Sales Data Cleaning and Analysis Report

Author: Osman

Dated: Jan 2025

Table of Contents

Executive Summary

This report presents an analysis of sales data to identify key trends, performance drivers, and actionable insights. The analysis focuses on understanding sales by product category, sales trends over time, top-performing customers, and regional performance.

Key Findings:

1. **Sales by Product Category:** The highest revenue-generating categories are identified, revealing their contribution to overall sales.
2. **Sales Over Time:** Sales trends indicate seasonality and growth patterns, with certain months showing peak activity.
3. **Top 10 Customers:** A small group of customers contribute significantly to the revenue, suggesting potential for loyalty programs.
4. **Top Categories by Sales:** Specific product categories dominate sales, highlighting opportunities for targeted marketing.
5. **Top Countries by Sales:** Geographic analysis reveals the most profitable regions and potential areas for expansion.

These findings provide a foundation for data-driven decision-making to optimize performance and maximize revenue opportunities.

Introduction

- **Purpose of the Report**: This report aims to provide a comprehensive analysis of sales data to identify trends, key growth areas, and actionable insights for improving business performance.
- **Scope of the Analysis**: The analysis covers a specific period and geographic regions as defined by the dataset. It includes data on product categories, sales transactions, customer details, and regional performance metrics.
- **Objectives**:
    - Understand overall sales trends and seasonal variations.
    - Identify top-performing product categories and key contributors to revenue.
    - Analyze customer behavior to recognize top customers and improve retention strategies.
    - Evaluate regional sales performance to determine high-potential markets.

Data Description

Dataset Overview:

The dataset contains information on sales transactions, including product categories, sales amounts, customer details, and regional data. Each row represents an individual transaction, while columns capture relevant details such as product ID, category, sale amount, customer ID, and geographic location.

Data Fields:

Key fields in the dataset include:

- **Transaction ID**: A unique identifier for each sales transaction.
- **Product Category**: The category to which a product belongs.
- **Sales Amount**: The revenue generated from each transaction.
- **Customer ID**: A unique identifier for customers.
- **Region/Country**: Geographic information about where the transaction occurred.
- **Date of Sale**: The date on which the transaction took place.

Dataset Period:

The dataset covers transactions recorded over a defined period, capturing both seasonal and long-term sales patterns.

Data Cleaning:

Basic data cleaning was performed to ensure accuracy and consistency. This included:

- Removing duplicates and erroneous records.
- Addressing missing or incomplete data.
- Standardizing categories and date formats.
- Filtering for valid transactions within the specified analysis scope.

Data Cleaning and Preparation

This section provides an overview of the cleaning and preprocessing performed to ensure the data is ready for analysis.

- **Handling Missing Values**: Missing data points were identified and addressed through imputation or removal to avoid skewing the analysis.
- **Duplicate Records**: Duplicate entries were identified and removed to maintain data accuracy.
- **Data Consistency**: Standardized data formats, particularly for dates, currencies, and categorical variables.
- **Outlier Treatment**: Addressed extreme values to ensure accurate trend analysis.
- **Categorical Encoding**: Transformed categorical variables into a format suitable for analysis.

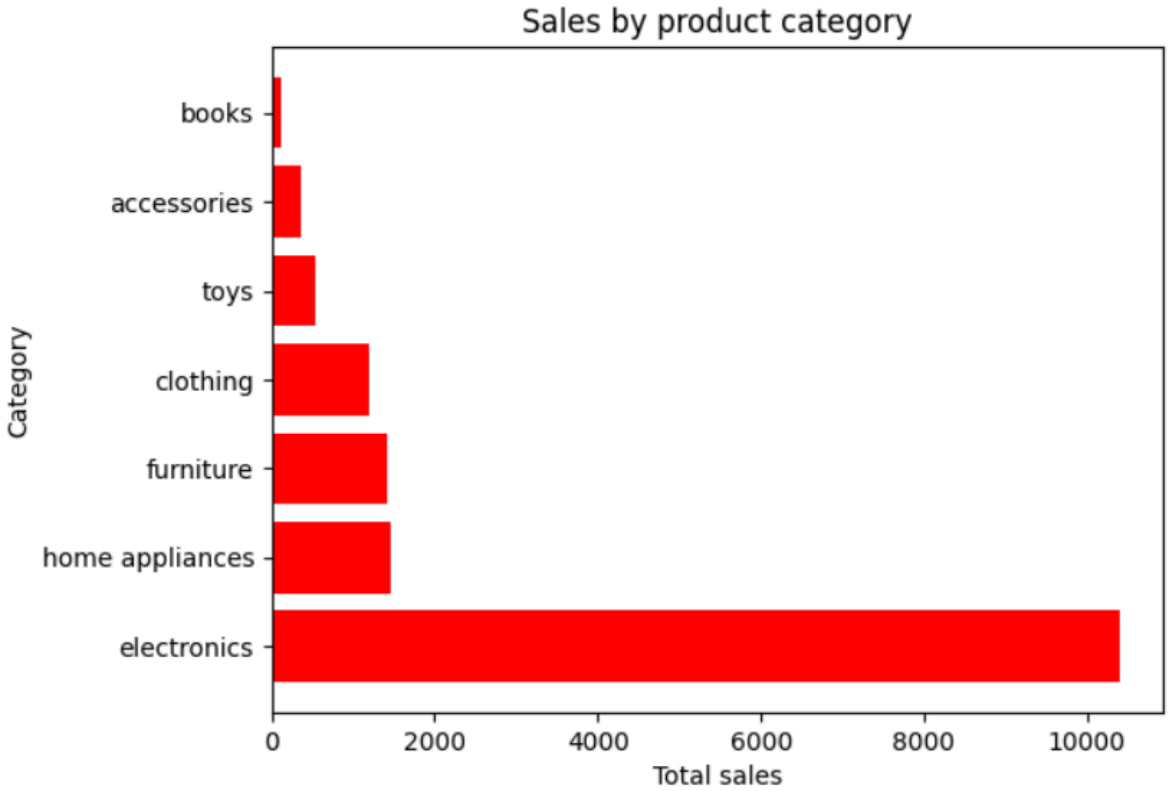
By applying these steps, the dataset was cleaned to allow for reliable and meaningful insights in subsequent analysis.

Analysis and Insights

Sales by Product Category

This section highlights the distribution of sales across various product categories:

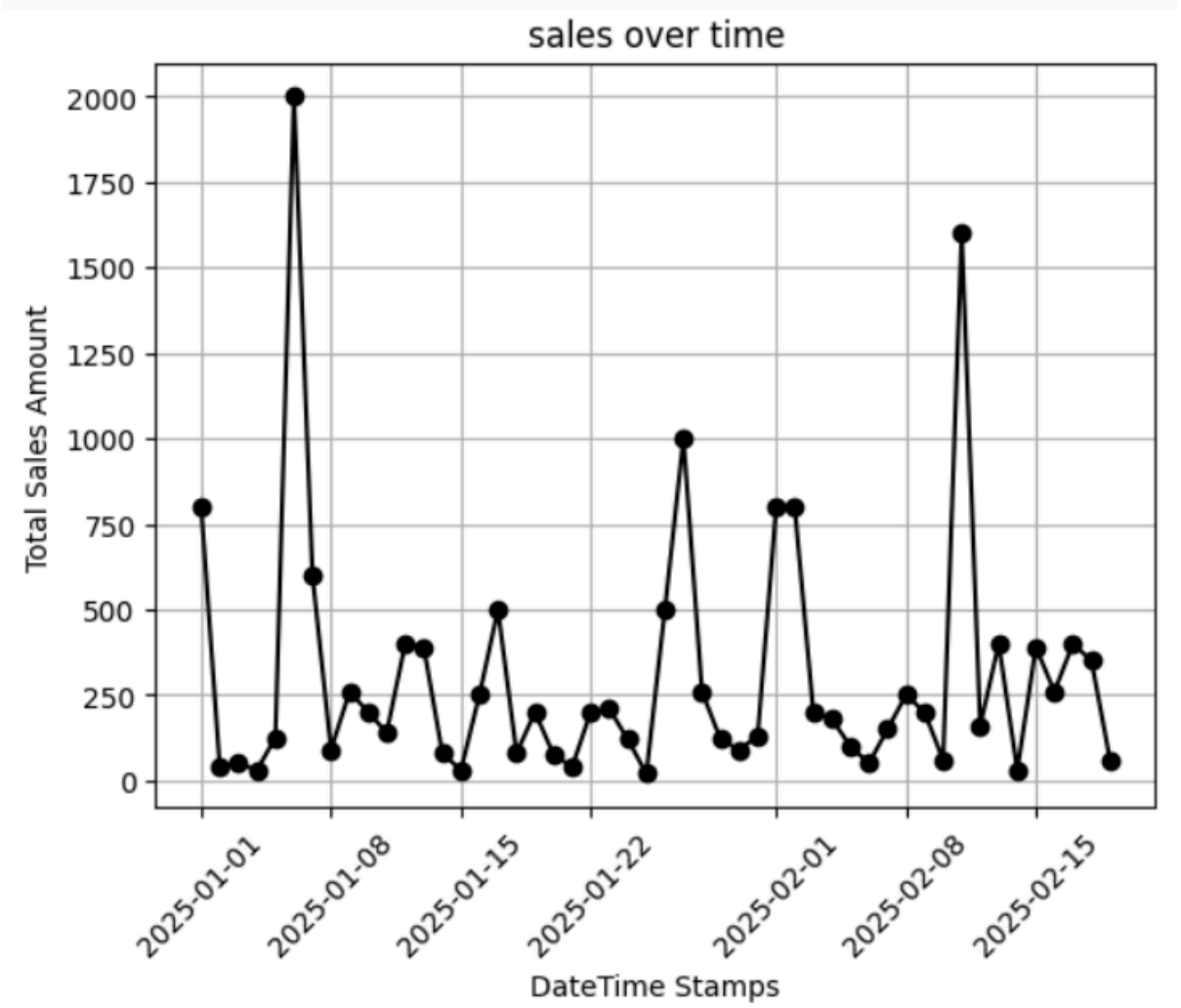
- Identifies the top-selling categories.
- Examines the revenue contribution of each category.
- Provides insights into product performance trends.



Sales Over Time

Analysis of sales trends over time includes:

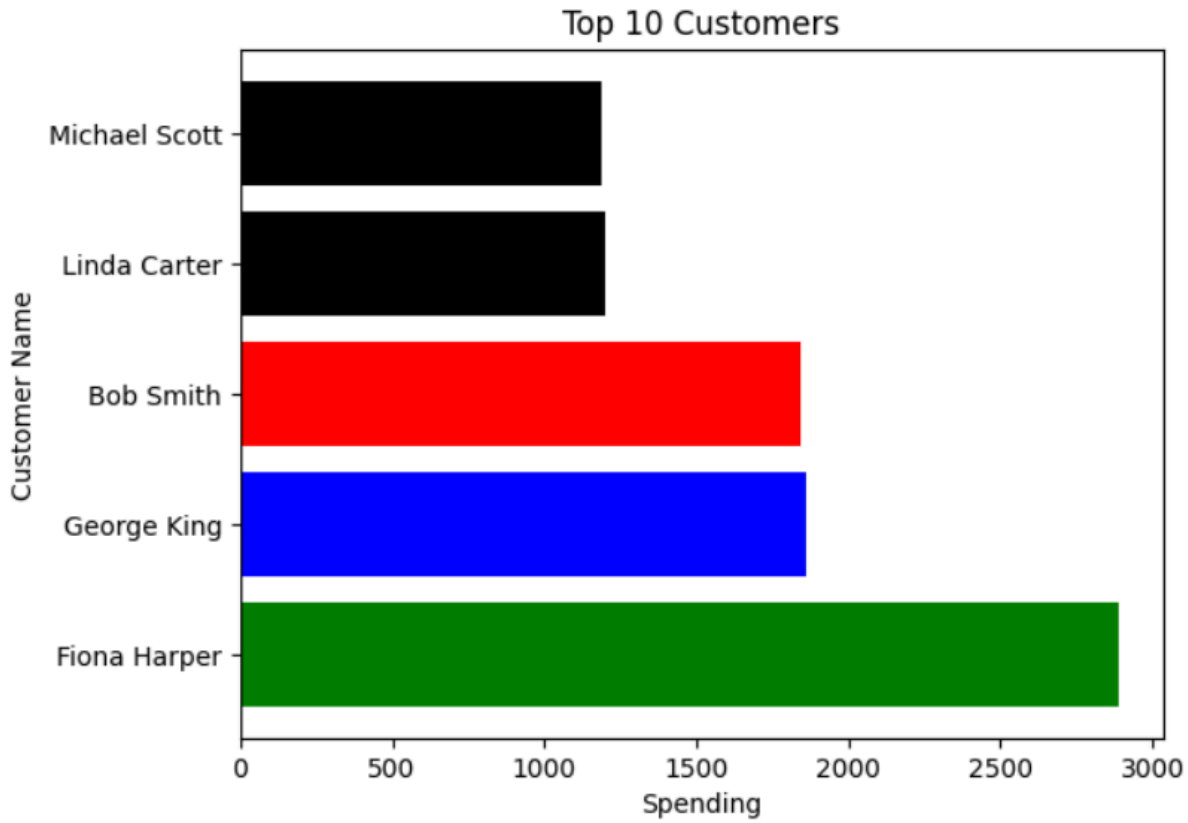
- A breakdown of monthly, quarterly, or yearly sales performance.
- Seasonal patterns and peak sales periods.
- Comparative growth rates over different timeframes.



Top 10 Customers

This section focuses on identifying and analyzing the top customers based on sales volume:

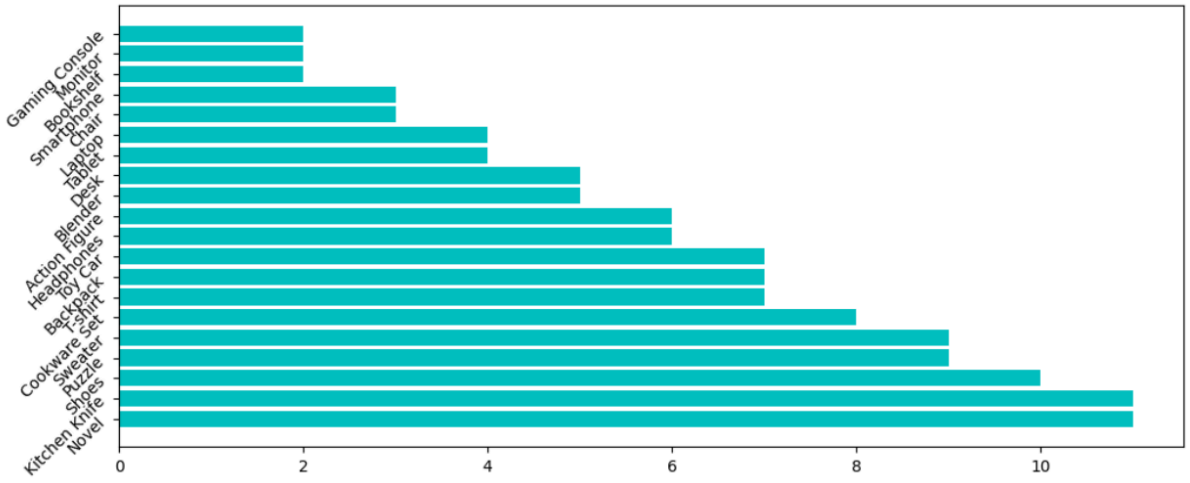
- List of top 10 customers with their sales contribution.
- Insights into customer buying patterns and frequency.
- Potential opportunities for customer retention or upselling.



Top Category by Sale

An in-depth analysis of the most successful product category:

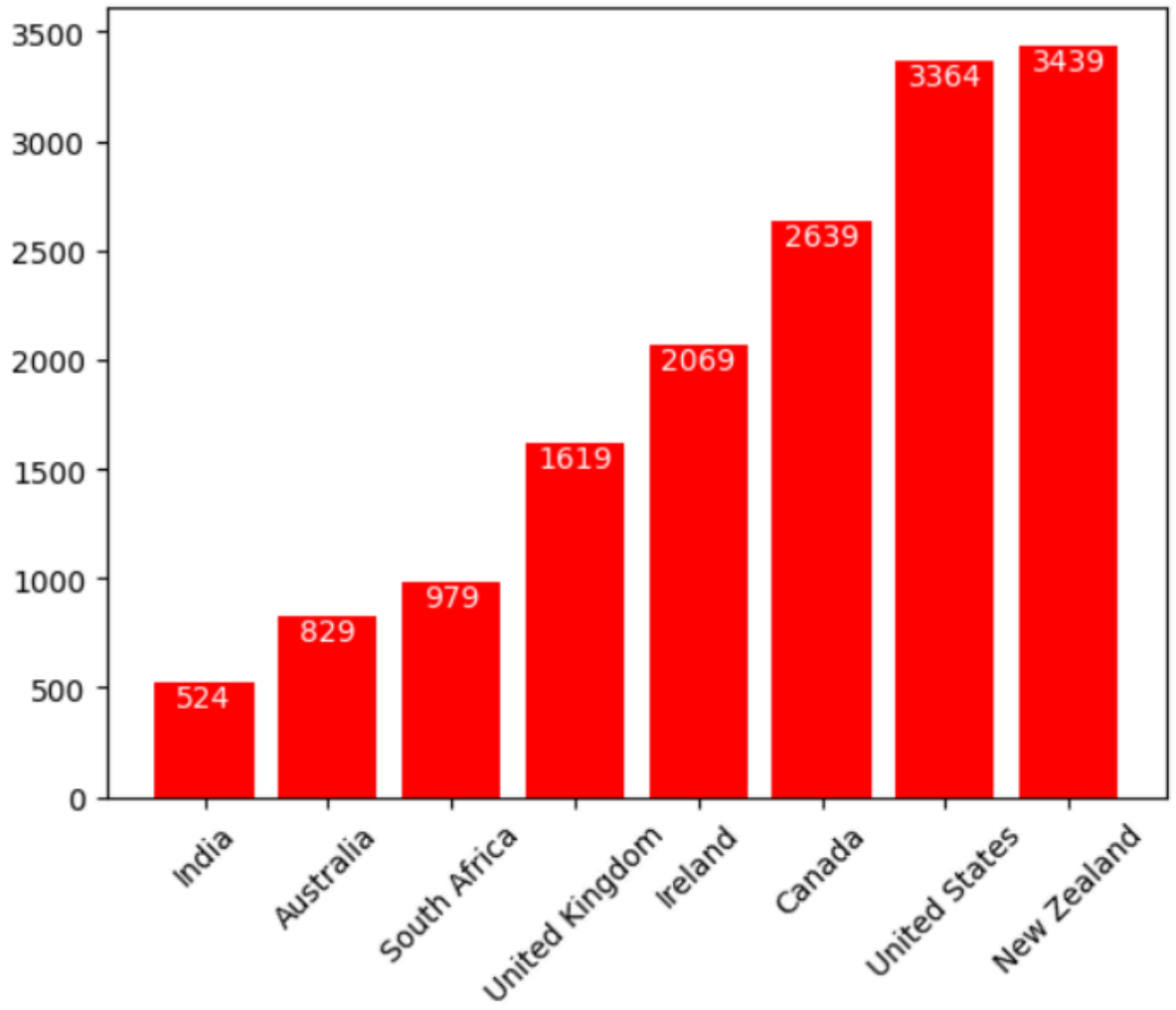
- Identification of the leading category and its sales trends.
- Factors contributing to its success, such as customer demand or marketing efforts.
- Recommendations for replicating success in other categories.



Top Countries by Sale

Evaluation of regional performance to understand geographic sales distribution:

- Identification of top-performing countries or regions.
- Regional sales trends and growth rates.
- Suggestions for targeted marketing or expansion in high-performing regions.


3500
3000
2500
2000
1500
1000
500
0
524
829
979
1619
2069
2639
3364
3439
India
Australia
South Africa
United Kingdom
Ireland
Canada
United States
New Zealand

Recommendations

Based on the insights derived from the analysis, the following actionable recommendations are proposed:

Enhance Top Product Categories

- Invest in the top-performing product categories through targeted promotions and increased inventory.
- Analyze customer feedback to maintain or improve product quality.
- Explore complementary products or bundles to increase average order value.

Leverage Seasonal Trends

- Capitalize on peak sales periods with seasonal promotions and advertising campaigns.
- Prepare inventory and staffing ahead of high-demand periods to meet customer needs.

Strengthen Customer Relationships

- Develop loyalty programs or personalized offers for the top 10 customers to encourage repeat business.
- Use data analytics to identify cross-selling or upselling opportunities tailored to customer preferences.

Expand in High-Performing Regions

- Focus marketing efforts and resource allocation in top-performing countries or regions.
- Conduct market research to understand the needs of customers in these regions better.

- Consider regional partnerships or distribution channels to boost local presence.

Address Underperforming Areas

- Investigate reasons for low sales in underperforming categories or regions.
- Reassess pricing strategies, marketing efforts, and product offerings in these areas.
- Experiment with pilot programs or discounts to stimulate demand.

The sales analysis provides key insights into business performance, customer behavior, and regional trends. These insights reveal significant opportunities for growth and optimization:

- **Top Product Categories**: Identifying and enhancing investment in high-performing categories can drive revenue growth.
- **Seasonal Trends**: Leveraging peak sales periods with strategic planning can maximize profitability.
- **Customer Insights**: Strengthening relationships with top customers through tailored programs ensures long-term loyalty.
- **Regional Performance**: Expanding operations in high-performing regions and addressing challenges in low-performing ones can improve overall market presence.

By implementing the recommendations outlined in this report, the business can effectively address current challenges, capitalize on growth opportunities, and achieve a competitive advantage in the market.

Source Code For the Project

```
!pip install pandas matplotlib mysql-connector-python
SQLAlchemy mysqlclient
```

```
# Project Dependencies
import pandas as pd
import matplotlib.pyplot as plt
import mysql.connector # mysql-connector-python
```

```
# create a connection with mysql database using
mysql-connector-python module
db = mysql.connector.connect(
    host="enter host name here",
    user="enter user name here",
    password="enter user password here",
    database="enter database name here"
)
```

```
from sqlalchemy import create_engine,text

engine =
create_engine('mysql://username:password@hostname/databas
e')

with engine.connect() as cur:
    result = cur.execute(text("select
customer_id,count(*) from orders group by customer_id"))

    orders = result.fetchall()
    for i in orders:
        print(i)
```

```
x = pd.read_sql('select * from products where 1',engine)

print(x)
```

```
# Fetching data of customers

customer_query = "select * from customers"
customer_df = pd.read_sql(customer_query,engine)

# Fetching data of products
products_query = "select * from products"
products_df = pd.read_sql(products_query,engine)

# Fetching data of orders
# orders_query = "select * from orders"
orders_query = """select o.order_id, o.customer_id,
o.product_id, o.order_date, o.quantity, o.price_per_unit,
o.total_amount,
c.customer_name, p.product_name, p.category
from orders o
join customers c on o.customer_id = c.customer_id
join products p on o.product_id = p.product_id
"""

# to sort fetched data in decending order we use below
line
# order by p.category desc

orders_df = pd.read_sql(orders_query,engine)

# print(customer_df)
# print(products_df)
print(orders_df)

# customer_id,customer_name,email,country
```

```
# product id, product_name,category,stock_quantity, price
# order_id, customer_id, product_id, order_date,
quantity,price_per_unit, total_amount
```

```
# grouping categories by amount

category_sales =
orders_df.groupby('category')['total_amount'].sum()
category_sales =
category_sales.sort_values(ascending=False)
print(category_sales)
```

```
# putting data in bar chart

# Accessing the category names
categories = category_sales.index.tolist()

# Accessing the sales amount for each category
sales_amounts = category_sales.values.tolist()

print(categories)
print(sales_amounts)
plt.barh(categories,sales_amounts,color='r')

plt.title('Sales by product category')
plt.xlabel('Total sales')
plt.ylabel('Category')

plt.show()
```

```
# Sales over Time (Sales vs Time)
```

```
print(orders_df)
# print order_date

# print(orders_df['order_date']) # you can print
order_date column from orders table here
# convert order_date to date format for reading with
pandas and lets save it again in the data frame

orders_df['order_date'] =
pd.to_datetime(orders_df['order_date'])
print(orders_df['order_date']) # printing dates from
orders table after converting them to datetime format
```

```
# now lets for example calculate sales over time

# sot variable for sales over time
sot =
orders_df.groupby('order_date')['total_amount'].sum()
sot_date = sot.index.tolist()
sot_data = sot.values.tolist()

plt.plot(sot_date,sot_data,marker='o', color='k')

plt.title('sales over time')
plt.xlabel('DateTime Stamps')
plt.ylabel('Total Sales Amount')

plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

```
# top customers by total spending
spending =
```

```
orders_df.groupby('customer_name')['total_amount'].sum()
spending = spending.sort_values(ascending=False)
# print(spending.head(7))

spending = spending.head(5)

tc_key = spending.index.tolist()
tc_value = spending.values.tolist()

myc = ['g','b','r','k','k']
plt.barh(tc_key,tc_value,color=myc)

plt.title('Top 10 Customers')
plt.xlabel('Spending')
plt.ylabel('Customer Name')

# plt.grid(True)

plt.show()
```

```
print(products_df)

# Sales per product matrix

psales =
orders_df.groupby('product_name')['quantity'].sum()
psales = psales.sort_values(ascending=False)
print(psales)

topcat_key = psales.index.tolist()
topcat_value = psales.values.tolist()

plt.figure(figsize=(12,5)) # manage chart size
plt.barh(topcat_key,topcat_value,color='c')
```

```
plt.yticks(rotation=45)
plt.show()
```

```
customer_df
print(orders_df)
# Sales by country

country_orders = pd.merge(orders_df,
customer_df,on='customer_id')
# print(country_orders)

c_sales =
country_orders.groupby('country')['total_amount'].sum()

c_sales = c_sales.sort_values()

print(c_sales)

country_key = c_sales.index.tolist()
country_value = c_sales.values.tolist()

fig, ax = plt.subplots()

bars = ax.bar(country_key,country_value,color='r')

for bar in bars:
    yval = bar.get_height()
    yval = int(yval)
    ax.text(bar.get_x() + bar.get_width() / 2, yval +
0.5, str(yval),
            ha='center', va='top',color='w')

plt.xticks(rotation=45)

plt.show()
```

Database Structure

summary of each table:

1. `customers` Table

- **Columns:**
  - `customer_id`: Integer (Primary Key)
  - `customer_name`: VARCHAR(100)
  - `email`: VARCHAR(100)
  - `country`: VARCHAR(50)
- **Purpose:** Stores customer information, including names, email addresses, and their country.

2. `orders` Table

- **Columns:**
  - `order_id`: Integer (Primary Key)
  - `customer_id`: Integer (Foreign Key referencing `customers.customer_id`)
  - `product_id`: Integer (Foreign Key referencing `products.product_id`)
  - `order_date`: DATE
  - `quantity`: Integer
  - `price_per_unit`: FLOAT
  - `total_amount`: FLOAT
- **Purpose:** Stores order details, including which customer ordered which product, the order date, quantity, price, and total amount.

3. `products` Table

- **Columns:**
  - `product_id`: Integer (Primary Key)
  - `product_name`: VARCHAR(100)
  - `category`: VARCHAR(50)
  - `stock_quantity`: Integer
  - `price`: FLOAT
- **Purpose:** Stores product details, such as name, category, available stock, and price.

Database Setup Script

```
-- Host: localhost    Database: salesrecord
```

```
-- Table structure for table `customers`


CREATE TABLE `customers` (
  `customer_id` int NOT NULL,
  `customer_name` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `country` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Table structure for table `orders`

CREATE TABLE `orders` (
  `order_id` int NOT NULL,
  `customer_id` int DEFAULT NULL,
  `product_id` int DEFAULT NULL,
  `order_date` date DEFAULT NULL,
  `quantity` int DEFAULT NULL,
  `price_per_unit` float DEFAULT NULL,
  `total_amount` float DEFAULT NULL,
  PRIMARY KEY (`order_id`),
  KEY `customer_id` (`customer_id`),
  KEY `product_id` (`product_id`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES
`customers` (`customer_id`),
  CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`product_id`) REFERENCES
`products` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Table structure for table `products`

CREATE TABLE `products` (
  `product_id` int NOT NULL,
  `product_name` varchar(100) DEFAULT NULL,
  `category` varchar(50) DEFAULT NULL,
  `stock_quantity` int DEFAULT NULL,
  `price` float DEFAULT NULL,
```

```
  PRIMARY KEY (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Database Populating Script

```
-- Dumping data for table `customers`

INSERT INTO `customers` VALUES (101,'Alice
Johnson','alice.johnson@example.com','United States'),(102,'Bob
Smith','bobsmith@example.com','Canada'),(103,'Charlie
Brown','charlie.brown@example.com','United Kingdom'),(104,'Daisy
Ridley','daisy.ridley@example.com','Australia'),(105,'Ethan
Hunt','ethan.hunt@example.com','India'),(106,'Fiona
Harper','fiona.harper@example.com','New Zealand'),(107,'George
King','george.king@example.com','Ireland'),(108,'Hannah
Lee','hannah.lee@example.com','South Africa'),(109,'Ian
Bell','ian.bell@example.com','United States'),(110,'Jenny
Moore','jenny.moore@example.com','Canada'),(111,'Kevin
Hart','kevin.hart@example.com','United States'),(112,'Linda
Carter','linda.carter@example.com','United Kingdom'),(113,'Michael
Scott','michael.scott@example.com','United States'),(114,'Nancy
Drew','nancy.drew@example.com','Australia'),(115,'Oscar
Wilde','oscar.wilde@example.com','Ireland'),(116,'Paul
Allen','paul.allen@example.com','India'),(117,'Quincy
Adams','quincy.adams@example.com','New Zealand'),(118,'Rebecca
Black','rebecca.black@example.com','South Africa'),(119,'Steve
Rogers','steve.rogers@example.com','Canada'),(120,'Tony
Stark','tony.stark@example.com','United States');
```

```
-- Dumping data for table `orders`


INSERT INTO `orders` VALUES
(2001,101,5001,'2025-01-01',1,799.99,799.99),(2002,102,5002,'2025-01-02'
,2,19.99,39.98),(2003,103,5003,'2025-01-03',1,49.99,49.99),(2004,104,500
4,'2025-01-04',3,9.99,29.97),(2005,105,5005,'2025-01-05',5,24.99,124.95)
,(2006,106,5006,'2025-01-06',2,999.99,1999.98),(2007,107,5007,'2025-01-0
7',3,199.99,599.97),(2008,108,5008,'2025-01-08',1,89.99,89.99),(2009,109
,5009,'2025-01-09',2,129.99,259.98),(2010,110,5010,'2025-01-10',4,49.99,
199.96),(2011,111,5011,'2025-01-11',2,69.99,139.98),(2012,112,5012,'2025
-01-12',1,399.99,399.99),(2013,113,5013,'2025-01-13',3,129.99,389.97),(2
```

```
014,114,5014,'2025-01-14',4,19.99,79.96),(2015,115,5015,'2025-01-15',1,2
9.99,29.99),(2016,116,5016,'2025-01-16',1,249.99,249.99),(2017,117,5017,
'2025-01-17',1,499.99,499.99),(2018,118,5018,'2025-01-18',2,39.99,79.98)
,(2019,119,5019,'2025-01-19',1,199.99,199.99),(2020,120,5020,'2025-01-20
',5,14.99,74.95),(2021,101,5002,'2025-01-21',2,19.99,39.98),(2022,102,50
07,'2025-01-22',1,199.99,199.99),(2023,103,5011,'2025-01-23',3,69.99,209
.97),(2024,104,5015,'2025-01-24',4,29.99,119.96),(2025,105,5004,'2025-01
-25',2,9.99,19.98),(2026,106,5017,'2025-01-26',1,499.99,499.99),(2027,10
7,5006,'2025-01-27',1,999.99,999.99),(2028,108,5013,'2025-01-28',2,129.9
9,259.98),(2029,109,5018,'2025-01-29',3,39.99,119.97),(2030,110,5020,'20
25-01-30',6,14.99,89.94),(2031,111,5009,'2025-01-31',1,129.99,129.99),(2
032,112,5001,'2025-02-01',1,799.99,799.99),(2033,113,5012,'2025-02-02',2
,399.99,799.98),(2034,114,5019,'2025-02-03',1,199.99,199.99),(2035,115,5
008,'2025-02-04',2,89.99,179.98),(2036,116,5014,'2025-02-05',5,19.99,99.
95),(2037,117,5005,'2025-02-06',2,24.99,49.98),(2038,118,5010,'2025-02-0
7',3,49.99,149.97),(2039,119,5016,'2025-02-08',1,249.99,249.99),(2040,12
0,5003,'2025-02-09',4,49.99,199.96),(2041,101,5002,'2025-02-10',3,19.99,
59.97),(2042,102,5001,'2025-02-11',2,799.99,1599.98),(2043,103,5018,'202
5-02-12',4,39.99,159.96),(2044,104,5007,'2025-02-13',2,199.99,399.98),(2
045,105,5015,'2025-02-14',1,29.99,29.99),(2046,106,5013,'2025-02-15',3,1
29.99,389.97),(2047,107,5009,'2025-02-16',2,129.99,259.98),(2048,108,501
2,'2025-02-17',1,399.99,399.99),(2049,109,5011,'2025-02-18',5,69.99,349.
95),(2050,110,5004,'2025-02-19',6,9.99,59.94);
```

```
-- Dumping data for table `products`

INSERT INTO `products` VALUES
(5001,'Laptop','electronics',50,799.99),(5002,'T-shirt','clothing',20,19
.99),(5003,'Blender','home
appliances',10,49.99),(5004,'Novel','books',100,9.99),(5005,'Toy
Car','toys',500,24.99),(5006,'Smartphone','electronics',200,999.99),(500
7,'Headphones','electronics',150,199.99),(5008,'Chair','furniture',75,89
.99),(5009,'Desk','furniture',40,129.99),(5010,'Backpack','accessories',
300,49.99),(5011,'Shoes','clothing',250,69.99),(5012,'Tablet','electroni
cs',90,399.99),(5013,'Cookware Set','home
appliances',30,129.99),(5014,'Puzzle','toys',200,19.99),(5015,'Action
Figure','toys',150,29.99),(5016,'Bookshelf','furniture',20,249.99),(5017
,'Gaming
Console','electronics',25,499.99),(5018,'Sweater','clothing',80,39.99),(
5019,'Monitor','electronics',120,199.99),(5020,'Kitchen Knife','home
appliances',300,14.99);
```

Final Thoughts

The analysis of sales data provides valuable insights into key trends, customer behavior, and regional performance, highlighting significant opportunities for business growth and optimization.

Key findings include:

- Strong performance of specific product categories and high-potential regions.
- Seasonal patterns and their impact on sales trends.
- The importance of top customers in driving revenue growth.

By implementing the recommendations outlined in this report—focusing on strategic investments, customer retention, and operational efficiency—the organization can capitalize on these insights to enhance profitability and long-term success. Regular monitoring and the adoption of data-driven decision-making practices will ensure sustained improvement and adaptability to market changes.