

# Coffee Shop Sales Analysis Report

by

Manmeet Singh – Data & Business Intelligence Analyst



Email: [manmeetsngh2002@gmail.com](mailto:manmeetsngh2002@gmail.com) | LinkedIn: [linkedin.com/in/manmeet-singh-182469233](https://www.linkedin.com/in/manmeet-singh-182469233) | GitHub: [github.com/Mr-Singh-oo1](https://github.com/Mr-Singh-oo1)

**Technical Stack & Tools Used:** Power BI, DAX, SQL, Excel, Data Modeling, Visualization, BI Automation

## Cover Letter

Dear Reader,

I am pleased to present my Coffee Shop Sales Analysis Report, a demonstration of my expertise in Power BI, DAX, SQL, and data analytics within a retail context. As a Fintech and Data Analyst specializing in fraud detection, BI automation, and strategic risk modelling, I have developed this report to showcase my ability to transform raw data into actionable business insights. By leveraging interactive dashboards, precise DAX calculations, and robust SQL validation, I provide stakeholders with reliable metrics to drive operational and strategic decisions.

My experience includes building data models and visualizations that enhance decision-making, as evidenced by my work on fraud detection and inventory management projects (available at [github.com/Mr-Singh-oo1](https://github.com/Mr-Singh-oo1)). This report highlights my technical proficiency and analytical mindset, making it a valuable addition to my portfolio.

Thank you for reviewing my work. I am confident it reflects my capability to deliver impactful data solutions.

Sincerely,  
Manmeet Singh

## Introduction

This Coffee Shop Sales Analysis Report showcases my expertise in Power BI, DAX, and SQL, focusing on financial and operational analytics in a retail coffee shop context. The objective was to analyse sales performance, identify trends, and validate results using multiple data tools, delivering clear and actionable insights for stakeholders.

## Problem Statement & Objective

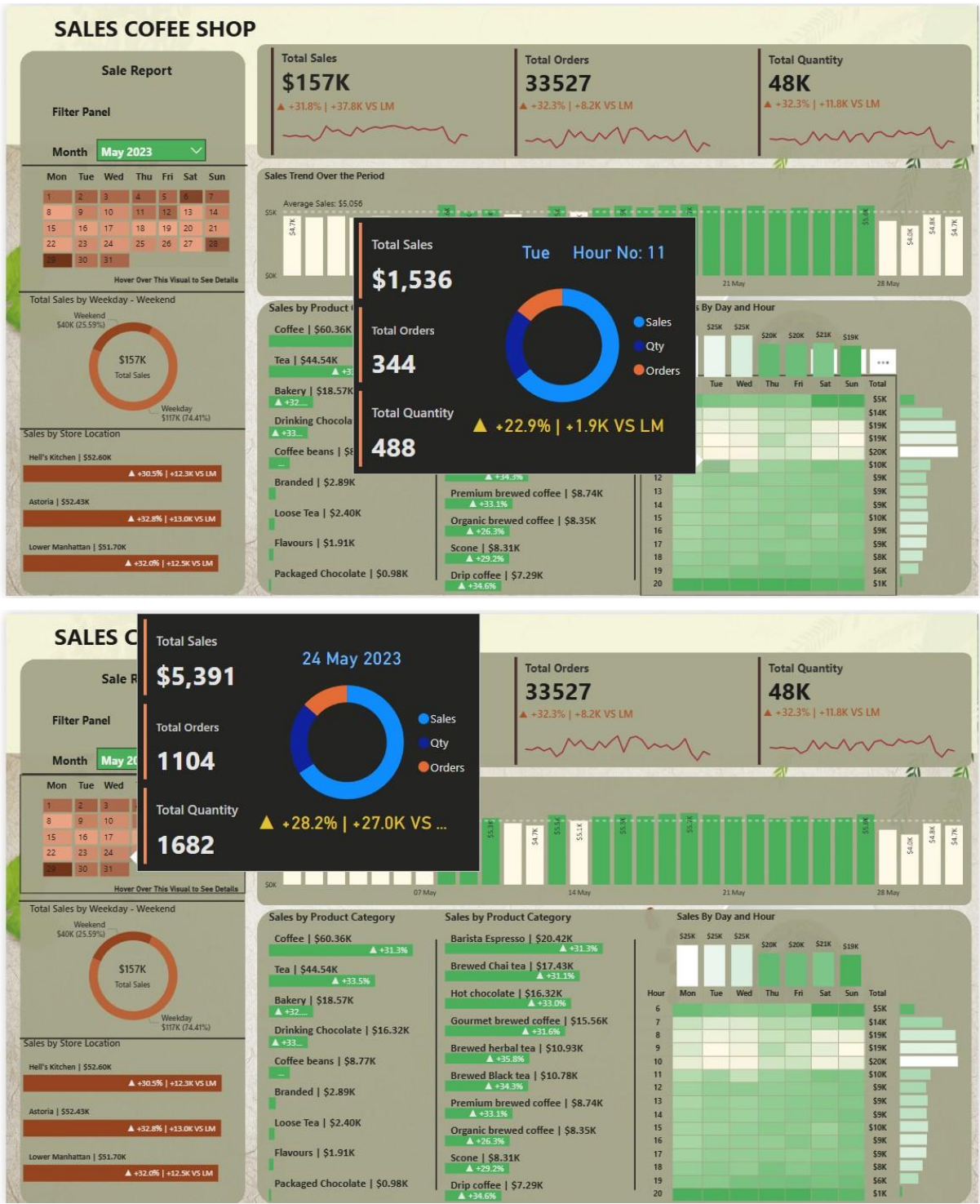
The report addresses the need to optimize coffee shop sales performance through key performance indicators (KPIs):

- Total Sales, Orders, and Quantity Sold, with month-over-month (MoM) comparisons to track growth.
- Performance analysis at store and product levels to identify strengths and weaknesses.
- Calendar heatmaps and segmentation by weekdays, weekends, and hourly trends to understand customer behaviour.

Interactive dashboards with filters for time periods and product categories enable dynamic data exploration, empowering stakeholders to pinpoint performance highlights and areas for improvement.

# Methodology Overview

The data was processed in Power BI with DAX measures for metrics computation and SQL scripts for validation. The approach ensured both analytical accuracy and dashboard interactivity for stakeholders.



## DAX Measures (Grouped)

### Basic Metrics

- **Total Sales** = SUM(Transactions[Sales])
- **Total Orders** = DISTINCTCOUNT(Transactions[transaction\_id])
- **Total Quantity** = SUM(Transactions[transaction\_qty])
- **Sales** = Transactions[unit\_price] \* Transactions[transaction\_qty]
- **Hour** = HOUR(Transactions[transaction\_time])
- **Daily Average Sales** = AVERAGEX(ALLSELECTED(Transactions[ transaction\_date]), [Total Sales])

### Current Month Metrics

- **Current Month Sales** =  
VAR selected\_month = SELECTEDVALUE('Date Table'[Month])  
RETURN  
TOTALMTD(CALCULATE([Total Sales], 'Date Table'[Month] = selected\_month), 'Date Table'[Date])
- **Current Month QTY\_Sold** =  
VAR selected\_month = SELECTEDVALUE('Date Table'[Month])  
RETURN  
TOTALMTD(CALCULATE([Total Quantity], 'Date Table'[Month] = selected\_month), 'Date Table'[Date])
- **Current Month Orders** =  
VAR selected\_month = SELECTEDVALUE('Date Table'[Month])  
RETURN  
TOTALMTD(CALCULATE([Total Orders], 'Date Table'[Month] = selected\_month), 'Date Table'[Date])

### Previous Month Metrics

- **Previous month Sales** =  
CALCULATE([Current Month Sales], DATEADD('Date Table'[Date], -1, MONTH))
- **Previous month Qty\_sold** =  
CALCULATE([Current Month QTY\_Sold], DATEADD('Date Table'[Date], -1, MONTH))
- **Previous month Orders** =  
CALCULATE([Current Month Orders], DATEADD('Date Table'[Date], -1, MONTH))

### MoM Growth and Labels

- **MOM Growth & diff Sales** =  
VAR month\_diff = [Current Month Sales] - [Previous month Sales]

```

VAR MOM = ([Current Month Sales] - [Previous month Sales]) / [ Previous
month Sales]
VAR _sign = IF(month_diff > 0, "+", "")
VAR _signtrend = IF(month_diff > 0, "\u25B2", "\u25BC")
RETURN
_signtrend & " " & _sign & FORMAT(MOM, "#0.0%") & " | " & _sign
&FORMAT(month_diff / 1000, "0.0K") & " VS LM"

```

- **MOM Growth & diff QTY\_Sold =**

```

VAR month_diff = [Current Month QTY_Sold] - [Previous month Qty_sold]
VAR MOM = ([Current Month QTY_Sold] - [Previous month Qty_sold]) /
[Previous month Qty_sold]
VAR _sign = IF(month_diff > 0, "+", "")
VAR _signtrend = IF(month_diff > 0, "\u25B2", "\u25BC")
RETURN
_signtrend & " " & _sign & FORMAT(MOM, "#0.0%") & " | " & _sign &
FORMAT(month_diff / 1000, "0.0K") & " VS LM"

```

- **MOM Growth & diff Orders =**

```

VAR month_diff = [Current Month Orders] - [Previous month Orders]
VAR MOM = ([Current Month Orders] - [Previous month Orders]) / [ Previous
month Orders]
VAR _sign = IF(month_diff > 0, "+", "")
VAR _signtrend = IF(month_diff > 0, "\u25B2", "\u25BC")
RETURN
_signtrend & " " & _sign & FORMAT(MOM, "#0.0%") & " | " & _sign &
FORMAT(month_diff / 1000, "0.0K") & " VS LM"

```

- **New MOM Label =**

```

VAR month_diff = [Current Month Sales] - [Previous month Sales]
VAR MOM = ([Current Month Sales] - [Previous month Sales]) / [ Previous month
Sales]
VAR _sign = IF(month_diff > 0, "+", "")
VAR _signtrend = IF(month_diff > 0, "\u25B2", "\u25BC")
RETURN
_signtrend & " " & _sign & FORMAT(MOM, "#0.0%")

```

## Formatting and Labels

- **Label For Store Locations =**

```

SELECTEDVALUE(Transactions[store_location]) & " | " &FORMAT([Total Sales] /
1000, "$0.00K")

```

- **Label For Product Type** =  
SELECTEDVALUE(Transactions[product\_type]) & " | " & FORMAT([Total Sales] / 1000, "\$0.00K")
- **Label For Product Category** =  
SELECTEDVALUE(Transactions[ product\_category]) & " | " & FORMAT([Total Sales] / 1000, "\$0.00K")
- **TT for Hour** = "Hour No: " & FORMAT(AVERAGE(Transactions[Hour]), 0)

### Miscellaneous

- **Colour for Bars** = IF([Total Sales] > [Daily Average Sales], "Above Average", "Below Average")
- **Placeholder** = 0

## SQL Validation Queries

### CONVERT COLUMN TO PROPER DATE FORMAT

```
UPDATE Coffee_Data
```

```
SET transaction_date = STR_TO_DATE(transaction_date, '%d-%m-%Y'),
```

```
transaction_time = STR_TO_DATE(transaction_time, '%H:%i:%s');
```

### ALTER COLUMN TO New DATA TYPE

```
ALTER TABLE coffee_Data
```

```
MODIFY COLUMN transaction_date DATE, transaction_time TIME,
```

```
CHANGE COLUMN `transaction_id` transaction_id INT;
```

### Updated Data Type by above Query

```
DESCRIBE coffee_Data;
```

	Field	Type	Null	Key	Default	Extra
►	transaction_id	int	YES		NULL	
	transaction_date	date	YES		NULL	
	transaction_time	time	YES		NULL	
	transaction_qty	int	YES		NULL	
	store_id	int	YES		NULL	
	store_location	text	YES		NULL	
	product_id	int	YES		NULL	
	unit_price	double	YES		NULL	
	product_category	text	YES		NULL	
	product_type	text	YES		NULL	
	product_detail	text	YES		NULL	

### TOTAL SALES for May Month

```
SELECT ROUND(SUM(unit_price * transaction_qty),2) as Total_Sales
```

```
FROM coffee_Data
```

```
WHERE MONTH(transaction_date) = 5
```

	Total_Sales
►	156727.76



## TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH

SELECT

```
MONTH(transaction_date) AS month,  
ROUND(SUM(unit_price * transaction_qty),2) AS total_sales,  
round( ((SUM(unit_price * transaction_qty) - LAG(SUM(unit_price * transaction_qty), 1)  
        OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(unit_price * transaction_qty), 1)  
        OVER (ORDER BY MONTH(transaction_date)) * 100),2) AS mom_increase_percentage
```

FROM

coffee\_Data

WHERE

```
MONTH(transaction_date) IN (1,2,3,4,5)
```

GROUP BY

```
MONTH(transaction_date)
```

ORDER BY

```
MONTH(transaction_date);
```

	month	total_sales	mom_increase_percentage
▶	1	81677.74	NULL
	2	76145.19	-6.77
	3	98834.68	29.8
	4	118941.08	20.34
	5	156727.76	31.77

## TOTAL ORDERS (for June)

```
SELECT COUNT(transaction_id) as Total_Orders
```

FROM coffee\_Data

```
WHERE MONTH (transaction_date)= 6
```

	Total_Orders
▶	35352

## TOTAL ORDERS KPI - MOM DIFFERENCE AND MOM GROWTH (For May and April)

SELECT

```
MONTH(transaction_date) AS month,
```



```

ROUND(COUNT(transaction_id)) AS total_orders,
round(((COUNT(transaction_id) - LAG(COUNT(transaction_id), 1)
OVER (ORDER BY MONTH(transaction_date))) / LAG(COUNT(transaction_id), 1)
OVER (ORDER BY MONTH(transaction_date)) * 100,2) AS mom_increase_percentage
FROM
    Coffee_Data
WHERE
    MONTH(transaction_date) IN (1, 2, 3, 4, 5)
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);

```

	month	total_orders	mom_increase_percentage
▶	1	17314	NULL
	2	16359	-5.52
	3	21229	29.77
	4	25335	19.34
	5	33527	32.33

### TOTAL QUANTITY SOLD (For May)

```

SELECT SUM(transaction_qty) as Total_Quantity_Sold
FROM coffee_Data
WHERE MONTH(transaction_date) = 5

```

	Total_Quantity_Sold
▶	48233

### TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH

```

SELECT
    MONTH(transaction_date) AS month,
    ROUND(SUM(transaction_qty)) AS total_quantity_sold,
    round(((SUM(transaction_qty) - LAG(SUM(transaction_qty), 1)
OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(transaction_qty), 1)

```

```

OVER (ORDER BY MONTH(transaction_date)) * 100,2) AS mom_increase_percentage
FROM
    coffee_Data
WHERE
    MONTH(transaction_date) IN (1, 2, 3, 4, 5)
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);

```

	month	total_quantity_sold	mom_increase_percentage
▶	1	24870	NULL
	2	23550	-5.31
	3	30406	29.11
	4	36469	19.94
	5	48233	32.26

### CALENDAR TABLE – DAILY SALES, QUANTITY and TOTAL ORDERS

```

SELECT
    CONCAT(ROUND(SUM(unit_price * transaction_qty) / 1000, 1),'K') AS total_sales,
    CONCAT(ROUND(COUNT(transaction_id) / 1000, 1),'K') AS total_orders,
    CONCAT(ROUND(SUM(transaction_qty) / 1000, 1),'K') AS total_quantity_sold
FROM
    coffee_Data
WHERE
    transaction_date = '2023-04-18';

```

	total_sales	total_orders	total_quantity_sold
▶	4.4K	0.9K	1.3K

### SALES TREND OVER PERIOD (for May)

```

SELECT round(AVG(total_sales),2) AS average_sales
FROM (
    SELECT

```

```

SUM(unit_price * transaction_qty) AS total_sales
FROM
    coffee_Data
WHERE
    MONTH(transaction_date) = 6
GROUP BY
    transaction_date
) AS internal_query;

```

	average_sales
▶	5549.53

### DAILY SALES FOR MONTH SELECTED (for May)

```

SELECT
    DAY(transaction_date) AS day_of_month,
    ROUND(SUM(unit_price * transaction_qty),1) AS total_sales
FROM
    coffee_Data
WHERE
    MONTH(transaction_date) = 5 -- Filter for May
GROUP BY
    DAY(transaction_date)
ORDER BY
    DAY(transaction_date);

```

	day_of_month	total_sales
▶	1	4731.4
	2	4625.5
	3	4714.6
	4	4589.7
	5	4701
	6	4205.1
	7	4542.7
	8	5604.2
	9	5101
	10	5256.3
	11	4850.1
	12	4681.1
	13	5511.5
	14	5052.6
	15	5385

	day_of_month	total_sales
	15	5385
	16	5542.1
	17	5418
	18	5583.5
	19	5657.9
	20	5519.3
	21	5370.8
	22	5541.2
	23	5242.9
	24	5391.4
	25	5230.8
	26	5300.9
	27	5559.2
	28	4338.6
	29	3959.5
	30	4835.5
	31	4684.1

#### SALES BY WEEKDAY / WEEKEND (for May)

SELECT

CASE

WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

```

        ELSE 'Weekdays'
    END AS day_type,
    ROUND(SUM(unit_price * transaction_qty),2) AS total_sales
FROM
    coffee_Data
WHERE
    MONTH(transaction_date) = 5
GROUP BY
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
        ELSE 'Weekdays'
    END;

```

	day_type	total_sales
▶	Weekdays	116627.84
	Weekends	40099.92

### SALES BY STORE LOCATION

```

SELECT
    store_location,
    round(SUM(unit_price * transaction_qty),2) as Total_Sales
FROM coffee_Data
WHERE
    MONTH(transaction_date) =5
GROUP BY store_location
ORDER BY SUM(unit_price * transaction_qty) DESC

```

	store_location	Total_Sales
▶	Hell's Kitchen	52598.93
	Astoria	52428.76
	Lower Manhattan	51700.07

### SALES BY PRODUCT CATEGORY

```

SELECT
    product_category,

```

```

ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_Data
WHERE
MONTH(transaction_date) = 6
GROUP BY product_category
ORDER BY SUM(unit_price * transaction_qty) DESC

```

	product_category	Total_Sales
▶	Coffee	64789
	Tea	46243.1
	Bakery	19251.3
	Drinking Chocolate	17106
	Coffee beans	9912.7
	Branded	3413
	Loose Tea	2770.6
	Flavours	2008
	Packaged Chocolate	992.2

### SALES BY PRODUCTS (TOP 10)

```

SELECT
    product_type,
    ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_Data
WHERE
    MONTH(transaction_date) = 6
GROUP BY product_type
ORDER BY SUM(unit_price * transaction_qty) DESC
LIMIT 10

```

	product_type	Total_Sales
▶	Barista Espresso	21860
	Brewed Chai tea	18188.2
	Gourmet brewed coffee	17142
	Hot chocolate	17106
	Brewed Black tea	11350.5
	Brewed herbal tea	11211
	Premium brewed coffee	9241.5
	Organic brewed coffee	8775
	Scone	8551.9
	Drip coffee	7770.5

## SALES BY DAY | HOUR

SELECT

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales,

SUM(transaction\_qty) AS Total\_Quantity,

COUNT(\*) AS Total\_Orders

FROM

coffee\_Data

WHERE

DAYOFWEEK(transaction\_date) = 5

AND HOUR(transaction\_time) = 12

AND MONTH(transaction\_date) = 6;

	Total_Sales	Total_Quantity	Total_Orders
▶	1750	556	381

## Key Insights & Recommendations

- Sales and orders showed steady month-over-month growth.
- Certain stores and product categories outperformed others.
- Customer demand varied by day and time, suggesting opportunities for targeted marketing and operational optimization.



## Conclusion

This project reflects my ability to design data models, create visual dashboards, and validate results using Power BI and SQL. It demonstrates how structured analytics can support data-driven decision-making in retail and fintech environments.