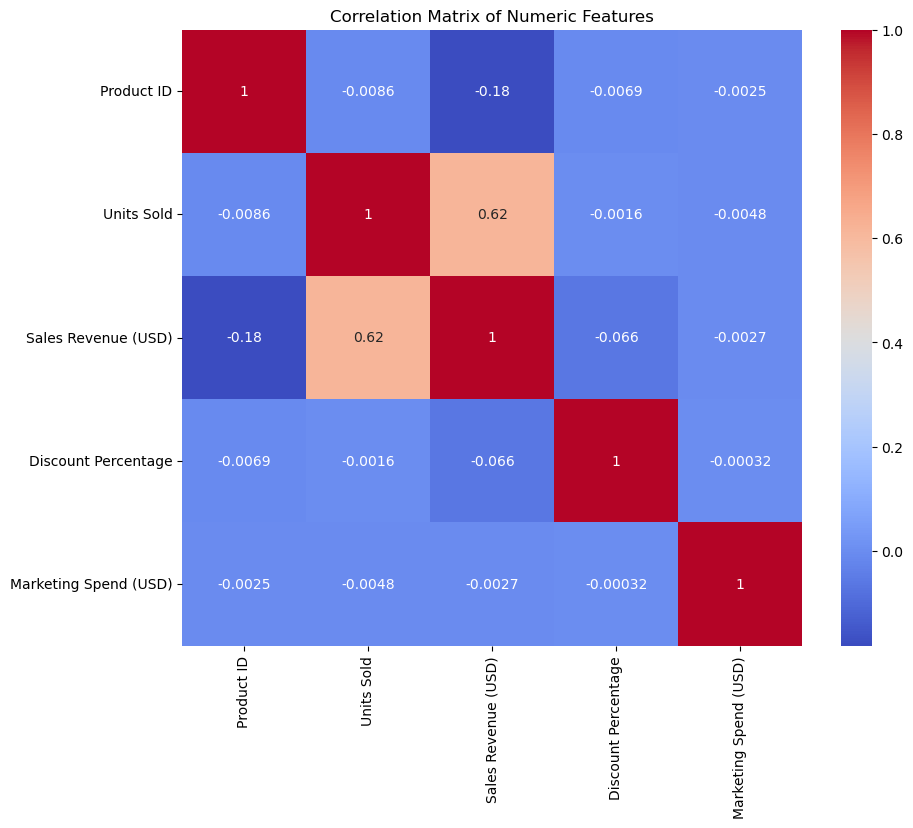
Above is a scatterplot of "Marketing Spend (USD)" vs. "Sales Revenue (USD)" colored by the "Day of the Week."

This plot aims to understand whether increasing marketing spend results in higher sales revenue. The color coding by "Day of the Week" adds an extra layer of analysis, showing how sales differ across different days.

If you observe clusters of high revenue on specific days, this could imply that certain days (e.g., weekends) are more lucrative for sales, perhaps due to higher customer traffic or specific promotions.

C. Multivariate Analysis

Correlation Matrix (Heatmap)

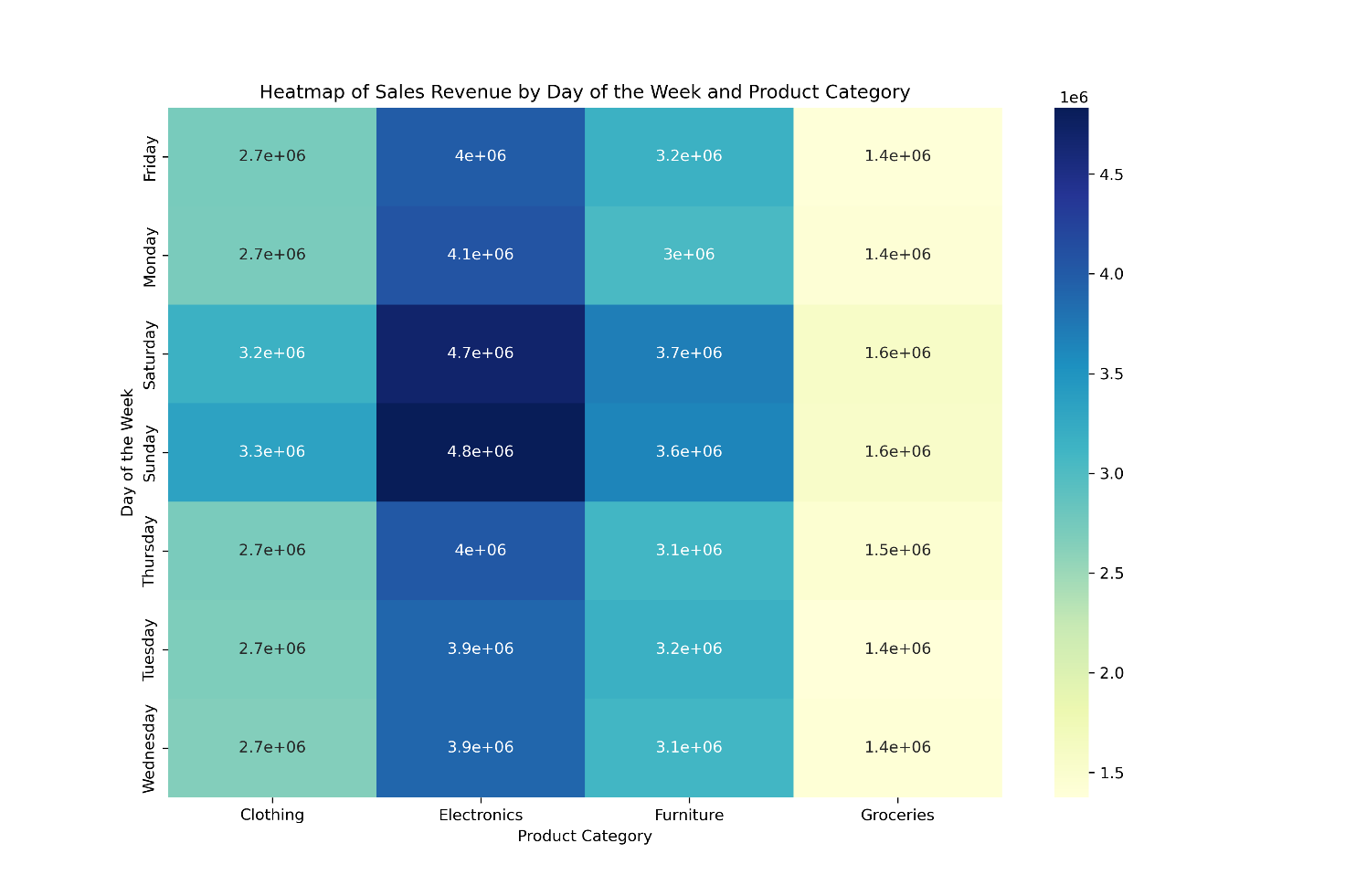


Above is a heatmap showing the correlation between different numerical variables (e.g., marketing spend, sales revenue, units sold).

The correlation matrix quantifies how strongly different numerical variables are related to one another. A value close to +1 or -1 indicates a strong positive or negative correlation, respectively.

Strong correlations between variables like "Units Sold" and "Sales Revenue" confirm expected relationships. Weak correlations or negative correlations can indicate areas where other factors (like marketing or discounting) may play a role.

Heatmap of Sales Revenue by Day of the Week and Product Category



The heatmap above shows the total sales revenue for each combination of "Day of the Week" and "Product Category."

This graph helps identify patterns between product categories and days of the week. Darker areas in the heatmap indicate higher revenue, while lighter areas suggest lower sales activity.

Certain product categories may sell better on specific days of the week, providing insights for scheduling promotions or adjusting inventory.

Conclusion

From the detailed visual analysis, the following conclusions have been drawn;

1. Sales Performance by Store Location: Certain stores consistently outperform others in terms of sales revenue. Management could focus on expanding operations or optimizing performance at these high-revenue locations.
2. Impact of Discounts: Discounts appear to have a mixed effect on sales revenue. While they may boost sales in some cases, there is no consistent pattern across all products. Careful experimentation with discount strategies is necessary to avoid cutting into profit margins.
3. Seasonal and Holiday Trends: Sales revenue significantly spikes during holiday periods, suggesting that customers respond well to holiday promotions. This insight can be used to time promotional campaigns more effectively.
4. Product Category Insights: Certain product categories generate more revenue than others. These insights can help in deciding which products to promote more aggressively or stock in larger quantities.
5. Marketing Spend: The relationship between marketing spend and sales is not always straightforward. While there is some correlation between higher marketing spend and increased sales, it’s not always guaranteed. A more targeted approach to marketing might yield better results.
6. Sales Behavior Across Days of the Week: The analysis shows a noticeable variation in sales revenue across different days of the week. Sales tend to peak on certain days, which could inform store staffing, inventory management, and promotional planning.

**The Code**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

#######################################################

# Data Wrangling

# Step 1: Load the dataset

df = pd.read\_csv('Retail\_sales.csv')

#Display info

print("\nInformation About Data:")

print(df.info())

print("\nFirst 5 observations of the dataset:")

print(df.head())

print("\nLast 5 observations of the dataset:")

print(df.tail())

print("\nStatistics Summary of Data:")

print(df.describe())

print("\nCheck for Duplication:")

print(df.nunique())

# Step 2: Check for missing values

print("\nNumber of missing records in each column")

print(df.isnull().sum())

# Drop rows with missing values

df = df.dropna()

# Step 3: Check for duplicate rows

df = df.drop\_duplicates()

# Step 4: Convert 'Date' to datetime format

df['Date'] = pd.to\_datetime(df['Date'])

# Step 5: One-hot encode categorical variables

#df\_encoded = pd.get\_dummies(df, columns=['Store Location', 'Product Category', 'Day of the Week'])

# Check data types

print("\nData Types:")

print(df.dtypes)

# Select columns with object (categorical) data types

categorical\_cols = df.select\_dtypes(include=['object']).columns

# Apply one-hot encoding to all categorical columns

df\_encoded = pd.get\_dummies(df, columns=categorical\_cols)

# View the first few rows of the encoded DataFrame

#print(df\_encoded.head())

# Step 6: Outlier detection

sns.boxplot(df['Units Sold'])

plt.title('Units Sold Outlier Detection')

plt.show()

# Step 7: Extract new features from 'Date'

df['Year'] = df['Date'].dt.year

df['Month'] = df['Date'].dt.month

df['Day'] = df['Date'].dt.day

# Step 8: Save the cleaned data

df.to\_csv('Retail\_sales\_cleaned.csv', index=False)

#####################################################

# EDA: Univariate Analysis

df\_encoded.hist(bins=50, figsize=(20, 15))

plt.savefig('all.png', dpi=300)

plt.show()

#################################################

# Bivariate Analysis

sns.scatterplot(x='Marketing Spend (USD)', y='Units Sold', data=df\_encoded)

plt.title('Marketing Spend vs Units Sold')

plt.savefig('spendVSunits.png', dpi=300)

plt.savefig('spendbyunitssold.png', dpi=300)

plt.show()

# Marketing Spend vs. Sales Revenue

plt.figure(figsize=(14, 7))

sns.scatterplot(data=df, x='Marketing Spend (USD)', y='Sales Revenue (USD)', alpha=0.5, hue='Day of the Week')

plt.title('Marketing Spend vs. Sales Revenue')

plt.xlabel('Marketing Spend (USD)')

plt.ylabel('Sales Revenue (USD)')

plt.savefig('revenuebyspend.png', dpi=300)

plt.show()

# Save the figure as a PNG file

plt.figure(figsize=(14, 7))

sns.boxplot(data=df, x='Day of the Week', y='Sales Revenue (USD)')

plt.title('Sales Revenue Distribution by Day of the Week')

plt.xlabel('Day of the Week')

plt.ylabel('Sales Revenue (USD)')

#plt.xticks(rotation=45)

plt.savefig('revenuebyweekday.png', dpi=300)

plt.show()

# Aggregate sales revenue by store location

store\_revenue = df.groupby('Store Location')['Sales Revenue (USD)'].sum()

# Sort the values and select the top N store locations

top\_n = 10 # Number of top store locations to display

top\_store\_revenue = store\_revenue.sort\_values(ascending=False).head(top\_n)

# Create the bar plot

plt.figure(figsize=(14, 7))

sns.barplot(x=top\_store\_revenue.index, y=top\_store\_revenue.values)

plt.title('Top Sales Revenue by Store Location')

plt.xlabel('Store Location')

plt.ylabel('Total Sales Revenue (USD)')

plt.xticks(rotation=45, ha='right') # Adjust rotation and alignment

plt.savefig('revenuebylocation.png', dpi=300)

plt.show()

#Sales Revenue by Product Category

plt.figure(figsize=(12, 8))

sns.boxplot(x='Product Category', y='Sales Revenue (USD)', data=df)

plt.title('Sales Revenue by Product Category')

plt.xlabel('Product Category')

plt.ylabel('Sales Revenue (USD)')

plt.xticks(rotation=45)

plt.savefig('revenuebyproduct.png', dpi=300)

plt.show()

plt.figure(figsize=(30,15))

sns.lineplot(data=df,x='Date',y='Sales Revenue (USD)', hue = 'Product Category',estimator='mean',palette='tab10')

plt.title('Average Sales Revenue')

plt.xlabel('Date')

plt.ylabel('Revenue (usd)')

plt.grid(True)

plt.savefig('revenueproductdate.png', dpi=300)

plt.show()

#Impact of Discounts on Sales Revenue

plt.figure(figsize=(10, 6))

#sns.boxplot(x='Discount Percentage', y='Sales Revenue (USD)', data=df\_encoded)

sns.scatterplot(x='Discount Percentage', y='Sales Revenue (USD)', data=df\_encoded)

plt.title('Impact of Discounts on Sales Revenue')

plt.xlabel('Discount Percentage')

plt.ylabel('Sales Revenue (USD)')

plt.savefig('revenuediscount.png', dpi=300)

plt.show()

#Sales revenue over time with holiday effect

plt.figure(figsize=(14, 7))

sns.lineplot(x='Date', y='Sales Revenue (USD)', hue='Holiday Effect', data=df)

plt.title('Sales Revenue Over Time with Holiday Effects')

plt.xlabel('Date')

plt.ylabel('Sales Revenue (USD)')

plt.legend(title='Holiday Effect')

plt.savefig('revenueholiday.png', dpi=300)

plt.show()

##################################################################

#Multivariate Analysis

# Select only numeric columns

numeric\_df = df\_encoded.select\_dtypes(include=['number'])

# Calculate correlation matrix on numeric columns

corr\_matrix = numeric\_df.corr()

# Plot the correlation matrix

plt.figure(figsize=(10, 8))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix of Numeric Features')

plt.savefig('corr\_matrix.png', dpi=300)

plt.show()

#Heatmap of Sales Revenue by Day of the Week and Product Category

plt.figure(figsize=(12, 8))

heatmap\_data = df.pivot\_table(index='Day of the Week', columns='Product Category', values='Sales Revenue (USD)', aggfunc='sum')

sns.heatmap(heatmap\_data, annot=True, cmap='YlGnBu')

plt.title('Heatmap of Sales Revenue by Day of the Week and Product Category')

plt.savefig('salesweekproduct.png', dpi=300)

plt.show()