GROUP PROJECT

SQL SALES PERFORMANCE ANALYSIS

Project owners

Mahinder Reddy

Jithin Varghese

Nikita Kumari

Henry Jose Tovar

Vanessa Zamora

Academic Program: Business Analytics

Course name: SQL and Data Analysis

Course: INFO8076 - Fall 2023 - Section 6

Instructor: Surabhi Singh

Conestoga College, December 10, 2023, Guelph, Ontario.

Table of contents

[1.](#_heading=h.gjdgxs) Project Goal Statement 2

[2.](#_heading=h.30j0zll) Objective 2

[3.](#_heading=h.1fob9te) Preliminary Exploration 3

[3.1 Introduction 3](#_heading=h.3znysh7)

[3.2](#_heading=h.2et92p0) Dataset Overview 3

[3.3](#_heading=h.tyjcwt) Loading the data 3

[4.](#_heading=h.3dy6vkm) Data Cleaning 4

[4.1 Findings 4](#_heading=h.1t3h5sf)

[5.](#_heading=h.4d34og8) Data Analysis 4

[5.2](#_heading=h.2s8eyo1) Business Logic: Identify top products in demand and their brands and details 4

[5.3](#_heading=h.17dp8vu) Business Logic: Identifying Valuable Customers. 6

[5.4](#_heading=h.3rdcrjn) Business Logic: Identify the sales cycle . 8

[6.](#_heading=h.lnxbz9) Conclusions 9

# Project Goal Statement

Goal: The "SQL Sales Performance Analysis" project aims to leverage SQL skills to analyze and derive meaningful insights from the sales data for an online seller. This project aims to provide a comprehensive understanding of sales performance, identify key trends, identify value buyers, and empower decision-makers with actionable information to enhance business strategies.

# Objective

We have formulated the below objectives to stay focused and derive the best results from our analysis.

* Data exploration & understanding: Explore the sales dataset to understand its structure, key variables, and data distribution.
* Data cleaning and normalization: Eliminate data redundancy and enhance data integrity in the table.
* Display SQL proficiency: Showcase proficiency in SQL by writing complex queries for data retrieval, aggregation, and analysis.
* Data Exporting & Visualization: Export the data and visualize it in Excel/ Power BI

# Preliminary Exploration

## 3.1 Introduction

In the project's initial phase, we explored the dataset to establish a foundational understanding of the data structure, key variables, and overall flavor of the dataset.

## Dataset Overview

The sales dataset, sourced from [[data source](https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-electronics-store/data%20https://rees46.com/)], comprises [850K] records and [8] attributes. The dataset encapsulates a wealth of sales transaction information, including order timestamps, product information, user identifiers, and pricing details.

Key Variables

* **event\_time:** Timestamp indicating the time of the sales event.
* **order\_id:** Unique identifier for each sales order.
* **product\_id:** Identifier for the sold product.
* **category\_id:** Identifier for the product category.
* **category\_code:** Code representing the product category.
* **brand:** Brand associated with the product.
* **price:** Monetary value of the product.
* **user\_id:** Unique identifier for each user.

The data table variable declaration below

CREATE TABLE sales(

event\_time TIMESTAMP with TIME ZONE,

order\_id BIGINT,

product\_id BIGINT,

category\_id BIGINT,

category\_code CHARACTER VARYING(100),

brand CHARACTER VARYING(100),

price MONEY,

user\_id BIGINT

);

## Loading the data

The data from the external CSV file was imported into the "sales" table using the COPY command. Post-import, a data integrity check was conducted to verify the accuracy of the loaded data. This involved cross-referencing a sample of records between the original dataset and the PostgreSQL table to confirm a successful import without data loss or corruption.

COPY sales (event\_time, order\_id, product\_id, category\_id, category\_code, brand, price, user\_id)

FROM 'D:\kz.csv\kz.csv'

DELIMITER ','

CSV HEADER

ENCODING 'UTF8';

# Data Cleaning

A preliminary data quality assessment was performed to identify any anomalies, missing values, or outliers. This check ensures the reliability of the dataset for subsequent analysis.

## 4.1 Findings

* Missing values/blanks were identified in critical columns and replaced with null values.
* Created an index to improve read operations performance.
* Outliers in pricing and quantity fields were detected and will be addressed during the data-cleaning phase.
* Timestamps were validated for consistency and adherence to the expected format.

# Data Analysis

**Key Assumption:** We don’t want to create an exhaustive report with all the queries, as it is already covered in the “.SQL file”. Hence, we considered adding a select few with corresponding queries, results, and visualization.

Queries are written with various agendas in mind, this is broadly divided into three subcategories:

* Identify Top products in demand and their brands and details so businesses have inventory plans.
* Identify valued customers so that business can focus on them for marketing promotions.
* Identify sales cycle to hire and capacity building for peak months.

## Business Logic: Identify top products in demand and their brands and details.

1. We started the analysis by finding the top-selling brands within the database. This is quite interesting, as we get a rough view of which brands are sold the most in the store. This is a valuable piece of information for future inventory stocking decisions.

SELECT brand, product\_id, COUNT(order\_id) AS sales\_count

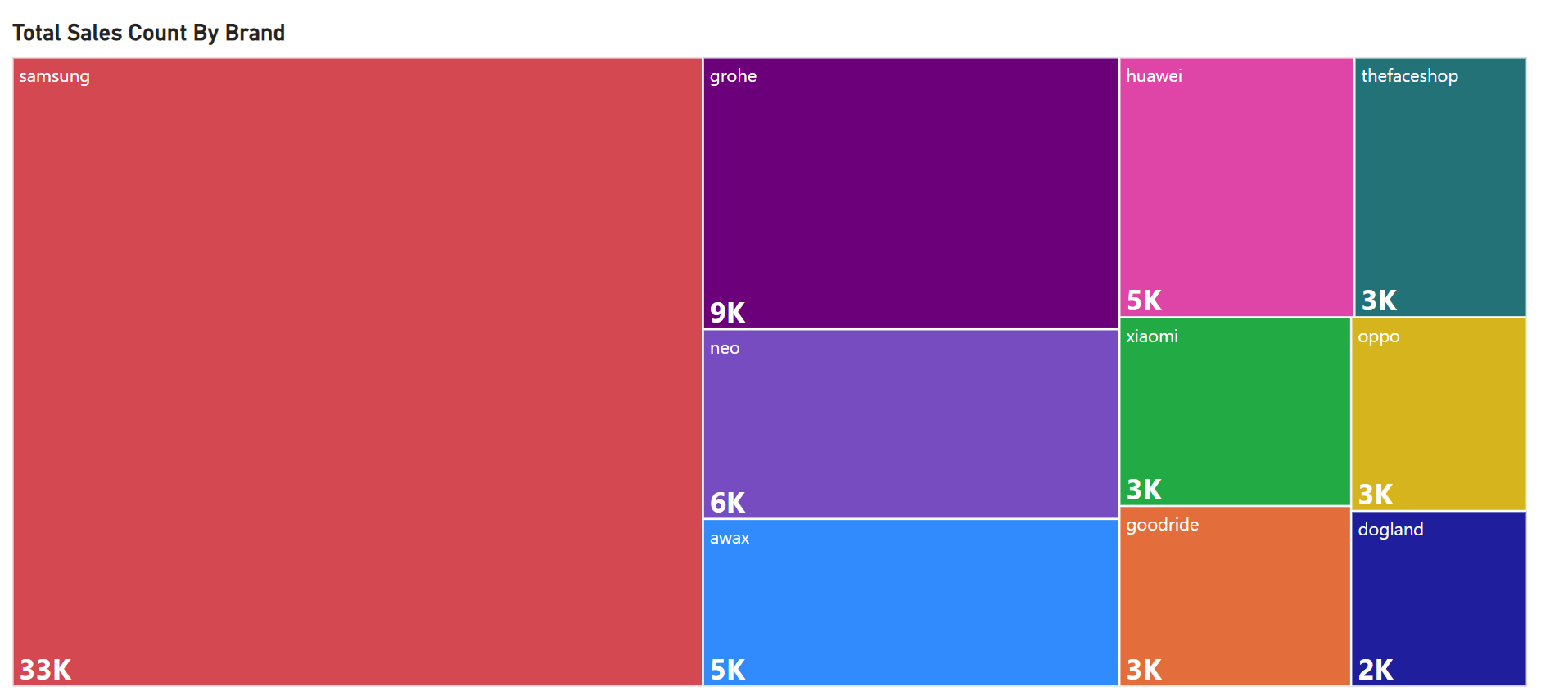
FROM sales

WHERE brand IS NOT NULL

GROUP BY brand, product\_id

ORDER BY sales\_count DESC

Limit 20;



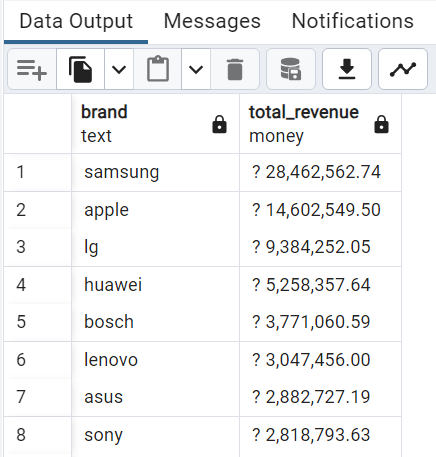
1. This query runs to find the total revenue contribution of each brand. Trim will remove the leading and trailing spaces. Coalesce Replaces NULL values with the 'Unknown' function in grouping. This view is excellent as seller can understand which brands are most sold under their account.

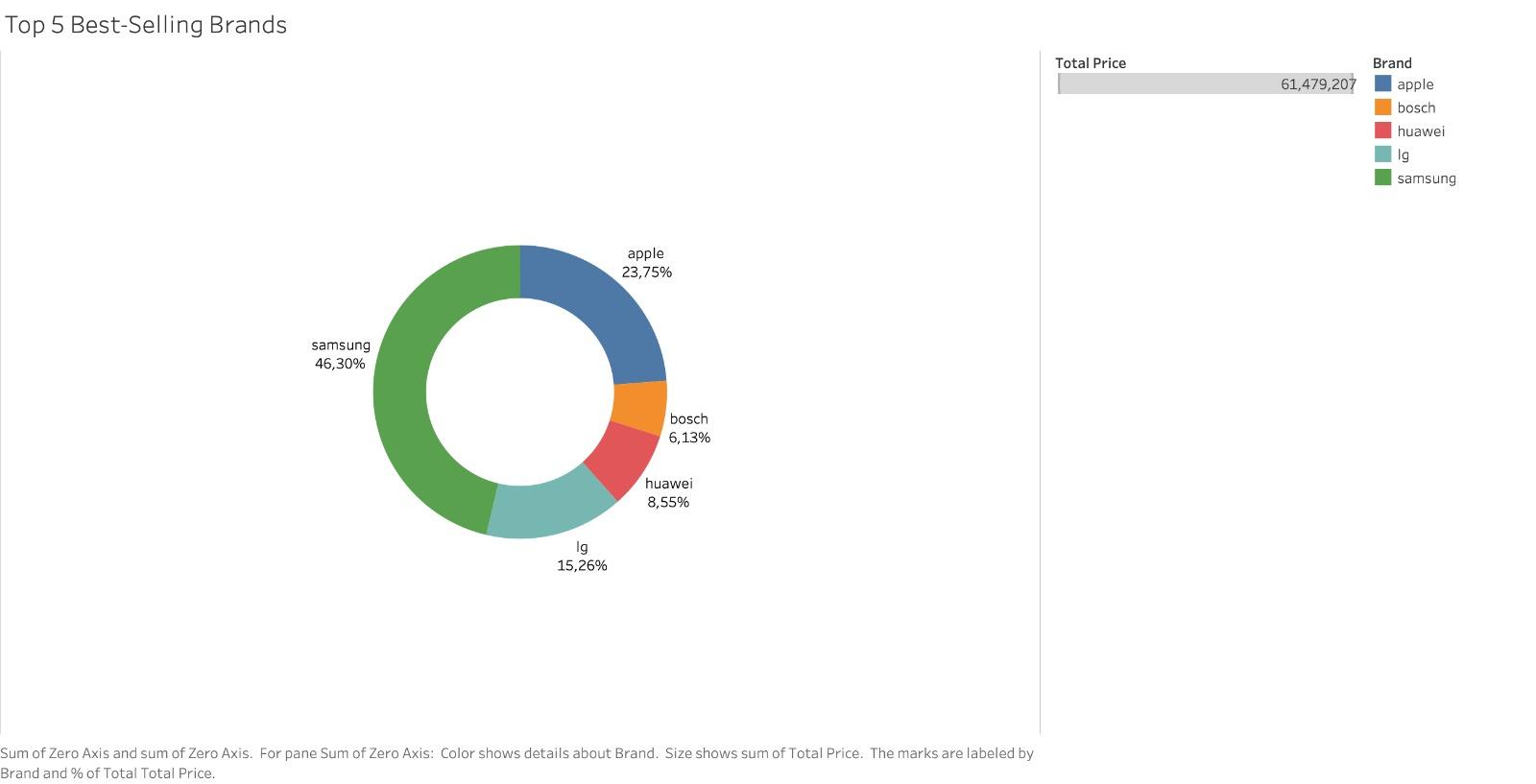
SELECT COALESCE(NULLIF(TRIM(brand), ''), 'Unknown') AS brand, SUM(price) AS total\_revenue

FROM sales

GROUP BY COALESCE(NULLIF(TRIM(brand), ''), 'Unknown')

ORDER BY total\_revenue DESC; -- 726 rows impacted





## Business Logic: Identifying Valuable customers.

1. This query identifies the user who made the highest purchase on a specific date, in this instance, on "Victoria Day - 22 May 2020."

SELECT user\_id, SUM(price) AS total\_purchase

FROM sales

WHERE DATE(event\_time) = '2020-05-22' -- Replace with the desired date

GROUP BY user\_id

ORDER BY total\_purchase DESC

LIMIT 1;

Such functions help to profile customers, for future target marketing, customer retention and sales optimization.

1. Identifying repeat customers

SELECT user\_id, COUNT(\*) AS purchase\_count

FROM sales

WHERE user\_id IS NOT NULL

GROUP BY user\_id

HAVING COUNT(\*) > 1

ORDER BY purchase\_count DESC; ---20895 rows impacted

This function helps to identify customers who have demonstrated repeat purchase behavior, showing loyalty to the store. Businesses can use this information to design targeted loyalty programs, exclusive offers, or personalized communication strategies to retain and further engage these loyal customers.

1. Identifying repeat customers & classification

WITH t1 AS (

SELECT user\_id,

CASE WHEN COUNT(\*) = 1 THEN 'Single Purchase'

WHEN COUNT(\*) = 2 THEN 'Occasional Buyer'

WHEN COUNT(\*) >= 3 THEN 'Frequent Buyer'

ELSE 'No Purchases' END AS user\_category

FROM sales

WHERE user\_id IS NOT NULL

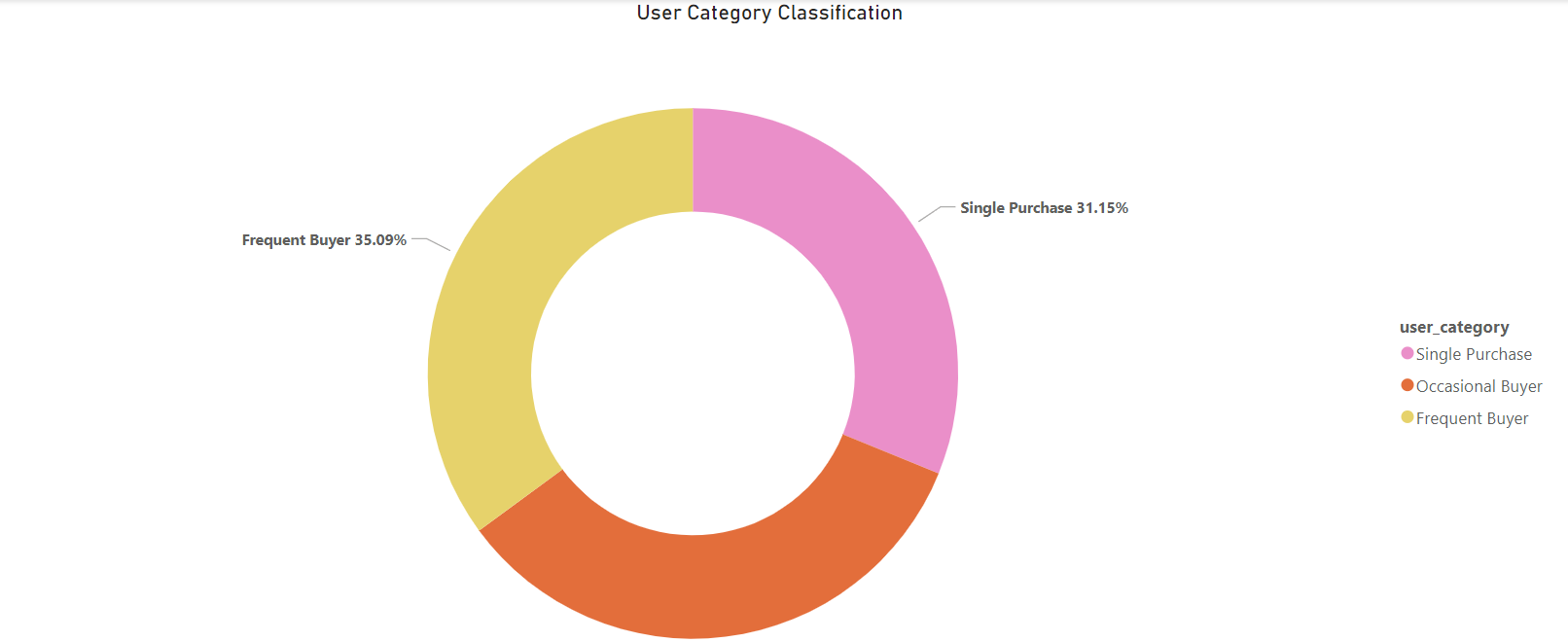
GROUP BY user\_id

)

SELECT COUNT(user\_id) AS no\_of\_users, user\_category

FROM t1

GROUP BY user\_category;



## Business Logic: Identify sales cycle.

1. Query to classify revenue by seasons:(case ) This query shows the total sales for each season, considering that January to March is winter, April to June is spring, July to September is summer and October to December is autumn.

SELECT season, SUM(total\_price) AS total\_sales\_by\_season

FROM (SELECT

EXTRACT(MONTH FROM event\_time) AS month,

SUM(price) AS total\_price,

CASE WHEN EXTRACT(MONTH FROM event\_time) IN (1, 2, 3) THEN 'Winter'

WHEN EXTRACT(MONTH FROM event\_time) IN (4, 5, 6) THEN 'Spring'

WHEN EXTRACT(MONTH FROM event\_time) IN (7, 8, 9) THEN 'Summer'

WHEN EXTRACT(MONTH FROM event\_time) IN (10, 11, 12) THEN 'Fall'

ELSE 'Unknown' END AS season

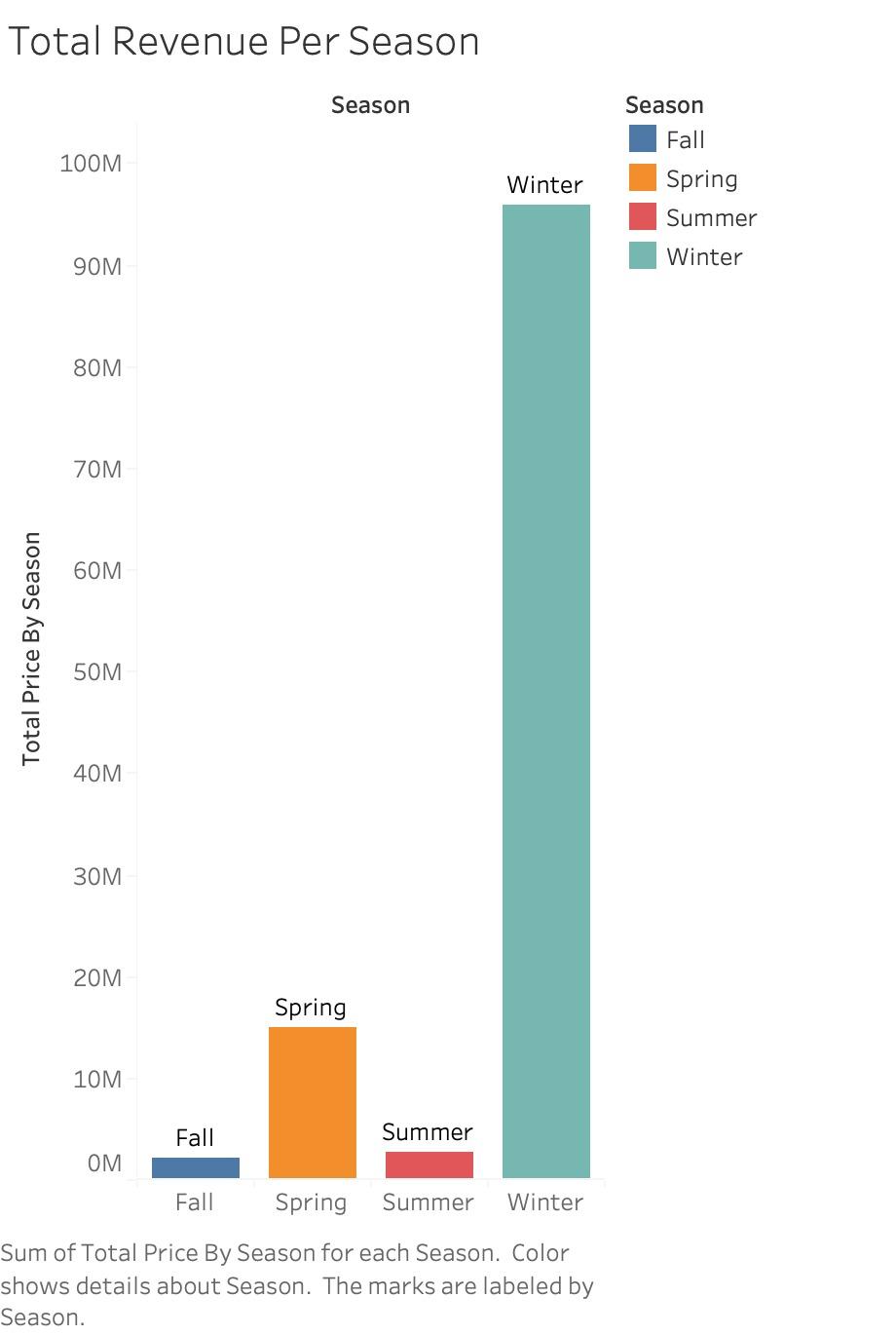
FROM SALES

GROUP BY month, season

) AS seasonal\_data

GROUP BY season

ORDER BY total\_sales\_by\_season;



# 

# Conclusions

Promotional Opportunities:

* + By identifying 8 users who made more than 60 purchases throughout the year, we have a clear opportunity to design targeted promotions and loyalty programs to further engage and reward these high-frequency customers.

Seasonal Sales Trends:

* + Our findings reveal that winter months emerge as the busiest periods, experiencing peak sales. In contrast, the fall season generates the lowest revenue. This insight can guide inventory management, marketing strategies, and promotional activities tailored to seasonal trends.

Brand Performance:

* + Samsung emerges as the top-performing brand in our store, leading both in terms of revenue and product quantity sold. Following closely are Apple and LG. Recognizing these brand dynamics allows us to optimize product placements, negotiate favorable partnerships, and cater to customer preferences effectively.