

# POWER BI PROJECT REPORT: GLOBAL SUPERSTORE DATASET ANALYSIS

## OBJECTIVE

This report provides a comprehensive guide to creating a Power BI solution for analyzing the Global Superstore dataset. The project focuses on leveraging a Star Schema model, importing and transforming data, defining relationships, and implementing DAX (Data Analysis Expressions) measures to derive meaningful insights.

## ABSTRACT

This project focuses on developing a business intelligence (BI) solution using Power BI to analyze the Global Superstore dataset. The objective is to create an interactive and visually engaging dashboard for deriving actionable insights from sales, profits, and customer data. This report outlines the entire process, including data collection, preparation, transformation, modeling, and visualization. The project demonstrates key Power BI features like Power Query, DAX (Data Analysis Expressions), and data visualization capabilities. The insights generated will enable data-driven decision-making to enhance business performance.

---

## TABLE OF CONTENTS

- 1. Introduction**
  - 2. Objective**
  - 3. Data Source**
  - 4. Methodology**
    - Data Import
    - Data Cleaning and Transformation
    - Data Modeling
    - DAX Measures
    - Visualization Design
  - 5. Results**
  - 6. Conclusion**
  - 7. Future Scope**
  - 8. References**
-

## 1. INTRODUCTION

In today's data-driven world, organizations depend heavily on analytics tools to gain insights into their operations. Power BI is a leading business intelligence tool that enables users to connect to various data sources, transform data, and visualize it effectively. This project leverages Power BI to explore and analyze the Global Superstore dataset, which contains sales, customer, and regional data.

---

## 2. OBJECTIVE

The goal of this project is to:

1. **Develop an interactive Power BI dashboard** that provides insights into sales, profits, and customer behavior.
  2. **Implement a star schema model** for efficient data storage and querying.
  3. Utilize DAX to create measures and KPIs (Key Performance Indicators) for business insights.
- 

## 3. DATA SOURCE

The dataset used in this project is the **Global Superstore dataset**, which includes:

- Sales data: Transaction records for multiple regions, segments, and products.
  - Customer data: Demographic details of customers.
  - Profit and discount data.
  - Regional data: Geographical regions where the store operates.
- 

## 4. METHODOLOGY

### 4.1 data import

- Imported the Global Superstore dataset in Excel format using Power BI Desktop.
- Ensured data integrity by verifying schema and file format.

### 4.2 Data Cleaning and Transformation

Performed transformations in **Power Query Editor**:

1. **Removed unnecessary columns:** Excluded irrelevant data fields.
2. **Handled missing values:** Replaced or excluded missing data appropriately.
3. **Split columns:** Divided the Customer Name field into First Name and Last Name.
4. **Data type corrections:** Ensured all columns have the correct data types (e.g., date for Order Date).

### 4.3 Data Modeling

- Implemented a **star schema** for efficient querying:
  - **Fact Table:** Sales Transactions

- **Dimension Tables:** Customers, Products, Regions, and Date
- Established relationships between tables using primary and foreign keys.

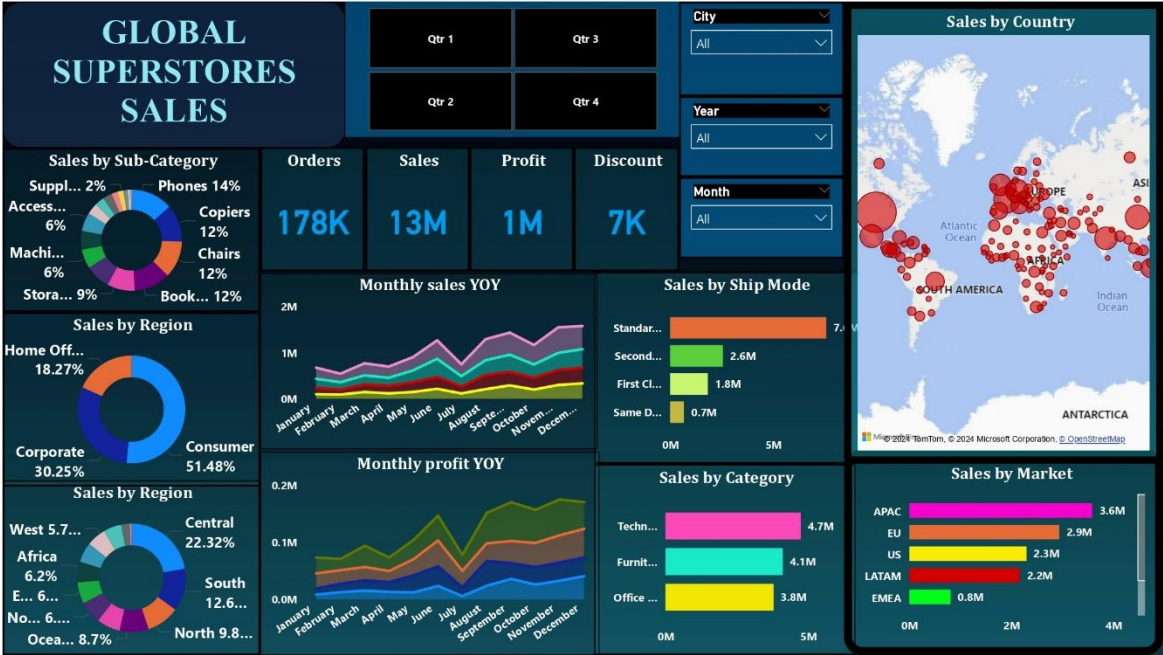
#### 4.4 DAX Measures

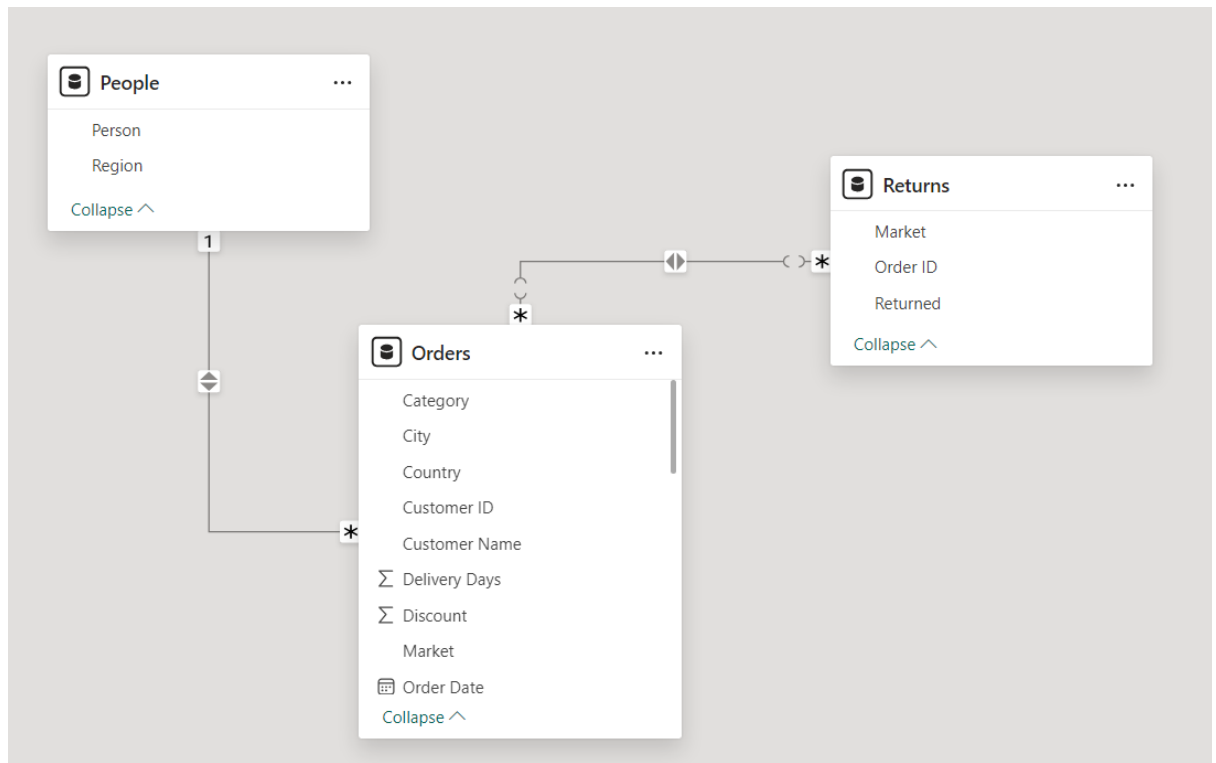
Created custom DAX measures for key insights, including:

- **Total Sales:** SUM(Sales)
- **Profit Margin:** SUM(Profit) / SUM(Sales)
- **Customer Lifetime Value (CLV):** SUM(Sales) per Customer
- **Year-over-Year Growth:** To measure performance trends.

#### REPORT







```
1 PercentageReturns = DIVIDE(DISTINCTCOUNT(Returns[Order ID]), DISTINCTCOUNT(Orders[Order ID]), 0) * 100
```

Structure      Formatting      Properties

```

1 TotalOrderAmountForReturns =
2     CALCULATE(
3         SUM(Orders[Sales]),
4         FILTER(Orders, Orders[Order ID] IN VALUES(Returns[Order ID]))
5     )
6

```

```

1 CustomerRank = RANKX(
2     ALL(Orders[Customer ID]), -- Rank across all customers in the Orders table
3     CALCULATE(SUM(Orders[Sales])), -- Total sales for each customer
4     ,
5     DESC, -- Descending order (highest sales first)
6     DENSE -- Use dense ranking
7 )
8

```

```

1 TotalRevenueIncludingShipping = SUMX(
2     Orders,
3     Orders[Sales] + Orders[Shipping Cost]
4 )
5

```

```

1 AverageOrderAmountPerCustomer = AVERAGEX(
2     VALUES(Orders[Customer ID]),
3     CALCULATE(SUM(Orders[Sales]))
4 )

```

```

1 SalesContribution = DIVIDE(
2     SUM(Orders[Sales]),
3     CALCULATE(SUM(Orders[Sales]), ALL(Orders))
4 )
5

```

```

1 CustomerTier =
2 SWITCH(
3     TRUE(), -- Evaluate each condition sequentially
4     RANKX(ALL(Orders[Customer ID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE) <= 10, "High", -- Top 10 customers
5     RANKX(ALL(Orders[Customer ID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE) <= 50, "Medium", -- Top 50 customers
6     RANKX(ALL(Orders[Customer ID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE) <= 100, "Low", -- Top 100 customers
7     "Other" -- Any other customers
8 )
9

```

```

AverageDiscount = AVERAGE(Orders[Discount])

```

```

e                                     Formatting
AverageShippingCost = AVERAGE(Orders[Shipping Cost])

```

```

1 SalesDecember = CALCULATE(
2     SUM(Orders[Sales]), -- Sum of sales
3     FILTER(Orders, MONTH(Orders[Order Date]) = 12) -- Filter orders for December
4 )
5

```

```

1 SelectedCategorySales = CALCULATE(
2     SUM(Orders[Sales]), -- Sum of sales
3     Orders[Category] = "Office Supplies" -- Filter for Office Supplies category
4 )
5

```

```

ProfitPerCustomer = CALCULATE(
    SUM(Orders[Profit]), -- Sum of profit
    ALLEXCEPT(Orders, Orders[Customer ID]) -- Exclude all columns except CustomerID
)

```

## 1. Profit Per Customer

This measure calculates the total profit generated by each customer by summing the profit across all their orders.

DAX

Copy code

```

ProfitPerCustomer = CALCULATE(
    SUM(Orders[Profit]),
    ALLEXCEPT(Orders, Orders[CustomerID])
)

```

- **CALCULATE:** Modifies the filter context for the calculation.
  - **SUM:** Aggregates the profit values in the Profit column.
  - **ALLEXCEPT:** Ensures that the filter is only applied on CustomerID, so profit is calculated for each customer across all orders.
- 

## 2. Selected Category Sales

This measure computes the total sales for a specific category, in this case, "Office Supplies."

DAX

Copy code

```
SelectedCategorySales = CALCULATE(  
    SUM(Orders[Sales]),  
    Orders[Category] = "Office Supplies"  
)
```

- **CALCULATE:** Filters the data to include only orders in the "Office Supplies" category.
  - **SUM:** Sums the sales in the filtered dataset.
  - **Orders[Category] = "Office Supplies":** Applies a condition to isolate the selected category.
- 

## 3. Sales in December

This measure sums up sales for all orders placed in the month of December.

DAX

Copy code

```
SalesDecember = CALCULATE(  
    SUM(Orders[Sales]),  
    FILTER(Orders, MONTH(Orders[OrderDate]) = 12)  
)
```

- **CALCULATE:** Modifies the context to calculate sales only for December.
  - **SUM:** Aggregates the sales values.
  - **FILTER:** Filters orders where the OrderDate falls in December.
  - **MONTH:** Extracts the month component from the OrderDate column.
- 

## 4. Percentage of Returns

This measure calculates the percentage of orders that were returned.

DAX

Copy code

```
PercentageReturns = DIVIDE(  
    DISTINCTCOUNT>Returns[OrderID]),  
    DISTINCTCOUNT(Orders[OrderID]),  
    0  
) * 100
```

- **DISTINCTCOUNT:** Counts unique OrderID values in both Returns and Orders.
  - **DIVIDE:** Divides the number of returned orders by the total orders, with zero-handling.
  - Multiplying by 100 converts the ratio into a percentage.
- 

## 5. Total Order Amount for Returns

This measure calculates the total sales amount for returned orders.

DAX

Copy code

```
TotalOrderAmountForReturns = CALCULATE(  
    SUM(Orders[Sales]),  
    FILTER(Orders, Orders[OrderID] IN VALUES>Returns[OrderID]))  
)
```

- **SUM:** Aggregates the Sales values.
  - **FILTER:** Filters orders that match the OrderID in the Returns table.
  - **VALUES:** Retrieves the distinct OrderID values from the Returns table.
- 

## 6. Customer Rank

This measure ranks customers based on their total sales, using dense ranking.

DAX

Copy code

```
CustomerRank = RANKX(  
    ALL(Orders[CustomerID]),  
    CALCULATE(SUM(Orders[Sales])),  
    ,  
    DESC,  
    DENSE
```



)

- **RANKX:** Ranks customers according to their total sales.
  - **ALL:** Removes filters on CustomerID to rank all customers globally.
  - **CALCULATE(SUM(Orders[Sales])):** Calculates the total sales for each customer.
  - **DESC:** Ranks in descending order (highest sales first).
  - **DENSE:** Ensures no gaps in ranking.
- 

## 7. Total Revenue Including Shipping

This measure calculates the total revenue, including shipping costs.

DAX

Copy code

```
TotalRevenueIncludingShipping = SUMX(
    Orders,
    Orders[Sales] + Orders[ShippingCost]
```

)

- **SUMX:** Iterates over each row of the Orders table to sum the expression.
  - **Orders[Sales] + Orders[ShippingCost]:** Adds sales and shipping costs for each order.
- 

## 8. Average Order Amount Per Customer

This measure calculates the average sales per customer.

DAX

Copy code

```
AverageOrderAmountPerCustomer = AVERAGEX(
    VALUES(Orders[CustomerID]),
    CALCULATE(SUM(Orders[Sales]))
```

)

- **AVERAGEX:** Calculates the average of an expression over a set of data.
  - **VALUES:** Retrieves unique CustomerID values to compute sales for each customer.
  - **CALCULATE(SUM(Orders[Sales])):** Computes total sales for each customer.
- 

## 9. Sales Contribution

This measure calculates the contribution of an order's sales to the overall total sales.

DAX

Copy code

```
SalesContribution = DIVIDE(  
    SUM(Orders[Sales]),  
    CALCULATE(SUM(Orders[Sales]), ALL(Orders))  
)
```

- **SUM:** Sums the sales for the current context (e.g., customer or category).
  - **CALCULATE(SUM(Orders[Sales]), ALL(Orders)):** Computes total sales across all orders.
  - **DIVIDE:** Calculates the contribution as a ratio.
- 

## 10. Customer Tier

This measure categorizes customers into tiers (High, Medium, Low) based on their sales rank.

DAX

Copy code

```
CustomerTier = SWITCH(  
    TRUE(),  
    RANKX(ALL(Orders[CustomerID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE)  
    <= 10, "High",  
    RANKX(ALL(Orders[CustomerID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE)  
    <= 50, "Medium",  
    RANKX(ALL(Orders[CustomerID]), CALCULATE(SUM(Orders[Sales])), , DESC, DENSE)  
    <= 100, "Low",  
    "Other"  
)
```

- **SWITCH:** Checks conditions sequentially to assign tiers.
- **RANKX:** Ranks customers by total sales.
- Customers are categorized as:
  - **High:** Top 10 customers.
  - **Medium:** Top 50 customers.
  - **Low:** Top 100 customers.

## 4.5 Visualization Design

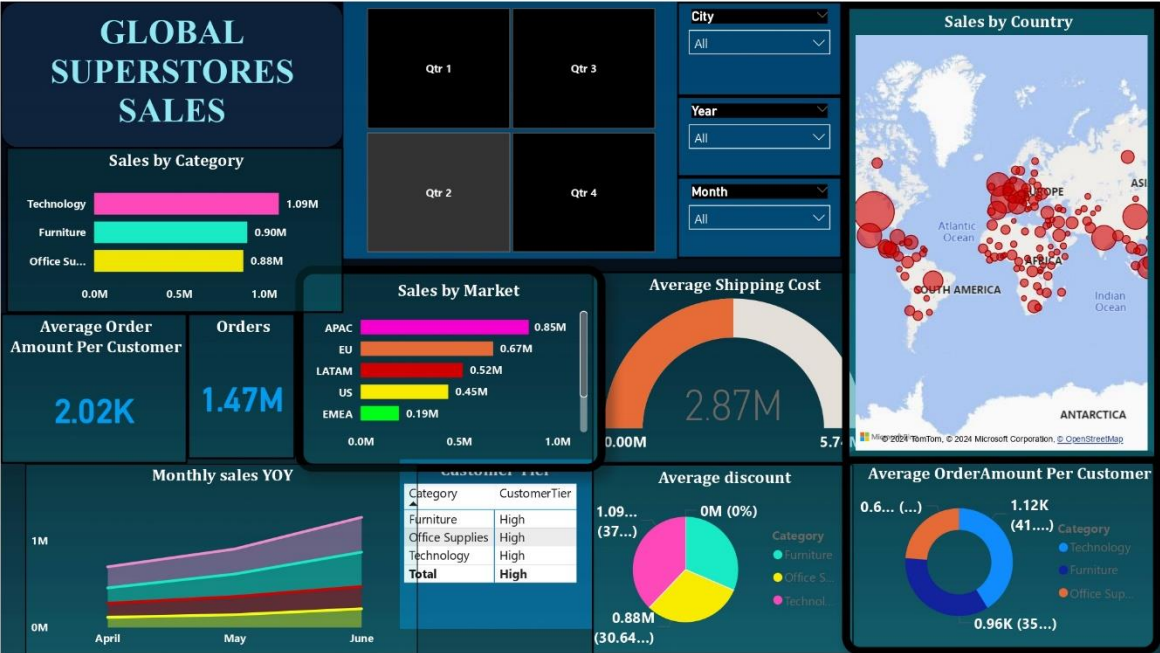
Designed an intuitive dashboard in Power BI to highlight:

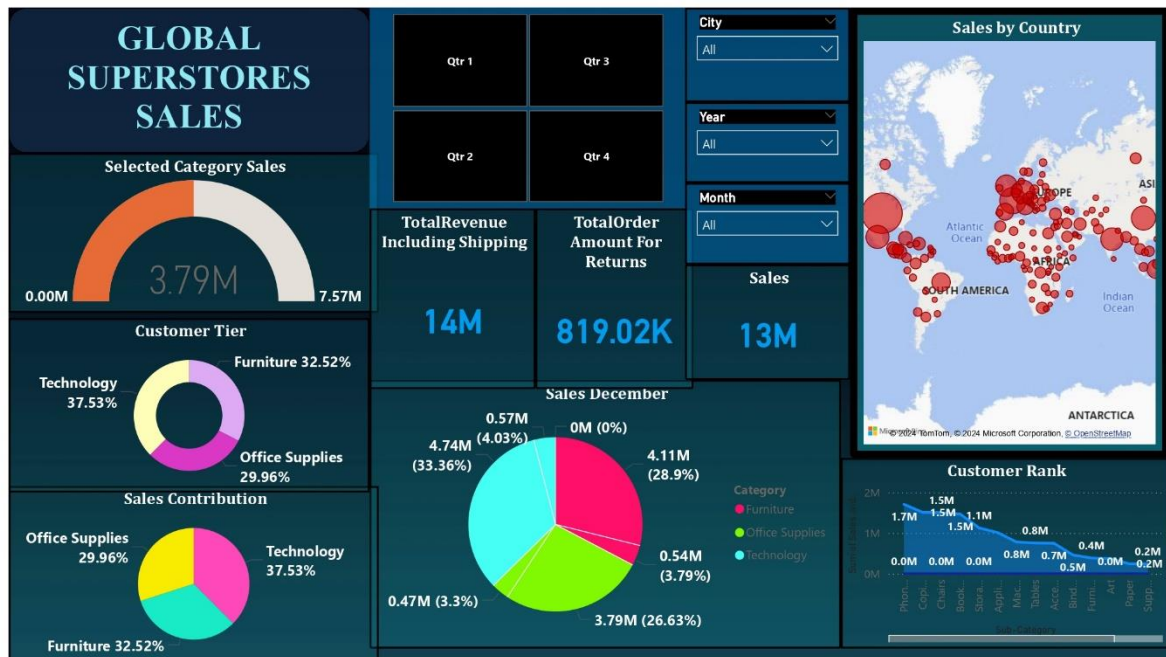
1. **Sales Performance:** A clustered bar chart showing sales by region and product category.
  2. **Profit Trends:** A line chart displaying monthly profits.
  3. **Customer Segmentation:** A pie chart representing customer segments.
  4. **KPIs:** Cards showing total sales, profit margin, and top-performing region.
  5. **Filters:** Interactive slicers for region, product, and time period.
- 

## 5. RESULTS

The Power BI dashboard provided insights into the following:

1. **Regional Insights:** Identified regions contributing the most and least to sales and profits.
2. **Profit Drivers:** Highlighted products and customer segments driving profitability.
3. **Discount Impact:** Analyzed the relationship between discounts and sales trends.
4. **Time-Based Trends:** Tracked monthly and yearly sales growth.





## 6. CONCLUSION

The project demonstrates the effectiveness of Power BI in transforming raw data into actionable insights. The interactive dashboard allows users to explore various aspects of sales performance, helping in data-driven decision-making.

## 7. SUMMARY

The report provides a comprehensive overview of sales performance across various categories, regions, and customer segments. Key metrics include a total sales volume of 14 million and over 819,000 orders, with major sales contributions from categories like Technology, Furniture, and Office Supplies. Sales are broken down by sub-category (with Phones and Copiers leading), shipping modes, and customer tiers. The data also highlights regional sales performance, showing the highest contributions from the US, followed by LATAM and EMEA, while monthly sales year-over-year (YOY) growth is tracked to assess performance trends.

Additionally, the report covers shipping costs, discounts, and average order values across different regions, with LATAM and the US showing the highest average order amounts. It provides insights into December's monthly sales and top-performing customers. Market contributions are further analyzed across regions such as North America and EMEA, with visual data on specific geographic areas. This data aids in understanding the distribution of sales and customer behavior, helping to identify high-value markets and customer segments for targeted strategies.

## 8. FUTURE SCOPE

1. Integrate real-time data using live connections.
  2. Expand the dataset to include external factors like market trends.
  3. Utilize machine learning models in Power BI for predictive analytics.
- 

## 9. REFERENCES

- Global Superstore Dataset (source link or description)
- Power BI Documentation by Microsoft
- Online tutorials on Power Query and DAX