BrightLight Sales Case Study

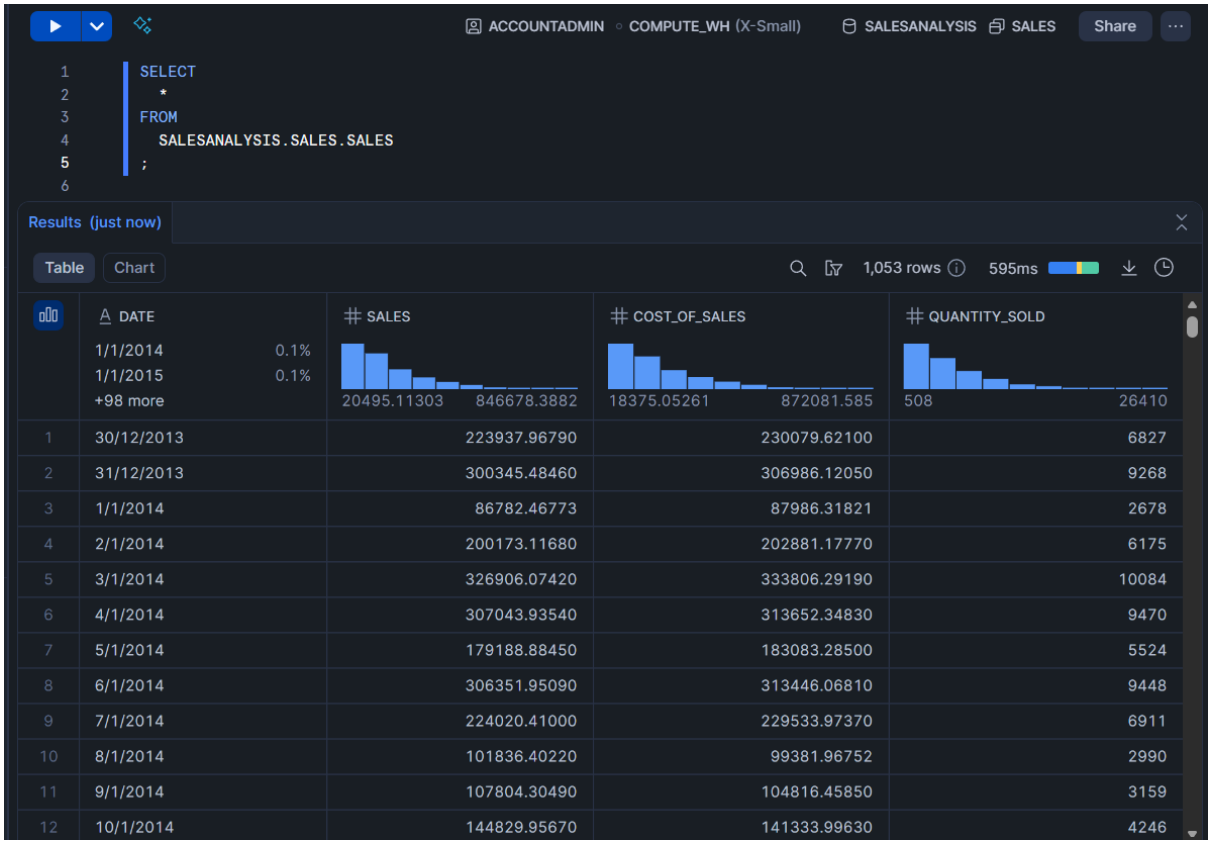
Phuti Lecia Mochueneng – Data Analyst

Introduction

This project was completed using **Snowflake SQL** and **Google Looker Studio** to analyze and visualize the *Sales Case Study* dataset. The dataset contains daily trading information for a retail store selling one specific product.
Each record includes:

- **Date** – The day the sales occurred
- **Sales** – Total Rand value of sales
- **Cost of Sales** – Total Rand value of the cost of sales
- **Quantity Sold** – Total number of units sold

The objective was to calculate several key metrics, perform deeper analysis, and derive insights through data visualizations and dashboards.



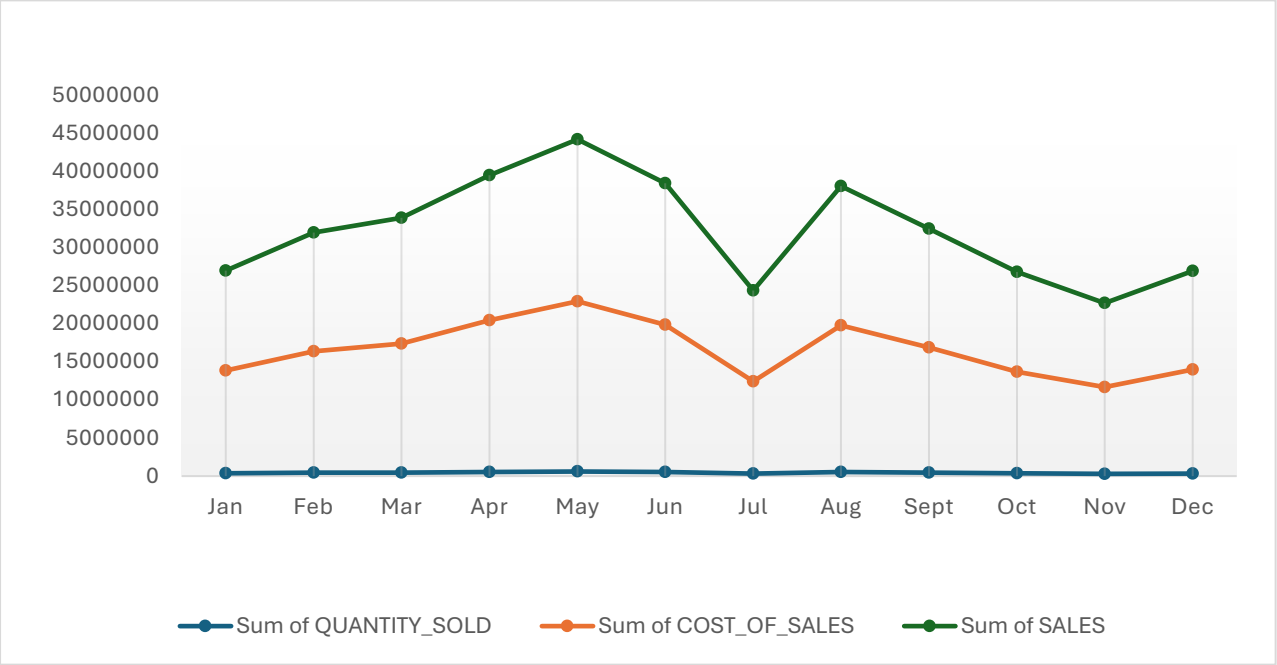
	DATE	SALES	COST_OF_SALES	QUANTITY_SOLD
	1/1/2014 0.1% 1/1/2015 0.1% +98 more	20495.11303 846678.3882	18375.05261 872081.585	508 26410
1	30/12/2013	223937.96790	230079.62100	6827
2	31/12/2013	300345.48460	306986.12050	9268
3	1/1/2014	86782.46773	87986.31821	2678
4	2/1/2014	200173.11680	202881.17770	6175
5	3/1/2014	326906.07420	333806.29190	10084
6	4/1/2014	307043.93540	313652.34830	9470
7	5/1/2014	179188.88450	183083.28500	5524
8	6/1/2014	306351.95090	313446.06810	9448
9	7/1/2014	224020.41000	229533.97370	6911
10	8/1/2014	101836.40220	99381.96752	2990
11	9/1/2014	107804.30490	104816.45850	3159
12	10/1/2014	144829.95670	141333.99630	4246

Data Preparation

After loading the dataset into Snowflake, the first step was to check:

- **Data completeness** (no missing values)
- **Data types** for each column
- **Duplicate records**
- **Date formatting** to ensure smooth time-based analysis

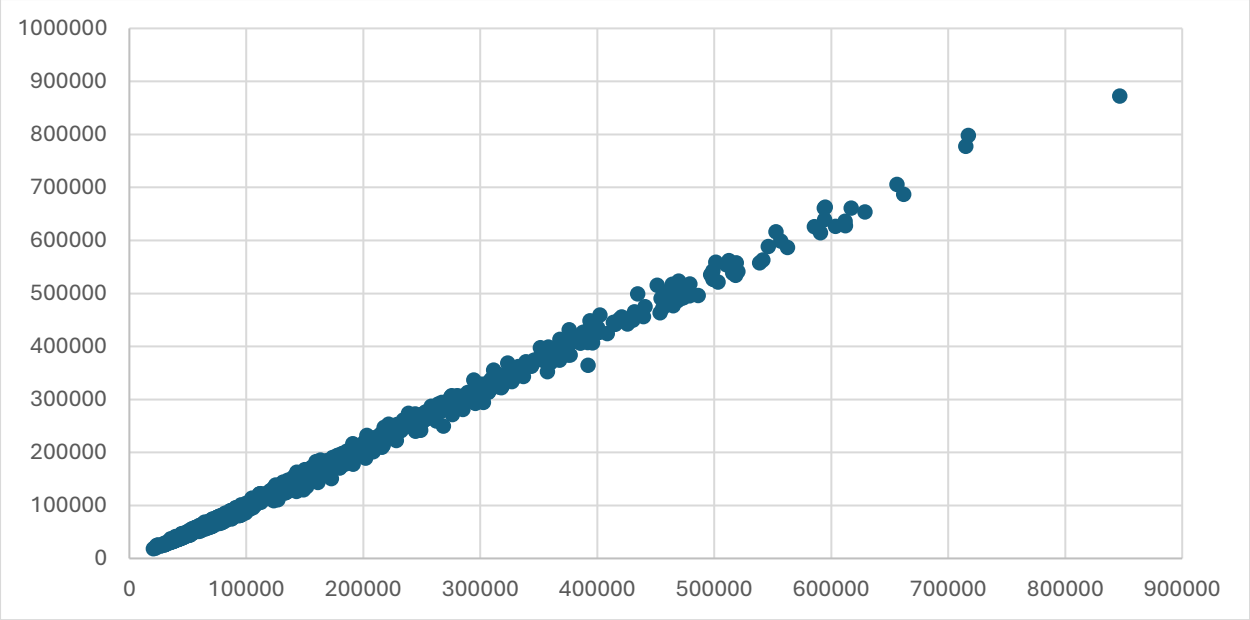
Once cleaned, the data was explored to understand the relationship between **Sales**, **Quantity Sold**, and **Cost of Sales**.



Observation:
As Sales increased, both Quantity Sold and Cost of Sales also increased, showing a direct positive relationship between these variables.

Exploratory Data Analysis (EDA)

To identify potential **outliers**, a scatter plot of *Sales vs. Cost of Sales* was created. The visual showed that most points followed a linear trend, with few extreme values indicating unusual sales days — possibly during **promotional events**.



Calculations and Metrics

Question 1: Daily Sales Price per Unit

Formula:

$$\text{Sales Price per Unit} = \frac{\text{Sales}}{\text{Quantity Sold}}$$

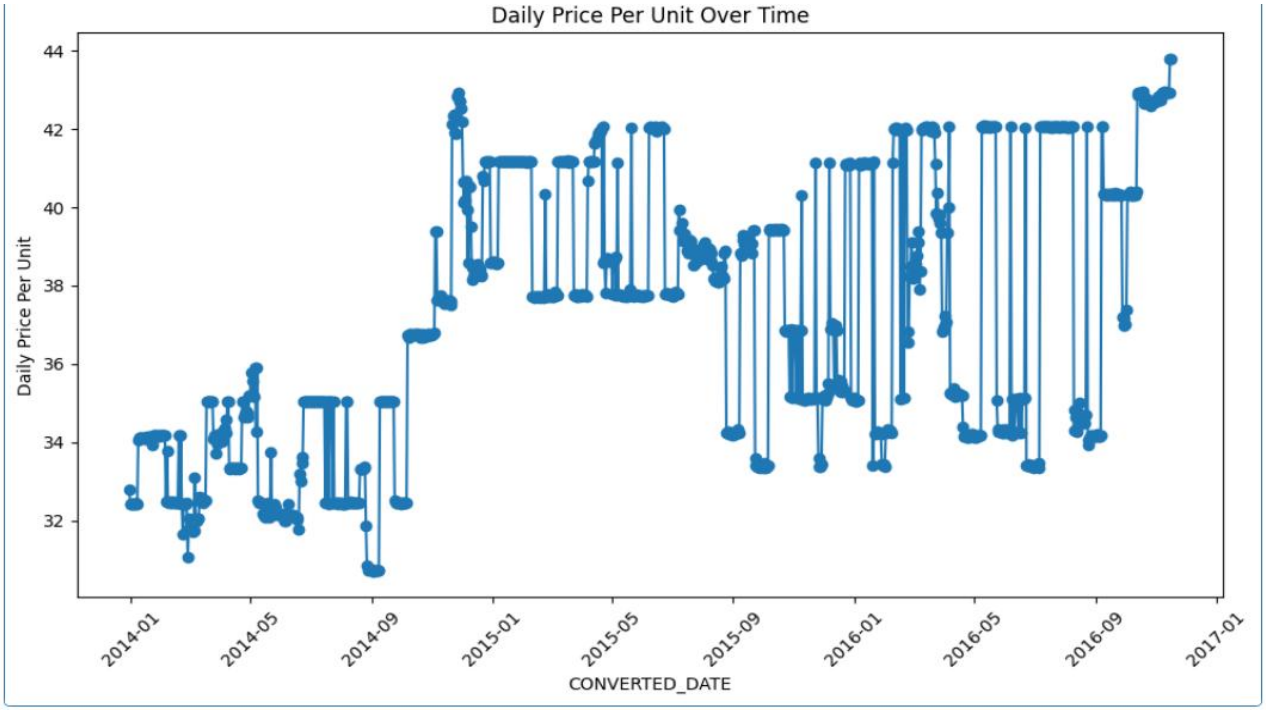
```
-- 1. What is the daily sales price per unit?
SELECT
    DATE,
    SALES,
    COST_OF_SALES,
    QUANTITY_SOLD,
    (SALES / IFNULL(QUANTITY_SOLD, 0)) AS Daily_Price_Per_Unit
FROM
    SALESANALYSIS.SALES.SALES;
```

ults (just now)

able | Chart | 1,053 rows | 27ms

DATE	SALES	COST_OF_SALES	QUANTITY_SOLD	DAILY_PRICE_PER_UNIT
30/12/2013	223937.96790	230079.62100	6827	32.80181161564
31/12/2013	300345.48460	306986.12050	9268	32.40672039275
1/1/2014	86782.46773	87986.31821	2678	32.40570116878
2/1/2014	200173.11680	202881.17770	6175	32.41669907692
3/1/2014	326906.07420	333806.29190	10084	32.41829375248
4/1/2014	307043.93540	313652.34830	9470	32.42280204857
5/1/2014	179188.88450	183083.28500	5524	32.43824846126
6/1/2014	306351.95090	313446.06810	9448	32.42505830864

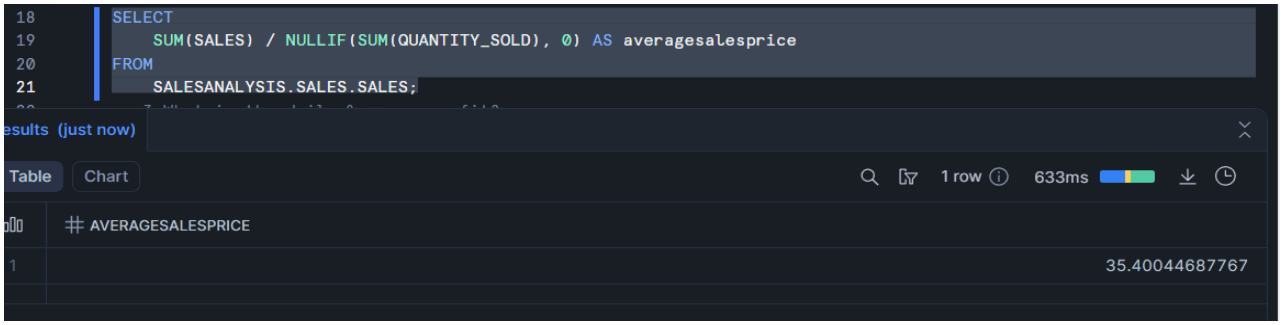
The result showed fluctuations over time, with stable pricing periods interrupted by short-term changes likely linked to promotions.



Question 2: Average Unit Sales Price

The average selling price was derived using the overall total:

$$\text{Average Unit Price} = \frac{\text{Total Sales}}{\text{Total Quantity Sold}}$$



This represents the general price level at which the product was sold throughout the observed period.

Question 3: Daily % Gross Profit

$$\text{Gross Profit} = \text{Sales} - \text{Cost of Sales}$$

$$\text{Gross Profit \%} = (\text{Gross Profit} / \text{Sales}) \times 100$$

A negative value indicates the product was sold at a loss (Cost of Sales exceeded Sales).

```sql
SELECT
    ((SALES - COST_OF_SALES) / NULLIF(SALES, 0)) * 100 AS dailygrossprofit
FROM
    SALESANALYSIS.SALES.SALES;
```

sults (just now)

Table | Chart | 1,053 rows | 464ms

DAILYGROSSPROFIT
-2.74256891700
-2.21099908000
-1.38720471000
-1.35285943700
-2.11076460300
-2.15226947600
-2.17334937400
-2.31567554200
-2.46118811200

Question 4: Daily % Gross Profit per Unit

$$\text{Gross Profit per Unit} = \frac{\text{Gross Profit}}{\text{Quantity Sold}}$$

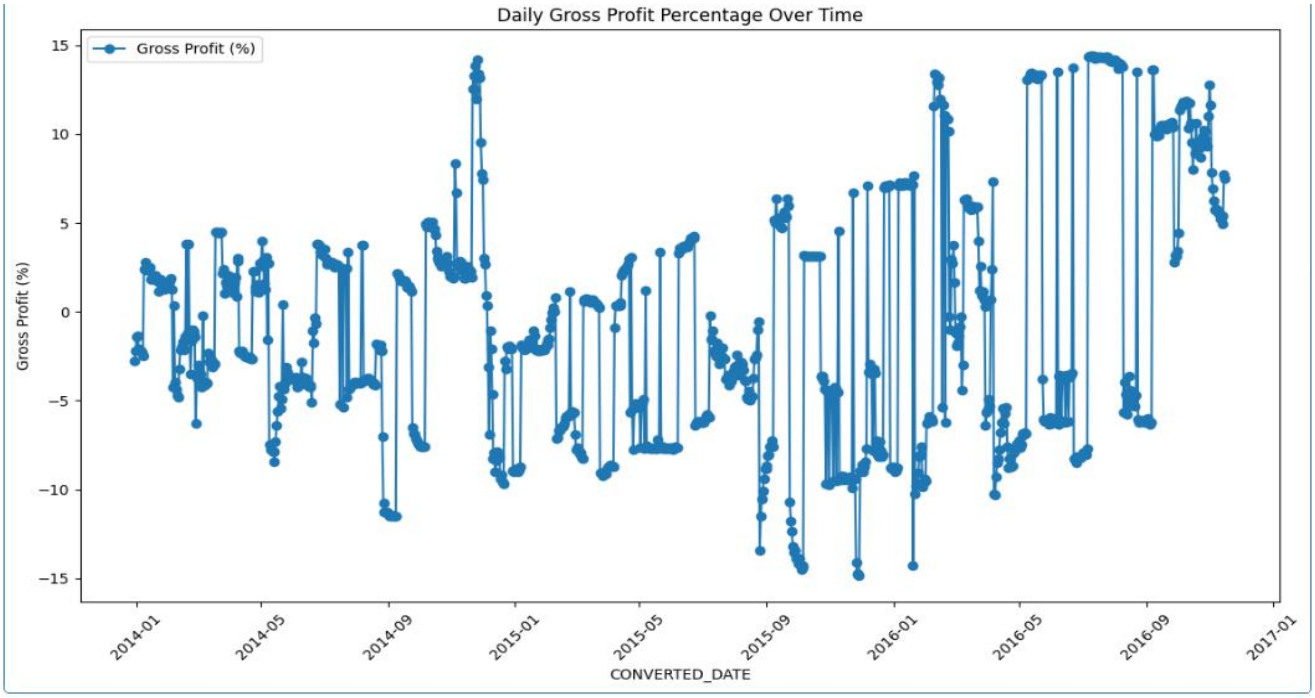
```sql
SELECT
    DATE,
    SALES,
    COST_OF_SALES,
    QUANTITY_SOLD,
    (SALES - COST_OF_SALES) / NULLIF(QUANTITY_SOLD, 0) AS Daily_Gross_Per_Unit
FROM
    SALESANALYSIS.SALES.SALES;
```

sults (just now)

Table | Chart | 1,053 rows | 98ms

DATE	SALES	COST_OF_SALES	QUANTITY_SOLD	DAILY_GROSS_PER_UNIT
30/12/2013	223937.96790	230079.62100	6827	-0.89961228944
31/12/2013	300345.48460	306986.12050	9268	-0.71651228960
1/1/2014	86782.46773	87986.31821	2678	-0.44953341299
2/1/2014	200173.11680	202881.17770	6175	-0.43855237247
3/1/2014	326906.07420	333806.29190	10084	-0.68427386950
4/1/2014	307043.93540	313652.34830	9470	-0.69782607181
5/1/2014	179188.88450	183083.28500	5524	-0.70499646995
6/1/2014	306351.95090	313446.06810	9448	-0.75085914479
7/1/2014	224020.41000	229533.97370	6911	-0.79779535523
8/1/2014	101836.40220	99381.96752	2990	0.82088116388

This helps assess the profitability per unit sold over time.



Question 5: Promotions and Price Elasticity of Demand (PED)

Based on spikes in sales from the *Date vs Sales* graph, the following promotional periods were identified:

1. **2013-12-30– 2013-12-31**
2. **2014-01-03 – 2014-01-04**
3. **2014-01-06 – 2014-01-07**

The **Price Elasticity of Demand (PED)** formula:

$$\text{PED} = \frac{\%\ \text{Change in Quantity Sold}}{\%\ \text{Change in Price}}$$

Analysis showed that the product was **price elastic**, meaning small decreases in price led to large increases in quantity sold.
This suggests that promotions boosted demand significantly.

```sql
USE DATABASE SALESANALYSIS;
USE SCHEMA SALES;
-----------------------------------------------------------------
    -- Create temp table for promo analysis (with Base_Start/Base_End)
    -----------------------------------------------------------------
    CREATE
    OR REPLACE TEMP TABLE SALESANALYSIS.SALES.PROMO_ANALYSIS (
        Promo_Name STRING,
        Base_Start DATE,
        Base_End DATE,
        Base_Price FLOAT,
        Promo_Price FLOAT,
        Base_Qty FLOAT,
        Promo_Qty FLOAT,
        Price_Change FLOAT,
        Qty_Change FLOAT,
        Price_Elasticity FLOAT
    );
-----------------------------------------------------------------
    -- Insert promo data with defined base and promo periods
    -----------------------------------------------------------------
INSERT INTO
    SALESANALYSIS.SALES.PROMO_ANALYSIS WITH promos AS (
        SELECT
            'Promo 1' AS Promo_Name,
            TO_DATE('30/12/2013', 'DD/MM/YYYY') AS Base_Start,
            TO_DATE('31/12/2013', 'DD/MM/YYYY') AS Base_End,
            TO_DATE('01/01/2014', 'DD/MM/YYYY') AS Promo_Start,
            TO_DATE('02/01/2014', 'DD/MM/YYYY') AS Promo_End
        UNION ALL
        SELECT
            'Promo 2',
            TO_DATE('03/01/2014', 'DD/MM/YYYY'),
            TO_DATE('04/01/2014', 'DD/MM/YYYY'),
            TO_DATE('05/01/2014', 'DD/MM/YYYY'),
            TO_DATE('06/01/2014', 'DD/MM/YYYY')
        UNION ALL
        SELECT
            'Promo 3',
            TO_DATE('06/01/2014', 'DD/MM/YYYY'),
            TO_DATE('07/01/2014', 'DD/MM/YYYY'),
            TO_DATE('08/01/2014', 'DD/MM/YYYY'),
            TO_DATE('08/01/2014', 'DD/MM/YYYY')
    )
```

```sql
SELECT
    p.Promo_Name,
    p.Base_Start,
    p.Base_End,
    AVG(b.SALES / NULLIF(b.QUANTITY_SOLD, 0)) AS Base_Price,
    AVG(pr.SALES / NULLIF(pr.QUANTITY_SOLD, 0)) AS Promo_Price,
    AVG(b.QUANTITY_SOLD) AS Base_Qty,
    AVG(pr.QUANTITY_SOLD) AS Promo_Qty,
    (
        (
            AVG(pr.SALES / NULLIF(pr.QUANTITY_SOLD, 0)) - AVG(b.SALES / NULLIF(b.QUANTITY_SOLD, 0))
        ) / NULLIF(AVG(b.SALES / NULLIF(b.QUANTITY_SOLD, 0)), 0)
    ) AS Price_Change,
    (
        (AVG(pr.QUANTITY_SOLD) - AVG(b.QUANTITY_SOLD)) / NULLIF(AVG(b.QUANTITY_SOLD), 0)
    ) AS Qty_Change,
    (
        (
            (AVG(pr.QUANTITY_SOLD) - AVG(b.QUANTITY_SOLD)) / NULLIF(AVG(b.QUANTITY_SOLD), 0)
        ) / NULLIF(
            (
                (
                    AVG(pr.SALES / NULLIF(pr.QUANTITY_SOLD, 0)) - AVG(b.SALES / NULLIF(b.QUANTITY_SOLD, 0))
                ) / NULLIF(AVG(b.SALES / NULLIF(b.QUANTITY_SOLD, 0)), 0)
            ),
            0
        )
    ) AS Price_Elasticity
FROM
    promos p
    LEFT JOIN SALESANALYSIS.SALES.SALES b ON TO_DATE(b.DATE, 'DD/MM/YYYY') BETWEEN p.Base_Start
    AND p.Base_End
    LEFT JOIN SALESANALYSIS.SALES.SALES pr ON TO_DATE(pr.DATE, 'DD/MM/YYYY') BETWEEN p.Promo_Start
    AND p.Promo_End
GROUP BY
    p.Promo_Name,
    p.Base_Start,
    p.Base_End
ORDER BY
    p.Promo_Name;
```

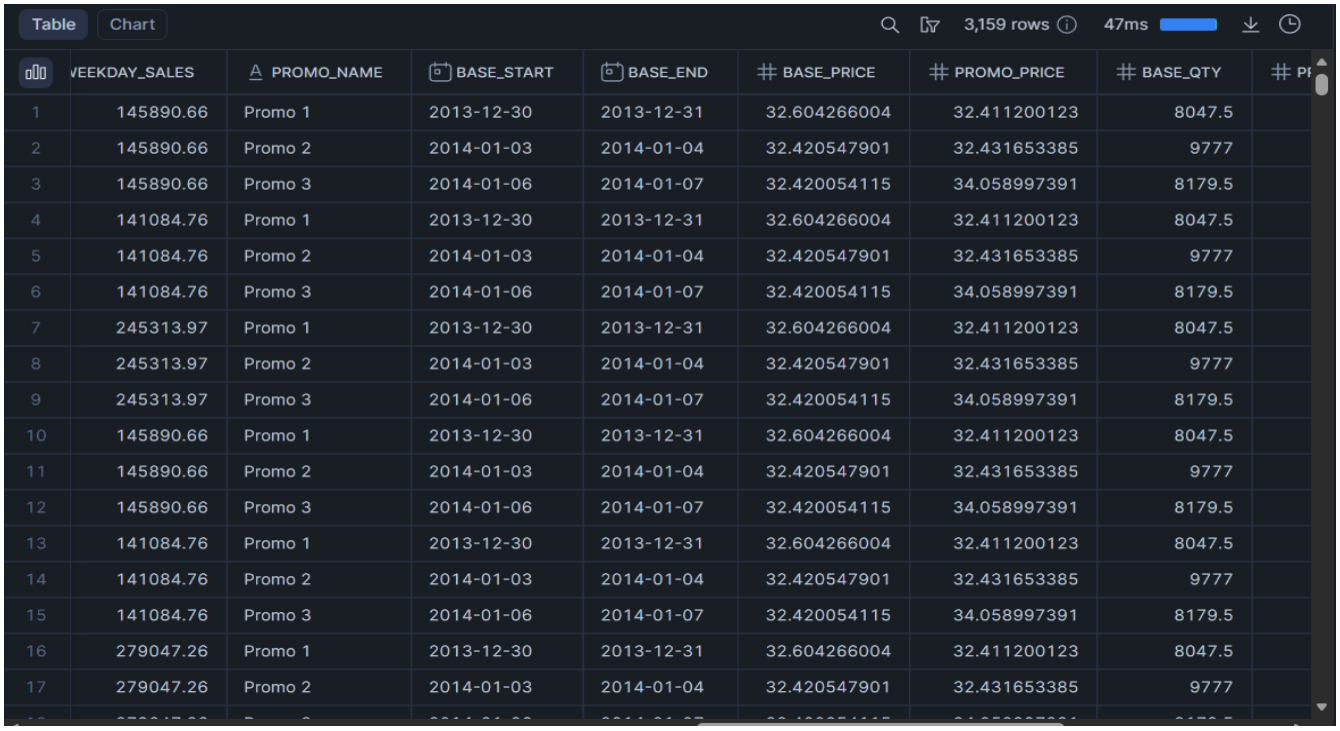


Table | Chart | 3,159 rows | 47ms

	WEEKDAY_SALES	PROMO_NAME	BASE_START	BASE_END	BASE_PRICE	PROMO_PRICE	BASE_QTY	PR
1	145890.66	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
2	145890.66	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	
3	145890.66	Promo 3	2014-01-06	2014-01-07	32.420054115	34.058997391	8179.5	
4	141084.76	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
5	141084.76	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	
6	141084.76	Promo 3	2014-01-06	2014-01-07	32.420054115	34.058997391	8179.5	
7	245313.97	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
8	245313.97	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	
9	245313.97	Promo 3	2014-01-06	2014-01-07	32.420054115	34.058997391	8179.5	
10	145890.66	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
11	145890.66	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	
12	145890.66	Promo 3	2014-01-06	2014-01-07	32.420054115	34.058997391	8179.5	
13	141084.76	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
14	141084.76	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	
15	141084.76	Promo 3	2014-01-06	2014-01-07	32.420054115	34.058997391	8179.5	
16	279047.26	Promo 1	2013-12-30	2013-12-31	32.604266004	32.411200123	8047.5	
17	279047.26	Promo 2	2014-01-03	2014-01-04	32.420547901	32.431653385	9777	

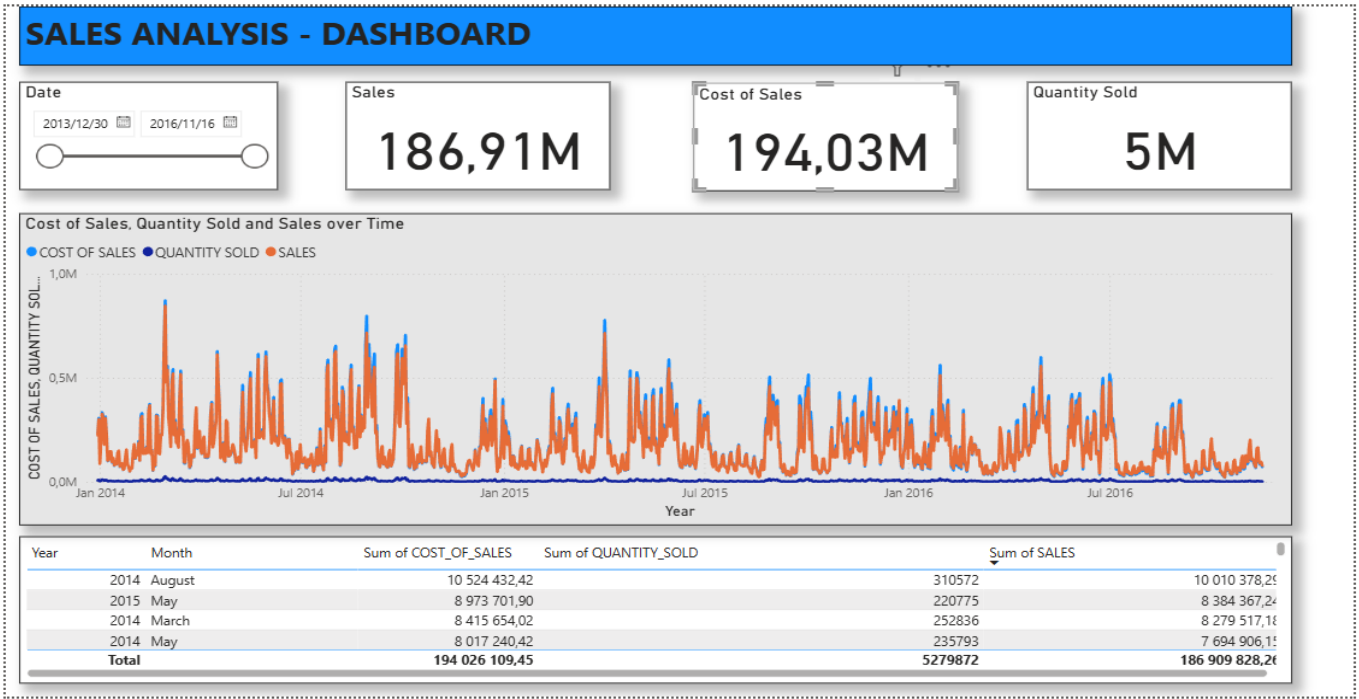
Additional Insights and KPIs

Using Power BI, additional KPIs and visuals were developed:

KPI / Metric	Description
Total Sales (R)	Overall revenue generated
Total Units Sold	Total quantity sold over time
Average Gross Profit %	Indicates profitability trends
Best Performing Days	Identifies days with peak sales
Price Elasticity Score	Highlights responsiveness to price changes

Visuals created:

- Line chart showing *Sales, Cost of Sales, and Quantity Sold* over time
- KPI cards for *Total Sales*, *Total Cost of Sales*, and *Quantity Sold*.



Conclusion

The analysis demonstrated that:

- Sales and quantity sold increase during promotional periods.
- The product is sensitive to price changes (elastic).
- Consistent profitability requires careful pricing and cost management.
- Interactive dashboards help track performance and identify sales trends in real time.