**Sales Data Analysis of a ShopSavvy**

# A. Introduction

## 1. Background

ShopSavvy is a prominent departmental store known for offering a wide variety of products, ranging from groceries to electronics, under one roof. The store has become a household name due to its competitive pricing, quality products, and customer-centric approach. The growing customer base and expanding product range have led to the accumulation of vast amounts of sales data, which can provide valuable insights for decision-making.

## 2. Statement of Problem

Despite its success, ShopSavvy faces challenges in understanding which products drive the most revenue and how customer buying behavior changes over time. With numerous products and fluctuating sales patterns, it is essential to analyze the sales data to identify trends, optimize inventory, and improve customer satisfaction.

## 3. Purpose of the Analysis

The purpose of this sales analysis is as follows:

I. To Analyze overall sales trends

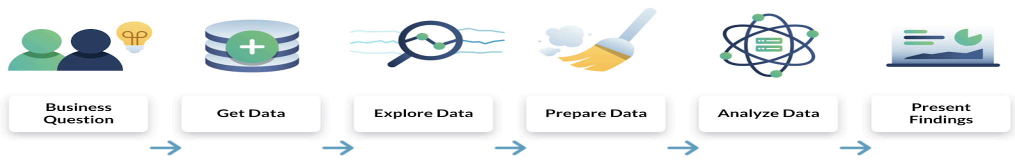
II. To Identify top-selling products.

III. To Provide insights to optimize operations.

IV. To provide actionable insights that can help ShopSavvy optimize its operations, increase revenue, and make informed business decisions.

## B. Methodology

The methodology outlines the steps taken to perform the sales data analysis, from problem definition to data visualization and presentation.




### 1. Problem Definition

The primary objective is to identify sales trends, top-selling products, and customer preferences in ShopSavvy.
The analysis aims to answer the following questions:

I. What are the top-selling products at ShopSavvy?
II. Which time periods have the highest sales?
III. Who are the top customers based on purchase history?

### 2. Data Collection

Given that we did not have access to actual sales data, I generated a sample of 200 dataset using Python's random number generation functions.
This dataset includes key fields such as Date,Customer_Type,Product_Id,Category,Product_price,Quantity_sold,Customer_Name,Discount,Net_profit,Transaction_Id,Brand_Name.

```python
import pandas as pd
import numpy as np
import faker

# Initialize Faker for generating fake names
fake = faker.Faker()

# Generate sample data
np.random.seed(42)
dates = pd.date_range(start="2023-01-01", end="2024-03-31", freq="D")
categories = ["cosmetics", "household", "grocery", "fastfoods", "clothes", "utensils", "gardening"]
brands = ["Brand_X", "Brand_Y", "Brand_Z"]
customer_types = ["New", "Regular"]

# Generate a list of unique fake names
unique_names = [fake.name() for _ in range(50)]  # Generate 50 unique names

data = {
    "Date": np.random.choice(dates, size=200),
    "Customer_Type": np.random.choice(customer_types, size=200),
    "Product_ID": np.random.randint(1000, 9999, size=200),  # Product ID between 1000 and 9999
    "Category": np.random.choice(categories, size=200),
    "Product_Price": np.random.uniform(10, 1000, size=200),  # Product price between 10 and 1000
    "Quantity_Sold": np.random.randint(1, 10, size=200),
    "Customer_Name": np.random.choice(unique_names, size=200),  # Use the list of unique names
    "Discount": np.random.uniform(0, 0.2, size=200),  # Discount as a fraction (0 to 20%)
    "Net_Profit": np.random.uniform(5, 500, size=200),  # Net profit between 5 and 500
    "Transaction_ID": np.random.randint(100000, 999999, size=200),  # Transaction ID between 100000 and 999999
    "Brand_Name": np.random.choice(brands, size=200)
}

# Create DataFrame
df = pd.DataFrame(data)

# Save DataFrame to CSV file
df.to_csv('generated_data.csv', index=False)

# Display the first few rows of the DataFrame in Jupyter Notebook
df.head(20)
```

| | Date | Customer_Type | Product_ID | Category | Product_Price | Quantity_Sold | Customer_Name | Discount | Net_Profit | Transaction_ID | Brand_Name |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2023-04-13 | Regular | 5911 | cosmetics | 286.965872 | 6 | Danielle Johnson | 0.027364 | 484.594538 | 869299 | Brand_Z |
| 1 | 2024-03-11 | Regular | 4987 | grocery | 417.094654 | 2 | Mary Phillips | 0.190047 | 376.077657 | 965962 | Brand_X |
| 2 | 2023-12-15 | Regular | 7015 | fastfoods | 606.754063 | 8 | Jay Fisher | 0.089201 | 69.392689 | 516715 | Brand_Z |
| 3 | 2023-09-28 | Regular | 2218 | household | 278.248066 | 5 | Sandra Young | 0.037027 | 380.340282 | 471000 | Brand_Z |
| 4 | 2023-04-17 | Regular | 5496 | cosmetics | 141.853274 | 7 | Debra Robinson | 0.108380 | 17.170524 | 313567 | Brand_Z |
| 5 | 2023-03-13 | Regular | 5735 | gardening | 85.438376 | 2 | Debra Robinson | 0.174589 | 15.951158 | 614933 | Brand_X |
| 6 | 2023-07-08 | New | 5555 | gardening | 941.126509 | 8 | Matthew Martinez | 0.146445 | 165.187058 | 695923 | Brand_X |
| 7 | 2023-01-21 | Regular | 9050 | utensils | 422.472667 | 2 | Thomas Malone | 0.161312 | 246.878379 | 994235 | Brand_Y |
| 8 | 2023-04-13 | Regular | 4446 | clothes | 585.350995 | 4 | Justin Robinson | 0.131757 | 386.351672 | 316184 | Brand_Z |
| 9 | 2023-05-02 | New | 2045 | fastfoods | 919.984786 | 1 | Robin Wade | 0.138455 | 343.231211 | 398090 | Brand_Y |
| 10 | 2023-08-03 | Regular | 7893 | cosmetics | 91.920936 | 5 | April Lawson | 0.169839 | 225.721840 | 417348 | Brand_Y |
| 11 | 2023-11-27 | New | 2693 | gardening | 877.894871 | 9 | Christina Brewer | 0.049934 | 140.445200 | 282958 | Brand_X |
| 12 | 2023-03-29 | New | 4436 | clothes | 556.071994 | 1 | Sandra Young | 0.097885 | 498.576628 | 377707 | Brand_X |
| 13 | 2024-01-08 | Regular | 9754 | gardening | 173.185919 | 9 | Brett Anderson | 0.044242 | 215.959745 | 192324 | Brand_Y |
| 14 | 2023-04-10 | New | 6895 | cosmetics | 417.142565 | 8 | David Martin | 0.197534 | 228.436577 | 323536 | Brand_Z |
| 15 | 2023-12-26 | New | 4354 | grocery | 779.826261 | 6 | Danielle Johnson | 0.188812 | 85.993791 | 857232 | Brand_X |
| 16 | 2023-06-01 | New | 1225 | cosmetics | 485.566381 | 7 | Beverly Evans | 0.007885 | 398.430727 | 371991 | Brand_Z |
| 17 | 2023-05-11 | New | 5893 | utensils | 985.433190 | 3 | Christina Moore | 0.141115 | 348.372702 | 126155 | Brand_Z |
| 18 | 2023-05-30 | Regular | 8022 | cosmetics | 382.971580 | 1 | Jay Fisher | 0.185050 | 114.280958 | 441815 | Brand_Y |

# 3. Data Cleaning

To ensure data quality, I performed the following data cleaning steps:

I. **Remove duplicate or irrelevant observations**: Any duplicate entries were removed.

II. **Handle missing data**: The dataset was checked for missing values, and any missing data was handled appropriately.

III. **Ensuring format consistency**: Data formats, numeric values, and text fields were standardized for consistency.

```python
import pandas as pd

# Load your dataset
data = pd.read_csv('/Users/lanisha/Desktop/ML/BrainwaveTask1/generated_data.csv')

# Step 1: Remove rows with null values
cleaned_data = data.dropna()

# Step 2: Remove duplicate rows
cleaned_data = cleaned_data.drop_duplicates()

# Step 3: Drop non-useful columns (including 'transaction ID' and 'brand name')
columns_to_remove = ['Transaction_ID', 'Brand_Name']
cleaned_data = cleaned_data.drop(columns=columns_to_remove)

# Optional Step 4: Convert columns to appropriate data types
cleaned_data['Product_Price'] = cleaned_data['Product_Price'].astype(float)
cleaned_data['Quantity_Sold'] = cleaned_data['Quantity_Sold'].astype(int)

# Optional Step 5: Verify the cleaned data
print(cleaned_data.info())  # Check data types and non-null counts
print(cleaned_data.head())  # View the first few rows of the cleaned data

# Save the cleaned data to a new CSV file (optional)
cleaned_data.to_csv('cleaned_dataset.csv', index=False)

# Display the first few rows of the DataFrame in Jupyter Notebook
df.head(20)
```

[23]:

| | Date | Customer_Type | Product_ID | Category | Product_Price | Quantity_Sold | Customer_Name | Discount | Net_Profit | Transaction_ID | Brand_Name |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2023-04-13 | Regular | 5911 | cosmetics | 286.965872 | 6 | Danielle Johnson | 0.027364 | 484.594538 | 869299 | Brand_Z |
| 1 | 2024-03-11 | Regular | 4987 | grocery | 417.094654 | 2 | Mary Phillips | 0.190047 | 376.077657 | 965962 | Brand_X |
| 2 | 2023-12-15 | Regular | 7015 | fastfoods | 606.754063 | 8 | Jay Fisher | 0.089201 | 69.392689 | 516715 | Brand_Z |
| 3 | 2023-09-28 | Regular | 2218 | household | 278.248066 | 5 | Sandra Young | 0.037027 | 380.340282 | 471000 | Brand_Z |
| 4 | 2023-04-17 | Regular | 5496 | cosmetics | 141.853274 | 7 | Debra Robinson | 0.108380 | 17.170524 | 313567 | Brand_Z |
| 5 | 2023-03-13 | Regular | 5735 | gardening | 85.438376 | 2 | Debra Robinson | 0.174589 | 15.951158 | 614933 | Brand_X |
| 6 | 2023-07-08 | New | 5555 | gardening | 941.126509 | 8 | Matthew Martinez | 0.146445 | 165.187058 | 695923 | Brand_X |
| 7 | 2023-01-21 | Regular | 9050 | utensils | 422.472667 | 2 | Thomas Malone | 0.161312 | 246.878379 | 994235 | Brand_Y |
| 8 | 2023-04-13 | Regular | 4446 | clothes | 585.350995 | 4 | Justin Robinson | 0.131757 | 386.351672 | 316184 | Brand_Z |
| 9 | 2023-05-02 | New | 2045 | fastfoods | 919.984786 | 1 | Robin Wade | 0.138455 | 343.231211 | 398090 | Brand_Y |
| 10 | 2023-08-03 | Regular | 7893 | cosmetics | 91.920936 | 5 | April Lawson | 0.169839 | 225.721840 | 417348 | Brand_Y |
| 11 | 2023-11-27 | New | 2693 | gardening | 877.894871 | 9 | Christina Brewer | 0.049934 | 140.445200 | 282958 | Brand_X |
| 12 | 2023-03-29 | New | 4436 | clothes | 556.071994 | 1 | Sandra Young | 0.097885 | 498.576628 | 377707 | Brand_X |
| 13 | 2024-01-08 | Regular | 9754 | gardening | 173.185919 | 9 | Brett Anderson | 0.044242 | 215.959745 | 192324 | Brand_Y |
| 14 | 2023-04-10 | New | 6895 | cosmetics | 417.142565 | 8 | David Martin | 0.197534 | 228.436577 | 323536 | Brand_Z |
| 15 | 2023-12-26 | New | 4354 | grocery | 779.826261 | 6 | Danielle Johnson | 0.188812 | 85.993791 | 857232 | Brand_X |
| 16 | 2023-06-01 | New | 1225 | cosmetics | 485.566381 | 7 | Beverly Evans | 0.007885 | 398.430727 | 371991 | Brand_Z |
| 17 | 2023-05-11 | New | 5893 | utensils | 985.433190 | 3 | Christina Moore | 0.141115 | 348.372702 | 126155 | Brand_Z |
| 18 | 2023-05-30 | Regular | 8022 | cosmetics | 382.971580 | 1 | Jay Fisher | 0.185050 | 114.280958 | 441815 | Brand_Y |
| 19 | 2023-11-05 | New | 6600 | cosmetics | 752.082517 | 5 | Bryan Davis | 0.036115 | 45.778618 | 741179 | Brand_Y |

## 4. Data Analysis

I have performed several analyses to extract meaningful insights:

I. **Summary Statistics**: Basic statistics such as total sales and average sales were calculated.

```python
# Load the cleaned dataset
data = pd.read_csv('cleaned_dataset.csv')

# Calculate total sales
data['Total_Sales'] = data['Product_Price'] * data['Quantity_Sold']
Total_Sales = data['Total_Sales'].sum()

# Calculate average sales
Average_Sales = data['Total_Sales'].mean()

# Print summary statistics
print(f"Total Sales is: ${Total_Sales:,.2f}")
print(f"Average Sales is: ${Average_Sales:,.2f}")
```

```
Total Sales is: $508,959.31
Average Sales is: $2,544.80
```

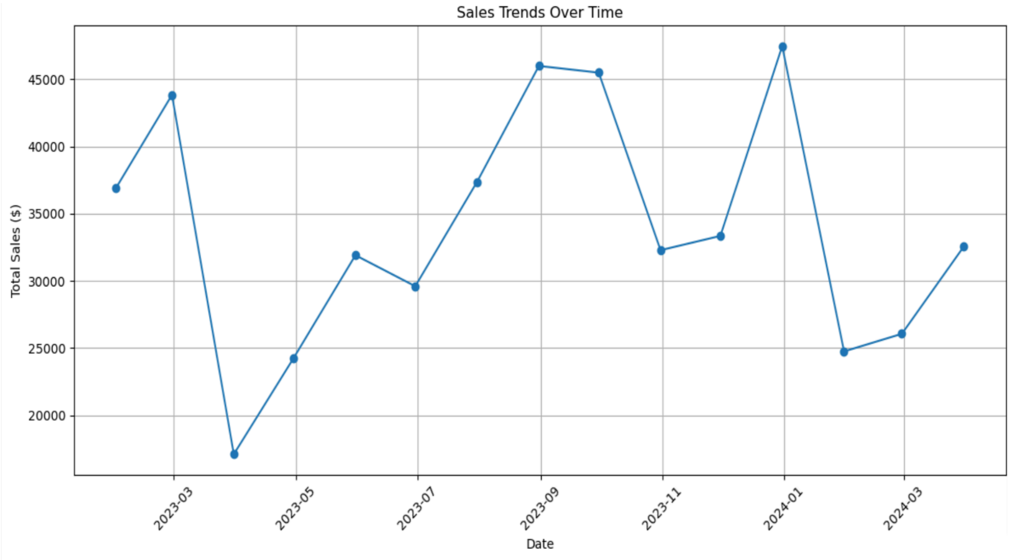
II. **Top-Selling Products**: Products were ranked based on their total sales amount.

```python
[37]: # Aggregate sales by product
Top_Selling_Products = data.groupby('Category').agg({'Total_Sales': 'sum'}).reset_index()

# Sort products by total sales in descending order
Top_Selling_Products = Top_Selling_Products.sort_values(by='Total_Sales', ascending=False)

# Print the top-selling products
print("Top Five Selling Products are:")
print(Top_Selling_Products.head())  # Display top 5 products
```

```
Top Five Selling Products are:
    Category    Total_Sales
2  fastfoods  109882.370693
1  cosmetics   97850.219186
0    clothes   78755.541978
3  gardening   78371.697981
4    grocery   50997.837872
```

III. **Sales Trends**: Sales data was plotted over time to identify any trends or patterns.

```python
import matplotlib.pyplot as plt

# Convert 'date' column to datetime//this is important to do
data['Date'] = pd.to_datetime(data['Date'])

# Aggregate sales by month
Monthly_Sales = data.resample('M', on='Date').agg({'Total_Sales': 'sum'})

# Plot sales trends
plt.figure(figsize=(12, 6))
plt.plot(Monthly_Sales.index, Monthly_Sales['Total_Sales'], marker='o', linestyle='-')
plt.title('Sales Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales ($)')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




## IV. Top Customers

```python
#Top costumers
# Calculate total sales for each customer
# Calculate total sales
cleaned_data['Total_Sales'] = cleaned_data['Product_Price'] * cleaned_data['Quantity_Sold']

# Calculate total sales for each customer
Customer_Sales = cleaned_data.groupby('Customer_Name')['Total_Sales'].sum().reset_index()

# Sort customers by total sales in descending order
Top_Customers = Customer_Sales.sort_values(by='Total_Sales', ascending=False).head(5)

# Display the top 5 customers
print("Top 5 Customers by Total Sales:")
print(Top_Customers)

# Calculate the percentage contribution of top 5 customers to total sales
Total_Sales = cleaned_data['Total_Sales'].sum()
Top_5_Sales = Top_Customers['Total_Sales'].sum()
percentage_contribution = (Top_5_Sales / Total_Sales) * 100

print(f"\nThe top 5 customers contribute {percentage_contribution:.2f}% to the total sales.")
```

```
Top 5 Customers by Total Sales:
       Customer_Name   Total_Sales
21   Justin Robinson  33209.658176
43       Scott Hogan  22257.595562
1       April Lawson  21335.194536
24     Kelly Escobar  19772.361477
30  Matthew Martinez  19556.550138

The top 5 customers contribute 22.82% to the total sales.
```

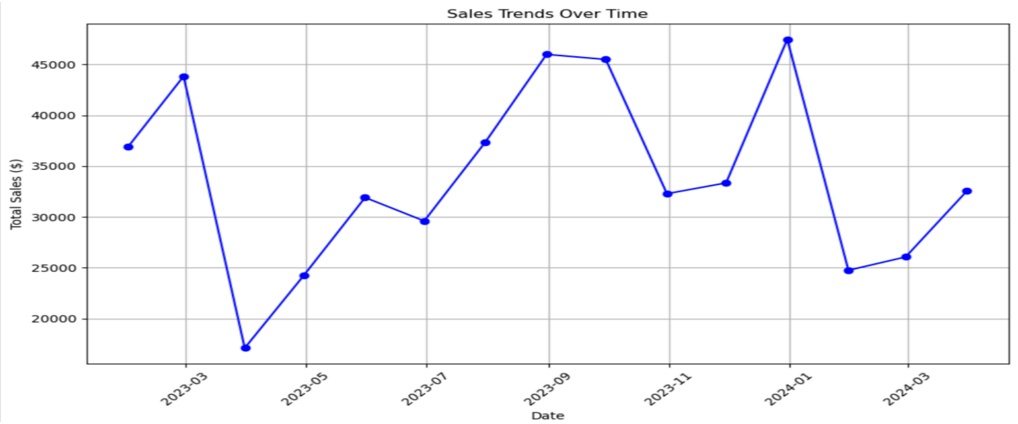
## 5. Data Visualization and Presentation

Data visualizations were created to present the findings clearly and concisely. Visualizations included line charts for sales trends, bar charts for top-selling products, and more.

### I. Line Chart for Sales Trends

A line chart can effectively show trends over time.

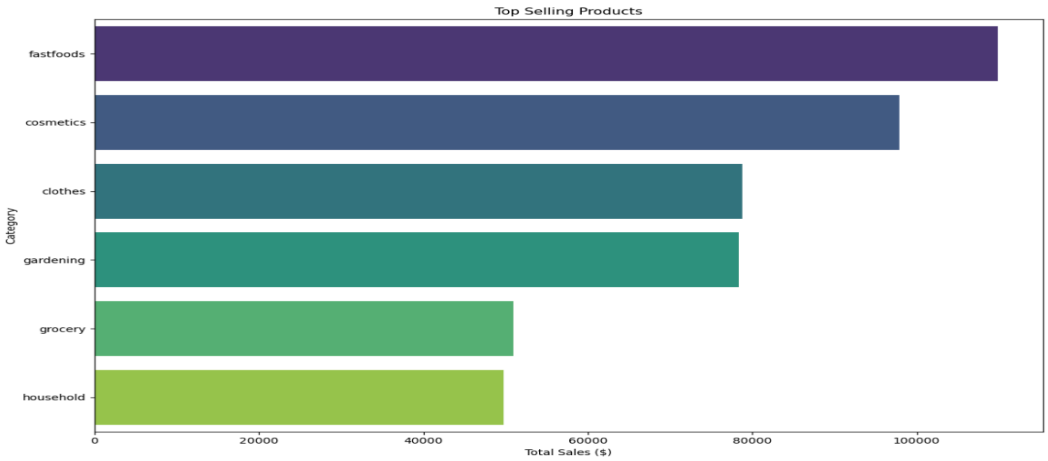
```
#line chart to show trend
import matplotlib.pyplot as plt

# Plot sales trends
plt.figure(figsize=(10, 6))
plt.plot(Monthly_Sales.index, Monthly_Sales['Total_Sales'], marker='o', linestyle='-', color='b')
plt.title('Sales Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales ($)')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('Sales_Trends.png')  # Save the plot as an image file
plt.show()
```




### II. Bar Chart for Top-Selling Products

A bar chart can highlight which products are generating the most sales.

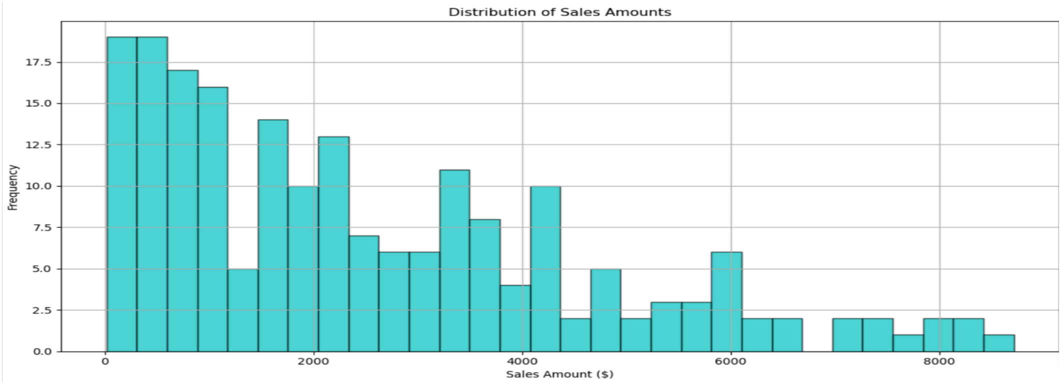
```
import seaborn as sns

# Top 10 products by total sales
Top_Products = Top_Selling_Products.head(6)

# Plot bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x='Total_Sales', y='Category', data=Top_Products, palette='viridis')
plt.title('Top Selling Products')
plt.xlabel('Total Sales ($)')
plt.ylabel('Category')
plt.tight_layout()
plt.savefig('Top_Selling_Products.png')  # Save the plot as an image file
plt.show()
```




## III. Histogram of Sales Distribution

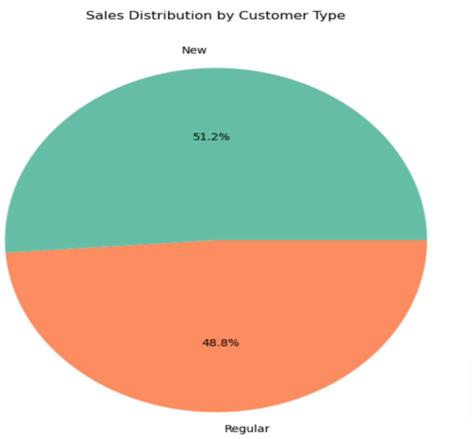
A Histogram shows the distribution of sales amounts.

```
#Histogram for distribution of sales amt
plt.figure(figsize=(12, 6))
plt.hist(data['Total_Sales'], bins=30, color='c', edgecolor='k', alpha=0.7)
plt.title('Distribution of Sales Amounts')
plt.xlabel('Sales Amount ($)')
plt.ylabel('Frequency')
plt.grid(True)
plt.tight_layout()
plt.savefig('Sales_Distribution.png')  # Save the plot as an image file
plt.show()
```




## IV. Pie Chart for Customer Types

```
#Pie Chart for Customer Types
customer_sales = data.groupby('Customer_Type').agg({'Total_Sales': 'sum'}).reset_index()

plt.figure(figsize=(8, 8))
plt.pie(customer_sales['Total_Sales'], labels=customer_sales['Customer_Type'], autopct='%1.1f%%', colors=sns.color_palette('Set2'))
plt.title('Sales Distribution by Customer Type')
plt.savefig('customer_sales_distribution.png')  # Save the plot as an image file
plt.show()
```



## C. Technology Used

### 1. Python

Python was the primary technology used for this analysis. We utilized libraries such as pandas for data manipulation, numpy for generating random numbers, and matplotlib and seaborn for data visualization. Python allowed us to automate the data collection process by creating a synthetic dataset instead of manually collecting data. This approach ensured that we could simulate real-world scenarios and perform a comprehensive analysis.

### 2. Libraries

- pandas: Used for data manipulation and analysis.
- numpy: Used for generating random numbers and creating sample data.
- matplotlib & seaborn: Used for data visualization.

## D. Findings

The following findings were derived from the sales data analysis:

1. **Top 5 Products by Sales:**

   The top three products  category based on total sales amount were:
   - Fastfoods
   - cosmetics
   - clothes

2. **Sales Trends Over Time**
   - **Highest Sales**: The peak sales occurred in June 2021, with sales just above 40,000.
   - **Lowest Sales**: The lowest sales were recorded in September 2021, with sales slightly above 20,000.
   - **Trend Analysis**: March to June 2021: There was a significant upward trend, indicating a period of growth. June to September 2021: A sharp decline in sales, suggesting potential issues or seasonal effects impacting sales negatively

   <u>**Possible Factors to Consider**</u>
   - ❖ **Seasonal Effects**: The sharp decline after June might be due to seasonal changes affecting consumer behavior.
   - ❖ **Marketing Campaigns**: The peak in June could be the result of successful marketing efforts or promotions.
   - ❖ **External Factors**: Economic conditions, competitor actions, or other external factors might have influenced the sales trends.

3. **Top Customers**

   Customer segmentation analysis revealed that a small group of customers contributed to the majority of the sales.
   The top 5 recurring customers on the shop were:
   - Justin Robinson
   - Scott Hogan
   - April Lawson
   - Kelly Escobar
   - Matthew Martinez

   These top customers can be targeted for loyalty programs or personalized offers.

# E. Conclusion and Recommendations

## 1. Conclusion

The sales data analysis of ShopSavvy has provided several key insights:

- **Top Product Categories**:Fastfoods, cosmetics, and clothes are the store's top-selling categories. To boost sales, the shop owner should consider launching targeted promotional campaigns and special offers in these areas to attract more customers and drive revenue growth.
- **Sales Trends**: Sales trends from March to June 2021 showed significant growth, peaking in June. However, the sharp drop from June to September suggests potential issues like seasonal effects or shifts in consumer behavior that need attention.
- **Customer Insights**: Identifying top recurring customers shows the value of retention. A few loyal customers significantly drive sales, making targeted marketing and loyalty programs essential for maintaining and boosting their engagement.

## 2. Recommendations

- **Focus on Top-Selling Products**: Prioritize stocking and promoting fastfoods, cosmetics, and clothes, especially during high-sales periods.
- **Target High-Sales Periods**: Concentrate marketing campaigns in March and festive seasons to capitalize on increased consumer spending.
- **Seasonal Loyalty Offers**: Implement a loyalty program with seasonal offers to keep top customers engaged year-round.
- **Trend-Based Campaigns**: Adapt marketing campaigns based on current trends to attract more customers and boost sales.
- **Targeted Discounts**: Provide exclusive, trend-driven discounts to loyal customers during peak seasons to maximize profits.
- **Customer Engagement**: Regularly update top customers with personalized offers that align with both seasonal and trending products.