# **Amazon E-Commerce Project Documentation**

**Objective Questions:**

1. **What is the total number of attributes in the customer table?**

* **Guidelines**: Total Number of Attributes in the Customer Table
  + The customer table has 3 attributes:
* CustomerID: a unique identifier for each customer
* Customer Age: the age of each customer
* Customer Gender: the gender of each customer

1. **How will you get the “Customer’s” ages in the “Order” tables according to customer IDs?**

* **Guidelines:** To Get Customer Ages in the Order Table using Excel:  
  In a new column in the Orders table, enter the following formula:

| **=INDEX(Customers!B:B,MATCH(D2,Customers!A:A,0))** |
| --- |

Assuming:

* Customers!B:B is the range of cells containing the customer ages.
* D2 is the cell containing the customer ID in the Orders table.
* Customers!A:A is the range of cells containing the customer IDs in the Customers table.

**What insights can be derived from this?**

* Age Distribution: By analyzing the age distribution of our customers, we can identify trends and patterns in their behavior. For instance, we might find that most of our customers are between 25-45 years old.
* Demographic Segmentation: We can segment our customers based on their age and analyze their purchasing behavior, preferences, and needs. This can help us tailor our marketing strategies and product offerings to specific age groups.
* Product Recommendations: Based on the age distribution of our customers, we can recommend products that are more likely to appeal to specific age groups. For instance, younger customers may be more interested in trendy or fashion-forward products.
* Marketing Strategies: By understanding the age distribution of our customers, we can develop targeted marketing strategies that resonate with specific age groups. For example, social media campaigns may be more effective for younger customers.

1. **In analyzing the dataset with Power BI, ensure data cleaning to address inconsistencies and missing values before further analysis.**

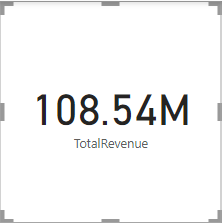
* **Guidelines:** For Cleaning and Preparing Data in Power BI

1. Load Data:
   * Import your dataset into Power BI and verify all the data has been loaded correctly.
2. Open Power Query Editor:
   * Navigate to Transform Data to access Power Query Editor, where most data cleaning happens.
3. Handle Missing Values:
   * Identify columns with missing values:
     + Use Transform > Count Rows or Transform > Replace Errors to locate them.
   * Replace or impute missing values:
     + Use Replace Values for categorical columns (e.g., fill Reason with "No Reason Provided").
     + Use Fill Down for sequential data where gaps occur.
4. Remove Duplicates:
   * Check for duplicate rows using Remove Duplicates in the toolbar.
   * Verify by selecting all columns or only unique identifiers like OrderID or CustomerID.
5. Correct Data Types:
   * Ensure all columns have appropriate data types:
     + Date columns (e.g., OrderDate, Delivery Date) as Date/Time.
     + Numerical columns (e.g., Unit Price, Shipping Fee) as Decimal Number.
     + Categorical columns (e.g., Product Category) as Text.
6. Resolve Inconsistencies:
   * Standardize text data using Transform > Format > Trim, Clean, or Uppercase/Lowercase.
   * Use Group By to spot outliers or inconsistent category names in columns like Location or Zone.
7. Filter Out Irrelevant Data:
   * Exclude rows with invalid or unnecessary information (e.g., outliers, inactive customers).
8. Validation:
   * Cross-check data against source files to confirm accuracy after cleaning.
   * Run summary statistics to detect anomalies.
9. **How can we calculate the total revenue generated by all the sales?**

* **Guidelines:** For Calculating Total Revenue in Power BI

Preparation Steps:

1. **Ensure Data Accuracy**:
   * Verify the dataset contains all relevant columns: Sale Price and Shipping Fee.
2. **Load and Understand Data**:
   * Import the dataset into Power BI.
   * Review the data model to ensure relationships are properly set if working with multiple tables (e.g., Orders and Customers tables).

**Which could look like this:**

Implementation Steps:

Step 1: Create a Measure for Total Revenue

Use the DAX formula:

| TotalRevenue = SUM(Orders[Sale Price]) + SUM(Orders[Shipping Fee]) |
| --- |

This formula adds the total values of Sale Price and Shipping Fee.

Step 2: Visualize Total Revenue

Go to the **Report View** in Power BI.

Add a **Card Visual**:

* + From the Visualizations pane, select the **Card** visual.
  + Drag the TotalRevenue measure to the card to display the total revenue amount.

1. **What is the total number of unique customers who made purchases each year? Is there any increase in the number over the years?**

* **Guidelines:** Total number of unique customers for each years

Step-by-Step Answer:

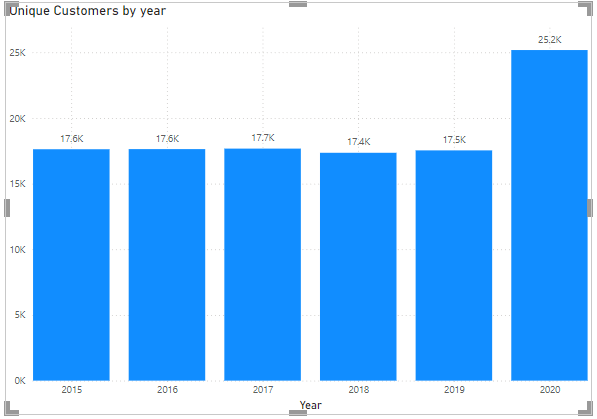
* Calculation of Unique Customers Per Year:

For each year, we calculate the total number of unique customers who made purchases. This can be done using the DAX formula for counting distinct customer IDs within the Orders dataset, broken down by the year.

* DAX Formula:

| Unique Customers by year = CALCULATE(DISTINCTCOUNT(Customers[CustomerID]),FILTER(ALL(Orders),YEAR(Orders[OrderDate])=MAX(Orders[year]))) |
| --- |

* Trend Analysis:
  + Once you have the number of unique customers for each year, you can plot these values using a bar chart or line chart to visualize the trend over time.
  + Observe if the number of unique customers has increased, decreased, or remained consistent over the years.

Which could look like this:   


**Insights:**

Increase in Unique Customers: If there is a consistent increase in unique customers year over year, it could indicate growth in the customer base, perhaps due to better marketing strategies, an increase in product variety, or other positive factors.

Fluctuations: If the numbers fluctuate significantly, this could suggest seasonality in purchases, promotions, or external factors like economic conditions.

1. **How can we determine the total number of unique products available in the company?**

* **Guidelines:** For Determining the Total Number of Unique Products Available in the Company:

Data Preparation:

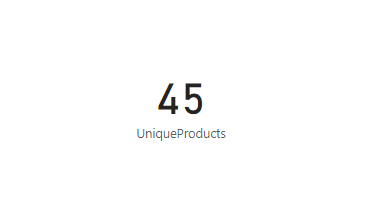
* Identify the relevant column in the Orders table that represents product data:
  + Typically, this will be a column like Product, ProductName, or ProductID. If you're using product names, ensure the column consistently lists unique product names without duplicates or formatting issues.

DAX Calculation:

* Using DISTINCTCOUNT: Since we don't have a separate Products table, we can still use the Orders table to count unique products.

Here's the DAX formula to calculate the number of unique products in the Orders table:

| UniqueProducts = DISTINCTCOUNT(Orders[Product]) |
| --- |

**Which could look like this:** 

**Insights:**

1. **Product Catalog Size**:
   * The total number of unique products tells you how broad your product offering is. A higher number of unique products generally indicates a larger and more diverse catalog.
2. **Catalog Growth**:
   * If the number of unique products increases over time, it could indicate that the company is expanding its product range. This may be driven by demand, new trends, or strategic decisions such as diversifying the product portfolio.
   * Conversely, a stagnant or decreasing number of unique products could point to a narrowing product focus, fewer new product launches, or discontinued products.
3. **Product Availability and Variety**:
   * A larger number of unique products usually correlates with greater variety, which can attract a wider range of customers. This might improve the chances of meeting diverse customer needs.
4. **Market Positioning**:
   * If your company has a large number of unique products, it might be seen as a one-stop shop in its industry. Conversely, if the number of unique products is low, this could imply a more specialized or niche market.
5. **Inventory and Supply Chain Insights**:
   * A growing number of unique products could also indicate that the company is managing a more complex inventory system. This could lead to opportunities for better inventory tracking and demand forecasting.
6. **Out-of-Stock Issues**:
   * If you're tracking product availability, a decrease in the number of unique products over time could signal out-of-stock issues or product discontinuation. This can help prioritize restocking efforts or replacements.
7. **What is the average number of days it takes for products to be delivered, get the metric for only the delivered orders.**

* **Guidelines:** For Calculating Average Delivery Days for Delivered Orders in Power BI

Preparation:

1. **Verify Dataset Completeness**:
   * Ensure your dataset contains the **Order Date** and **Delivery Date** columns, as well as the **Status** column to filter for delivered orders.
   * Confirm that the **Delivery Date** is populated for all delivered orders, and missing values are handled if needed.
2. **Data Cleaning**:
   * If there are any missing or invalid delivery dates, you should either impute them (e.g., using the order date) or exclude those records to ensure accurate calculations.
   * Clean the **Status** column to ensure it is consistent (e.g., “Delivered” should be the exact match for all delivered orders).

**Steps in Power BI:**

* + Create a New Measure for Average Delivery Days:
  + Go to Model View or Data View.
  + Click on New Measure.
  + Use the following DAX formula to calculate the average number of days it took for orders to be delivered:

| AverageDeliveryDays = AVERAGEX(FILTER(Orders,Orders[Status]="Delivered" && NOT(ISBLANK(Orders[Delivery Date]))),DATEDIFF(Orders[OrderDate],Orders[Delivery Date],DAY)) |
| --- |

**Insights:**

* **Operational Efficiency**:
  + The average delivery time is an essential metric for understanding the efficiency of your logistics and order fulfillment processes.
  + A decrease in the average delivery days over time indicates an improvement in shipping operations, while an increase suggests inefficiencies that should be addressed.
* **Customer Satisfaction**:
  + Shorter delivery times are generally linked to higher customer satisfaction. If you see a trend toward faster delivery times, this could suggest improvements in customer service and operational capabilities.
* **Impact of Shipping Methods**:
  + By comparing average delivery days across different delivery methods (e.g., standard vs. express), you can gain insights into the effectiveness of different shipping strategies and optimize your logistics network.
* **Bottleneck Detection**:
  + If the average delivery days start increasing, this could signal potential problems in your order fulfillment process, such as stock shortages, shipping delays, or capacity constraints. It’s essential to investigate the causes of delays and optimize the workflow.
* **Trend Analysis**:
  + Monitoring this metric over time (e.g., quarterly or yearly) allows you to track improvements or setbacks in delivery speed. If there’s a sharp increase in delivery times in a particular period, it could warrant a deeper investigation into operational challenges or external factors affecting logistics.

1. **Which products, categories, and subcategories are the most popular?**

* **Guidelines:**

Define Popularity Metric:

* Popularity can be measured by total sales volume (number of items sold) or total revenue generated. You should decide which metric makes the most sense for your analysis.
* Sales Volume (Metric):  
   To find the most popular products, categories, and subcategories, you can use the total sales volume, which is the sum of Order Quantity for each product, category, or subcategory.

Step-by-Step Process:

Step 1: Calculate Total Sales Volume  
You can calculate total sales volume using this DAX formula:

| TotalSalesVolume = SUM(Orders[Order Quantity]) |
| --- |

Step 2: Calculate Total Revenue (if desired)  
 If you'd like to also analyze revenue, use this formula:

| TotalRevenue = SUM(Orders[Sale Price]) + SUM(Orders[Shipping Fee]) |
| --- |

Step 3: Create Measures for Popularity  
 You can create separate measures to calculate the popularity of products, categories, and subcategories.

* Total Sales Volume for Products:

| ProductSalesVolume = SUM(Orders[Order Quantity]) |
| --- |

* Total Sales Volume for Categories:

| CategorySalesVolume = CALCULATE(SUM(Orders[Order Quantity]),ALLEXCEPT(Orders,Orders[Product Category])) |
| --- |

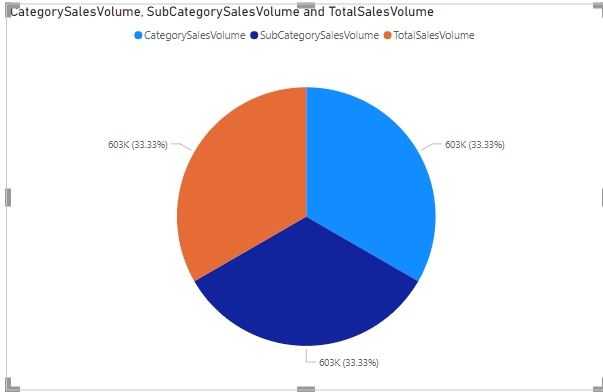
* Total Sales Volume for Subcategories:

| SubCategorySalesVolume = CALCULATE(SUM(Orders[Order Quantity]),ALLEXCEPT(Orders,Orders[SubCategory])) |
| --- |

Step 4: Create Visualizations

* For Product Popularity:  
   Create a bar or column chart with Product on the axis and either TotalSalesVolume or TotalRevenue as the value.  
   You can sort the products in descending order to show the most popular ones first.
* For Category and Subcategory Popularity:  
   Create a bar or column chart with Product Category and SubCategory on the axis.  
   Use TotalSalesVolume or TotalRevenue as the value to measure popularity.  
   Sort the chart to see the most popular categories and subcategories at the top.

**Which could look like this:**



### **Insights:**

* + **Most Popular Products:** Products that show the highest total sales volume or revenue can be considered the most popular.
  + **Category and Subcategory Insights:** Categories or subcategories contributing significantly to total sales or revenue indicate areas of high demand. These insights are useful for product assortment, marketing focus, and inventory management.
  + **Performance Trends:** By visualizing popularity over time, you can identify seasonal variations or shifts in customer preferences, which can inform future product strategies.

1. **Which products have seen an increase or decrease in sales over the year?**

### **Guidelines: For Analyzing Sales Trends for Products:**

Define the Metric for Analysis:

**Sales Trend Analysis** can focus on sales volume (number of items sold) or revenue over a specific period, such as year-over-year.

* **Sales Volume Trend:** Analyze the change in the quantity of products sold over time.
* **Revenue Trend:** Analyze the change in revenue generated for each product over time.

Step-by-Step Process:

**Step 1: Create a Measure for Total Sales Volume Per Year**Use this DAX formula to calculate total sales volume for a given year:

| YearlySalesVolume = CALCULATE(SUM(Orders[Order Quantity]),VALUES(Orders[year])) |
| --- |

**Step 2: Calculate the Difference in Sales Volume Year-Over-Year**To identify an increase or decrease, calculate the difference between the sales volume of the current year and the previous year:

| SalesVolumeChange = SUM(Orders[Order Quantity]) - CALCULATE(SUM(Orders[Order Quantity]),DATEADD(Orders[OrderDate],-1,YEAR)) |
| --- |

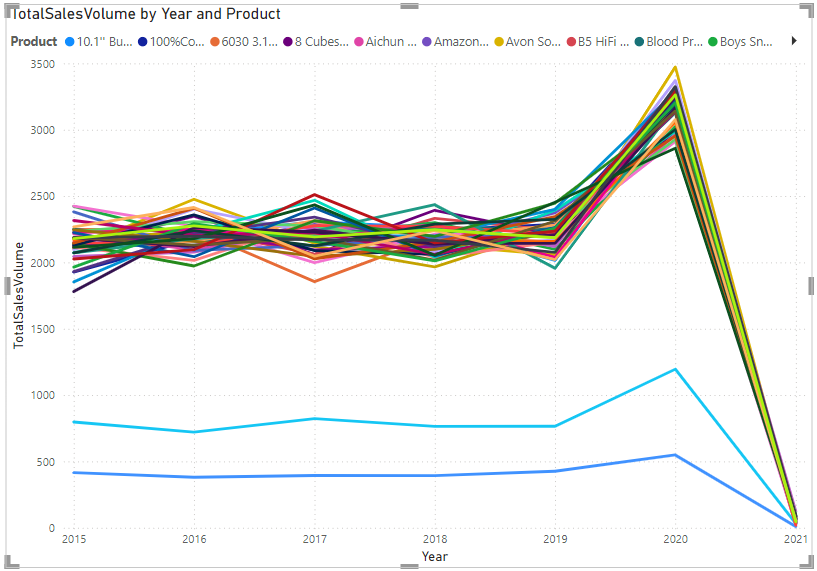
**Step 3: Create a Flag for Increase or Decrease**To easily identify the trend, create a new measure to indicate if sales increased or decreased:

| SalesTrend =  IF([SalesVolumeChange] > 0, "Increase",  IF([SalesVolumeChange] < 0, "Decrease", "No Change")) |
| --- |

**Step 4: Create Visualizations**

* + **Sales Trend for Products:**Create a line chart or a bar chart with Year on the x-axis and Sales Volume on the y-axis, with Product as a legend or slicer.
  + **Identify Increases or Decreases:**Use a table or matrix visualization to display Product, Year, SalesVolumeChange, and SalesTrend.

**Which could look like this:**



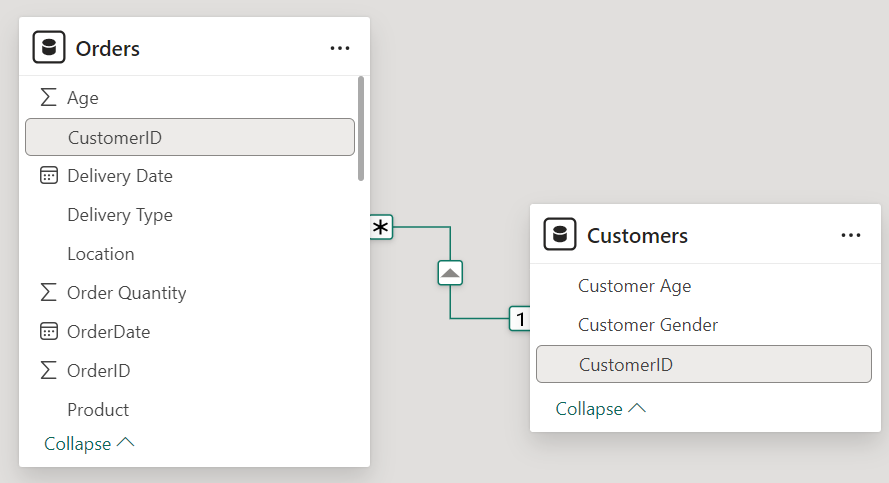
**Insights:**

* Highlight products with significant increases or decreases in sales volume or revenue.
* Analyze the possible reasons for these trends, such as promotions, seasonality, or changing customer preferences.

1. **While modeling the data relationships, what will be the type of relationship between the customer ID of Orders and customer tables?**

* **Guidelines:**
  + **Understand the Data Structure**:
    1. The **Customer Table** contains unique customer records with a unique identifier, **Customer ID** (primary key).
    2. The **Orders Table** contains multiple entries, each representing an order, where **Customer ID** acts as a foreign key to link each order to a customer.
  + **Define the Relationship Type**:
    1. The relationship between the **Customer ID** in the Orders table and the Customer table will be a **One-to-Many (1:\*) relationship**.
    2. Each customer can place multiple orders, but each order is linked to only one customer.
  + **Cardinality**:
    1. **One-to-Many (1:\*)**: One record in the Customer table corresponds to many records in the Orders table.
  + **Cross-Filter Direction**:
    1. Use **Single** direction to allow filters to flow from the Customer table to the Orders table.
    2. Use **Both** directions if analysis requires filters to propagate in both directions.
  + **Implementation**:
    1. In Power BI or any data modeling tool:
       1. Drag and drop to connect the **Customer ID** field in the Customer table to the **Customer ID** field in the Orders table.
       2. Set the cardinality to **One-to-Many (1:\*)**.

**Which could look like this:**



### **Insights:**

1. **Enables Customer-Centric Analysis**:
   * Linking these tables allows for customer-centric insights, such as total orders, revenue, or average order value per customer.
2. **Improved Data Filtering**:
   * Filtering by a specific customer in reports or dashboards will automatically filter their associated orders.
3. **Scalable for Aggregation**:
   * The relationship supports aggregated metrics like the number of orders, total revenue, and order frequency for each customer.
4. **Supports Drill-Down Analysis**:
   * Enables drill-down capabilities to view customer-level details, such as order patterns or behavior over time.
5. **Foundation for Advanced Analysis**:
   * This relationship forms the foundation for advanced analytics, such as identifying top customers, churn analysis, or customer segmentation.
6. **How have you handled the null values in the data?**

### **Guidelines: For Handling Null Values in Data:**

* + Identify Null Values:
    1. Inspect the dataset to locate fields with null or missing values using tools or queries (e.g., ISNULL in Power BI Data View).
  + Determine Context and Impact:
    1. Assess the importance of the column with null values and whether the missing data will affect calculations, visualizations, or insights.
  + Choose the Appropriate Strategy:
    1. Depending on the data context, use one of the following approaches:
  + a. Remove Null Values:
    1. If null values are insignificant (e.g., <5% of the dataset) and removing them won't distort insights, exclude those rows.
       1. Example: Filter out null rows in Power BI using a Remove Blank Rows transformation in Power Query.
  + b. Replace Null Values (Imputation):
    1. Replace nulls with appropriate default values:
       1. For Numeric Data: Use zero, mean, median, or another statistical measure.
       2. For Categorical Data: Use the mode or a default category like "Unknown."
       3. For Dates: Replace with a default or placeholder date (e.g., 01/01/1900).
  + c. Leave Null Values:
    1. If nulls represent valid missing data (e.g., data collection incomplete), retain them and handle their impact during analysis (e.g., filter them out in calculations).
  + Implement Changes:

In Power Query: Use Replace Values or Fill Down/Up options.

* + Validate Changes:
    1. Verify that the replacement or removal does not introduce inaccuracies or skew results.

**Insights:**

1. **Consistency in Data**:
   * Handling null values ensures a consistent dataset, reducing errors in calculations or visualizations.
2. **Improved Data Quality**:
   * Replacing or removing nulls enhances the reliability of the analysis.
3. **Tailored Strategy for Analysis**:
   * The chosen strategy (imputation or removal) should align with the type of analysis. For example:
     + Mean imputation for averages ensures unbiased results.
     + Mode imputation is helpful in categorical trends.
4. **Were there any data format issues in the data, and if there were/are how you would handle them?**

### **Guidelines: For Identifying and Handling Data Format Issues:**

Step 1: Load Data into Power BI

* Import your dataset using Power BI's Get Data option.
* Open the Power Query Editor (Home > Transform Data) to inspect the data structure.

Step 2: Identify Data Format Issues

1. **Check Data Types**:
   * In the Power Query Editor, each column’s data type is indicated at the top left of the column header.
   * Look for columns with incorrect data types (e.g., dates as text or numbers as strings).
2. **Spot Inconsistencies**:
   * Use the Column Profile (View > Column Profile) to visualize value distributions and spot issues like:
     + Blank values.
     + Mixed data types.
     + Unexpected values or outliers.
3. **Preview Errors**:
   * Any errors in rows will be flagged, allowing you to identify problematic data.

Step 3: Fix Data Format Issues

1. **Convert Column Data Types**:
   * Select the column and choose the correct data type from the dropdown in the ribbon (e.g., Date/Time, Decimal Number, Text).
2. **Standardize Text**:
   * Use Transform > Format options to:
     + Trim spaces (Trim).
     + Convert text to lowercase/uppercase (Uppercase or Lowercase).
3. **Fix Dates**:
   * Use the Date type for columns containing dates.
   * Use Transform > Date functions to extract or reformat date components (e.g., year, month).
4. **Handle Blank or Null Values**:
   * Use the Replace Values feature to replace blanks/nulls with appropriate defaults (e.g., 0 for numeric columns, Unknown for text fields).
   * Example:
     + Select a column > Right-click > Replace Values.
5. **Remove Special Characters**:
   * Use the Replace Values feature to eliminate unwanted characters (e.g., $, #).
   * Example: Replace "$100" with "100".
6. **Correct Numeric Columns Stored as Text**:
   * Convert text to numbers using the Decimal Number or Whole Number type.
   * Example: "123.45" stored as text converts to 123.45.
7. **Standardize Categorical Values**:
   * Group similar values (e.g., "CA" and "California") using the Group By or Replace Values options.

Step 4: Validate Changes

* Check the Applied Steps pane in the Power Query Editor to review transformations.
* Preview the data to ensure all issues are resolved.

Step 5: Close and Apply

* After making corrections, click Close & Apply to load the cleaned data into Power BI.

### **Insights:**

1. **Accurate Visualizations**: Properly formatted data ensures error-free reports and dashboards.
2. **Ease of Analysis**: Consistent formats allow you to focus on deriving insights instead of fixing errors repeatedly.
3. **Improved Reporting Speed**: Automated cleaning steps streamline data preparation for future updates.
4. **When we add a column in Power Query what’s the code that comes in M language in the formula bar? What do you know about M-query?**

### **Code in M Language When Adding a Column in Power Query**

When you add a column in Power Query, the formula bar generates a line of code in M language that represents the transformation applied. For instance:

* **Custom Column**:
  + If you add a custom column:

| = Table.AddColumn(#"Replaced Value", "Custom Column", each [Unit Price] + [Order Quantity]) |
| --- |

* **Conditional Column**:
* If you add a conditional column:

| = Table.AddColumn(#"Added Custom", "Conditional Column", each if [Order Quantity] >= 2 then "Order Taken" else if [Order Quantity] < 2 then "Order Rejected" else null) |
| --- |

### **Overview of M Query Language**

M, or Power Query Formula Language, is the scripting language used in Power Query for data transformation. It’s case-sensitive and functional in nature, with a syntax optimized for defining data manipulations.

#### **Key Features of M Query:**

* + Functional Language:
    1. M uses functions for all operations (e.g., Table.AddColumn, Table.TransformColumns).
    2. Each transformation step creates a new query object.
  + Step-by-Step Transformations:
    1. M records all data manipulation steps sequentially. Each step builds on the previous one.
  + Case-Sensitive:
    1. Column names and functions must match exactly in case (e.g., [Column1] is different from [column1]).
  + Immutable:
    1. M queries do not modify the original data source but create a transformed version.
  + Extensibility:
    1. You can write custom M code in the Advanced Editor to perform complex transformations.

### **Insights About M-Query:**

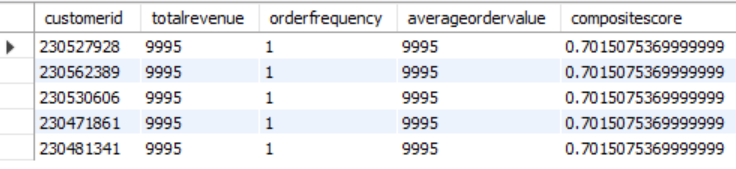
1. **Readability**: Each step in M is named descriptively, making it easy to understand the flow of transformations.
2. **Customizability**: While the Power Query UI generates M code, you can modify or write custom M code for advanced scenarios.
3. **Performance**: M is optimized for processing large datasets efficiently.
4. **Integration**: Works seamlessly with various data sources, including databases, Excel files, and web APIs.
5. **Identify the top 5 most valuable customers using a composite score that combines three key metrics: (SQL)**
   * **Total Revenue (50% weight): The total amount of money spent by the customer.**
   * **Order Frequency (30% weight): The number of orders placed by the customer, indicating their loyalty and engagement.**
   * **Average Order Value (20% weight): The average value of each order placed by the customer, reflecting the typical transaction size.**
   * **Guidelines :**

To identify the top 5 most valuable customers using a composite score, we need to combine three key metrics:

* + **Total Revenue (50% weight)**: This represents the total amount of money spent by the customer. This metric is essential for understanding which customers are generating the highest revenue for your business.
  + **Order Frequency (30% weight)**: This reflects the total number of orders placed by a customer. Customers who place frequent orders are typically more loyal and engaged with your brand.
  + **Average Order Value (20% weight)**: This metric represents the average value of each order placed by a customer. It gives an indication of the typical transaction size, and customers with higher average order values are often more profitable.

**SQL Query:**

| with customermetrics as (  select  o.customerid,  sum(o.saleprice \* o.orderquantity + o.shippingfee) as totalrevenue,  count(o.orderid) as orderfrequency,  avg(o.saleprice \* o.orderquantity + o.shippingfee) as averageordervalue  from orders o  group by o.customerid  ),  maxvalues as (  select  max(totalrevenue) as maxtotalrevenue,  max(orderfrequency) as maxorderfrequency,  max(averageordervalue) as maxaverageordervalue  from customermetrics  ),  weightedscores as (  select  cm.customerid,  cm.totalrevenue,  cm.orderfrequency,  cm.averageordervalue,  (0.50 \* cm.totalrevenue / mv.maxtotalrevenue) +  (0.30 \* cm.orderfrequency / mv.maxorderfrequency) +  (0.20 \* cm.averageordervalue / mv.maxaverageordervalue) as compositescore  from customermetrics cm  cross join maxvalues mv  )  select  c.customerid,  c.customername,  ws.totalrevenue,  ws.orderfrequency,  ws.averageordervalue,  ws.compositescore  from weightedscores ws  join customers c on ws.customerid = c.customerid  order by ws.compositescore desc  limit 5; |
| --- |

**Which could look like this:   
**

### **Insights Derived from the Composite Score:**

* + Customer Value Identification: The top 5 customers based on the composite score represent those who are the most valuable overall, considering their revenue, frequency of orders, and transaction size.
  + Loyalty and Engagement: The number of orders (Order Frequency) is an indicator of customer loyalty. Customers who frequently order are more likely to remain loyal and engaged with the brand.
  + High-Spending Customers: Total Revenue is a strong indicator of which customers are contributing the most to the business's bottom line. These customers are high-priority for marketing and retention efforts.

1. **Calculate the month-over-month growth rate in total revenue across the entire dataset. (SQL)**

### **Guidelines: For Calculating Month-over-Month Growth Rate in Total Revenue**

Data Preparation:

* Ensure all relevant fields are available, particularly SalePrice, OrderQuantity, and ShippingFee for revenue calculations.

Month Extraction:

* Use SQL functions like STR\_TO\_DATE to convert the OrderDate to a proper date format and extract the month and year ('%Y-%m').
* This allows grouping by month and enables comparison across different months.

Revenue Calculation:

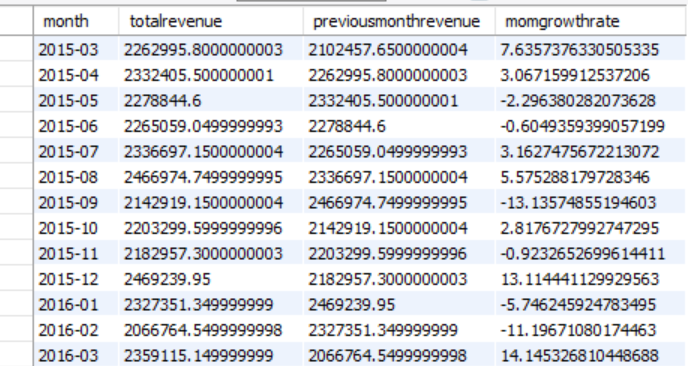
* Calculate total revenue per month by summing up the revenue generated by each order. In this case, the formula SalePrice \* OrderQuantity + ShippingFee is used to compute the total value of each order.
* Group the data by the extracted month.

Month-over-Month Calculation:

* To calculate MoM growth, use window functions like LAG() to retrieve the previous month's revenue for comparison.

**SQL Query:**

| with monthlyrevenue as (  select  date\_format(str\_to\_date(orderdate, '%d/%m/%Y'), '%Y-%m') as month,  sum(saleprice \* orderquantity + shippingfee) as totalrevenue  from orders  group by month  ),  revenueswithgrowth as (  select  mr.month,  mr.totalrevenue,  lag(mr.totalrevenue) over (order by mr.month) as previousmonthrevenue  from monthlyrevenue mr  )  select  month,  totalrevenue,  previousmonthrevenue,  case  when previousmonthrevenue is null then null  when previousmonthrevenue = 0 then null  else  ((totalrevenue - previousmonthrevenue) / previousmonthrevenue) \* 100  end as momgrowthrate  from revenueswithgrowth  order by month; |
| --- |

**Which could look like this: **

### **Insights from Month-over-Month Growth Calculation**

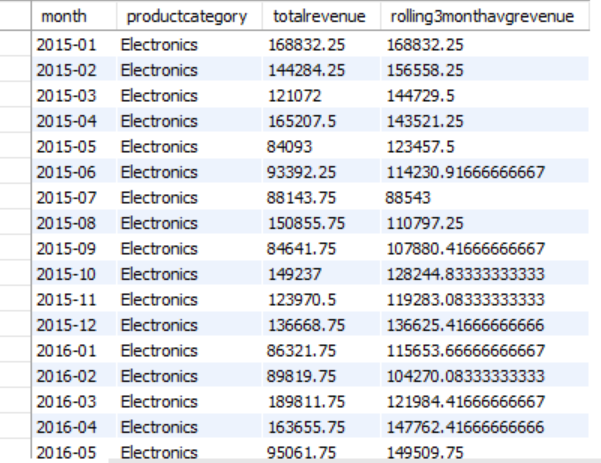
1. Identifying Growth Trends:
   * By calculating the MoM growth rate, you can identify months where the revenue has increased or decreased. This is valuable for spotting trends, such as seasonal fluctuations or the impact of promotional activities.
2. Tracking Business Performance:
   * Consistent growth in total revenue suggests that the business is expanding and the customer base is increasing.
3. Revenue Fluctuations:
   * If the growth rate is highly volatile, it might suggest issues such as supply chain problems, irregular promotions.
4. **Calculate the rolling 3-month average revenue for each product category. (SQL)**
   * **Guidelines:**

To calculate the Rolling 3-Month Average Revenue for each product category, we need to compute the monthly total revenue for each product category and then calculate a rolling 3-month average for each.

The Rolling 3-Month Average will consider the total revenue for the current month and the previous two months.

**SQL Query:**

| **with monthlyrevenue as (**  **select**  **date\_format(str\_to\_date(orderdate, '%d/%m/%Y'), '%Y-%m') as month,**  **productcategory,**  **sum(saleprice \* orderquantity + shippingfee) as totalrevenue**  **from orders**  **group by month, productcategory**  **),**  **rollingavgrevenue as (**  **select**  **month,**  **productcategory,**  **totalrevenue,**  **avg(totalrevenue) over (partition by productcategory order by month rows between 2 preceding and current row) as rolling3monthavgrevenue**  **from monthlyrevenue**  **)**  **select**  **month,**  **productcategory,**  **totalrevenue,**  **rolling3monthavgrevenue**  **from rollingavgrevenue**  **order by productcategory, month;** |
| --- |

**Which could look like this:   
  
Insights:**

* Smoothing of Revenue: The rolling 3-month average smooths the fluctuations in monthly revenue and gives a clearer picture of the long-term revenue trend.
* Identification of Seasonal Trends: This can help identify which product categories are performing better in certain months or seasons.

1. **Update the orders table to apply a 15% discount on the `Sale Price` for orders placed by customers who have made at least 10 orders. (SQL)**
   * **Guidelines:**

Identify Customers with More Than 10 Orders:

* First, determine which customers have placed at least 10 orders. This is essential because we only want to apply the discount to loyal customers.
* Use a GROUP BY clause with HAVING COUNT(OrderID) >= 10 to filter customers based on the number of orders.

Apply Discount to Sale Price:

* Once you have the list of customers who qualify, apply a 15% discount to their SalePrice. This can be done by multiplying the current SalePrice by 0.85 (i.e., 100% - 15%).

Safe Update Mode:

* + If you’re using MySQL Workbench and encountering issues with updating the table, ensure that the Safe Update Mode is turned off. This mode prevents updates unless a WHERE clause involving the primary key or indexed column is included.

**SQL Query:**

| **set sql\_safe\_updates = 0;**  **update orders o**  **join (**  **select customerid**  **from orders**  **group by customerid**  **having count(orderid) >= 10**  **) as eligible\_customers**  **on o.customerid = eligible\_customers.customerid**  **set o.saleprice = o.saleprice \* 0.85;** |
| --- |

### **Insights:**

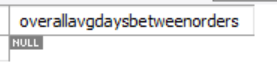
* + Customer Loyalty:
    1. The goal of applying a 15% discount to customers with more than 10 orders is to reward loyal customers and potentially increase their lifetime value.
    2. By targeting customers who frequently engage with the platform, you can improve retention rates and drive further sales.
  + Revenue Impact:
    1. Offering discounts to loyal customers can increase the perceived value of your products or services and encourage repeat purchases. However, it's crucial to ensure that the discount does not negatively affect the profitability of the company.

1. **Calculate the average number of days between consecutive orders for customers who have placed at least five orders. (SQL)**

**SQL Query:**

| with eligiblecustomers as (  select  customerid  from  orders  group by  customerid  having  count(orderid) >= 5  ),  orderintervals as (  select  o.customerid,  o.orderdate,  datediff(o.orderdate, lag(o.orderdate) over (  partition by o.customerid order by o.orderdate  )) as daysbetweenorders  from  orders o  where  o.customerid in (select customerid from eligiblecustomers)  ),  averagedayspercustomer as (  select  customerid,  avg(daysbetweenorders) as avgdaysbetweenorders  from  orderintervals  where  daysbetweenorders is not null  group by  customerid  )  select  avg(avgdaysbetweenorders) as overallavgdaysbetweenorders  from  averagedayspercustomer; |
| --- |

**Which could look like this:**



**Insights**:

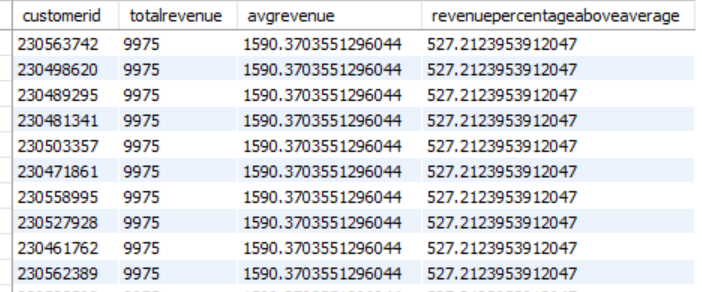
* + Frequent Buyers: Customers with shorter intervals between orders are loyal and ideal for loyalty programs or subscription services.
  + Irregular Buyers: Longer intervals suggest seasonal or low-priority purchases, requiring targeted campaigns or reminders.
  + Product Patterns: Categories like consumables drive frequent orders, while durable goods show longer purchase gaps.

1. **Identify customers who have generated revenue that is more than 30% higher than the average revenue per customer. (SQL)**

**SQL Query:**

| with customerrevenue as (  select  customerid,  sum(saleprice \* orderquantity) as totalrevenue  from orders  group by customerid  ),  averagerevenue as (  select  avg(totalrevenue) as avgrevenue  from customerrevenue  )  select  cr.customerid,  cr.totalrevenue,  ar.avgrevenue,  (cr.totalrevenue / ar.avgrevenue - 1) \* 100 as revenuepercentageaboveaverage  from  customerrevenue cr  join  averagerevenue ar  where  cr.totalrevenue > (ar.avgrevenue \* 1.30)  order by  cr.totalrevenue desc; |
| --- |

**Which could look like this:**



**Insights:**

* + High-Value Customers: Customers generating 30% more revenue than the average are vital assets, contributing significantly to overall sales.
  + Personalized Offers: These customers may benefit from exclusive discounts, early access to products, or premium memberships to encourage continued engagement.
  + Retention Focus: Prioritize retaining these customers through excellent service and customized loyalty rewards.

1. **Determine the top 3 product categories that have shown the highest increase in sales over the past year compared to the previous year. (SQL)**
   * **Guidelines:**

To determine the top 3 product categories with the highest increase in sales over the past year compared to the previous year, we can follow these steps:

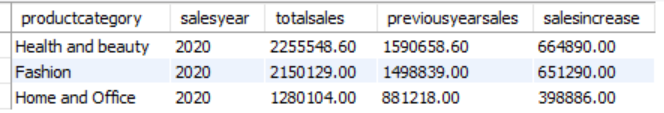
Steps:

1. **Extract Yearly Sales Data**:
   * Group sales data by year and product category.
   * Sum up the sales for each product category for each year.
2. **Calculate Year-over-Year Sales Increase**:
   * Compute the difference in sales between the current year and the previous year for each product category.
3. **Rank Product Categories**:
   * Rank product categories based on the sales increase and pick the top 3.

**SQL Query:**

| **with yearlycategorysales as (**  **select**  **date\_format(str\_to\_date(orderdate, '%d/%m/%Y'), '%Y') as salesyear,**  **productcategory,**  **sum(saleprice) as totalsales**  **from**  **orders**  **group by**  **salesyear, productcategory**  **),**  **categorysalesgrowth as (**  **select**  **ycs.productcategory,**  **ycs.salesyear,**  **ycs.totalsales,**  **lag(ycs.totalsales) over (partition by ycs.productcategory order by ycs.salesyear) as previousyearsales,**  **(ycs.totalsales - lag(ycs.totalsales) over (partition by ycs.productcategory order by ycs.salesyear)) as salesincrease**  **from**  **yearlycategorysales ycs**  **),**  **rankedcategories as (**  **select**  **csg.productcategory,**  **csg.salesyear,**  **csg.totalsales,**  **csg.previousyearsales,**  **csg.salesincrease,**  **dense\_rank() over (order by csg.salesincrease desc) as `rank`**  **from**  **categorysalesgrowth csg**  **where**  **csg.salesyear = (select max(salesyear) from orders)**  **)**  **select**  **productcategory,**  **salesyear,**  **round(totalsales, 2) as totalsales,**  **round(previousyearsales, 2) as previousyearsales,**  **round(salesincrease, 2) as salesincrease**  **from**  **rankedcategories**  **where**  **`rank` <= 3**  **order by**  **salesincrease desc;** |
| --- |

**Which could look like this:**

****

**Insights:**

1. Sales Trends:
   * This query identifies product categories with significant sales growth, indicating trending products or effective marketing.
2. Business Focus:
   * Highlighting top-performing categories helps businesses allocate resources effectively, focusing on high-growth areas.

**Subjective Question:**

1. **Explain the revenue breakdown by year and by-product. Evaluate how different products contribute to annual revenue and come up with suggestions to increase the sales of the low-selling items.**
   * **Guidelines:**

Analyze revenue breakdown by year and product to understand how different products contribute to annual revenue and provide actionable suggestions to improve sales for low-performing items.

**Prepare the Data in Power Query**

1. **Convert OrderDate to Date Format**:
   * Go to the Power Query Editor.
   * Select the OrderDate column.
   * Change the data type to "Date."
2. **Extract Year**:
   * Add a new column to extract the year from OrderDate:
     + Go to the "Add Column" tab.
     + Use the formula: Year([OrderDate]).
3. **Calculate Revenue**:
   * Add a custom column named Revenue with the formula: [SalePrice] \* [OrderQuantity].

**Create Measures in Power BI:**

**Total Revenue by Product:**

| **TotalRevenue by Product = SUM(Orders[Revenue])** |
| --- |

**Total Revenue by Year:**

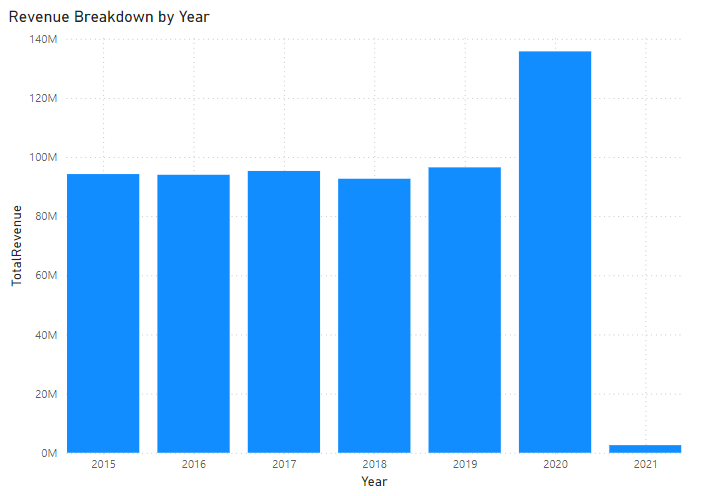
| **RevenueByYear = CALCULATE(SUM(Orders[year]),VALUES(Orders[year]))** |
| --- |

**Contribution to Annual Revenue:**

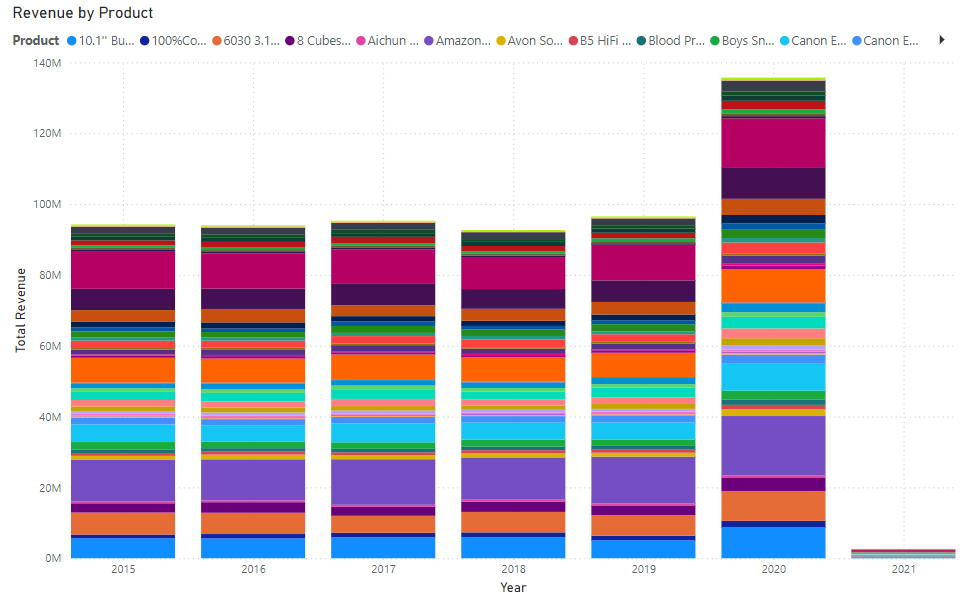
| **ContributionPercentage = DIVIDE(SUM(Orders[year]),CALCULATE(SUM(Orders[Revenue]),ALLEXCEPT(Orders,Orders[year])))** |
| --- |

**Which could look like this:**

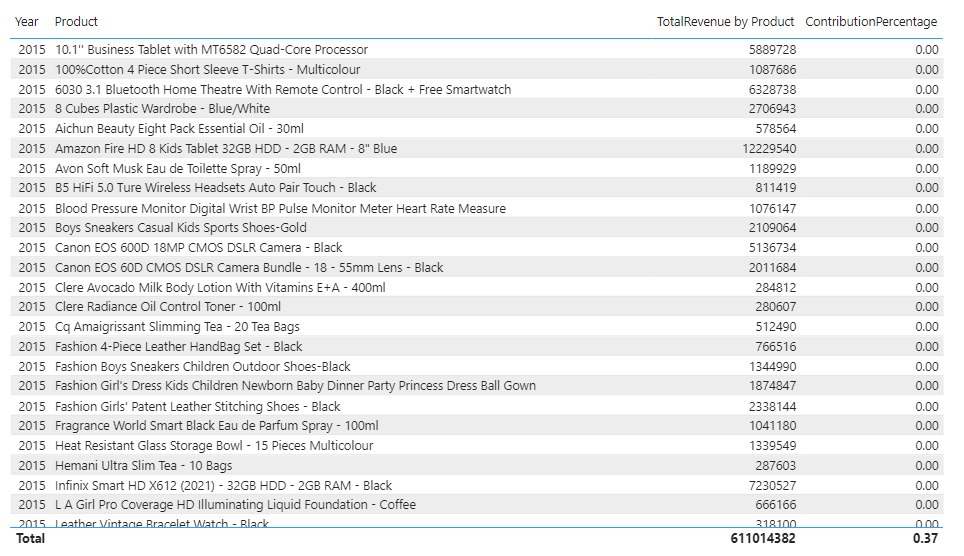
**Revenue Breakdown by Year:**



**Revenue by Product**:



**Product Contribution**:



Insights:

* Identify low-performing products by looking at their contribution percentage.
* Highlight products with consistent or increasing revenue trends.

Suggestions:

* Marketing Focus: Promote low-performing products with discounts, bundles, or campaigns.
* Customer Feedback: Investigate reasons for low sales by collecting customer feedback.
* Targeted Discounts: Use data to create targeted discounts for underperforming products.
* Product Placement: Optimize placement in catalogs or online stores to enhance visibility.

1. **How many products were returned? Use a DAX function to get this metric. Examine the possible reasons for returns and consider how this metric could indicate improvements in product descriptions or quality control.**
   * **Guidelines:**

Determine the total number of products returned and analyze possible reasons for returns. Use this insight to suggest improvements in product descriptions, quality control, or other operational areas.

Steps to Calculate Returns Using DAX :

Create a Measure to Count Returned Products

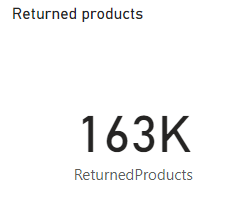
| ReturnedProducts = SUMX(FILTER(Orders,Orders[Status] = "Returned"),Orders[Order Quantity]) |
| --- |

**Alternative: Count Unique Returns**

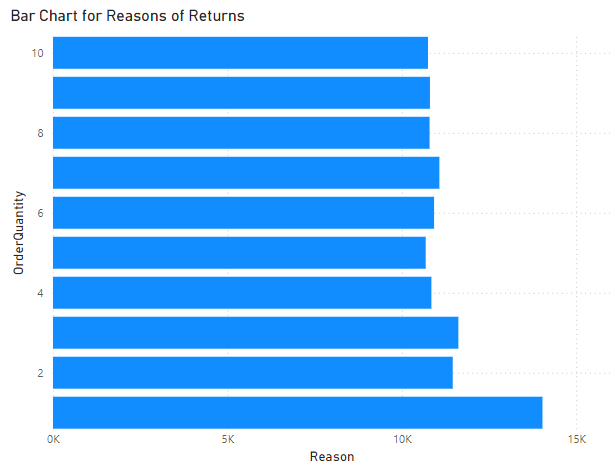
| UniqueReturnedProducts = CALCULATE(DISTINCTCOUNT(Orders[Product]),Orders[Status] = "Returned") |
| --- |

**Which could look like this:**

**Create a Card Visualization:**

****

**Bar Chart for Reasons of Returns:**

****

Insights:

* Primary Reasons for Returns:
  + Identify the most frequent reasons for returns (e.g., "Description mismatch," "Damaged product").
* Category Analysis:
  + Determine if certain product categories or subcategories have higher return rates.
* Customer Trends:
  + Evaluate if specific customers or regions report returns more frequently.

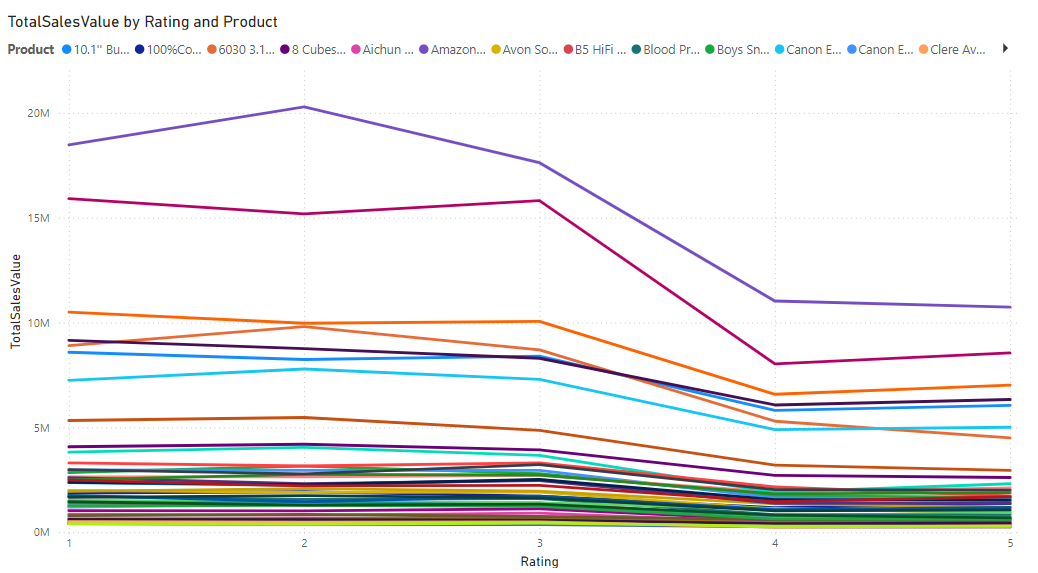
1. **Whenever a customer goes to Amazon, they’ll filter the most rated products to buy the better category. Can you verify this using any visualization or table that the ratings of products impact their sales value?**
   * **Guidelines:**

To analyze the relationship between product ratings and sales value in Power BI, you can follow these steps:

Create a New Measure for Total Sales Value

| TotalSalesValue = SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity]) |
| --- |

**Which could look like this:**

****

### **Insights & Recommendations**

* + High-Rating Products Drive Sales:
    1. Focus on maintaining high ratings for products through better quality and descriptions.
    2. Highlight top-rated products in promotional materials.
  + Low-Rating Products Require Attention:
    1. Investigate and address common issues in reviews.
    2. Consider product improvements or removal of consistently poorly rated products.
  + Leverage Ratings for Marketing:
    1. Use ratings as a filter for personalized recommendations to customers.
    2. Offer discounts or promotions on top-rated products to increase visibility.

1. **Investigate how revenue distribution varies across different locations. Explore which geographical areas contribute most to sales and consider the strategic implications for regional marketing and distribution efforts. How might location-based trends inform the company's market segmentation and resource allocation approach?**

* **Guidelines:**

To investigate revenue distribution across locations and derive strategic insights in Power BI, follow these steps:

### **Steps for Analysis in Power BI**

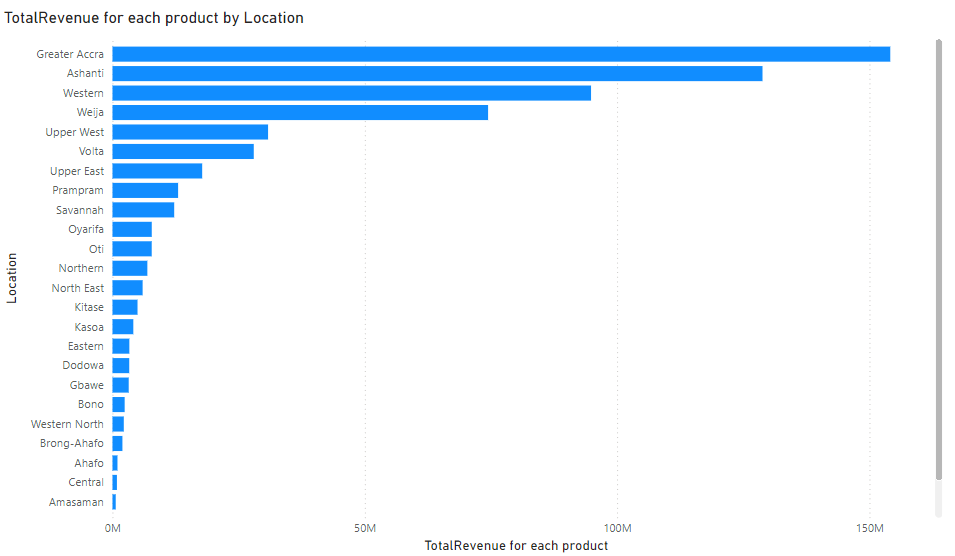
Create a Measure for Revenue:

| TotalSalesValue = SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity]) |
| --- |

**Geographic Visualization**



**Revenue Distribution:**



#### **Insights:**

* + High-Contributing Locations:
    1. Identify the locations that generate the most revenue.
    2. Investigate what drives success in these areas (e.g., population, demand, effective marketing).
  + Low-Contributing Locations:
    1. Pinpoint underperforming regions and analyze potential reasons (e.g., lack of awareness, logistical challenges).
    2. Cross-check with demographic and competitor data to identify opportunities.
  + Regional Revenue Trends:
    1. Observe trends over time in different locations to assess the effectiveness of marketing or new product launches.
  + Look for seasonal spikes or consistent patterns in revenue distribution.

1. **Determine which month could benefit from enhanced promotional offers to boost sales. Can you suggest some targeted marketing strategies here?**

* **Guidelines:**

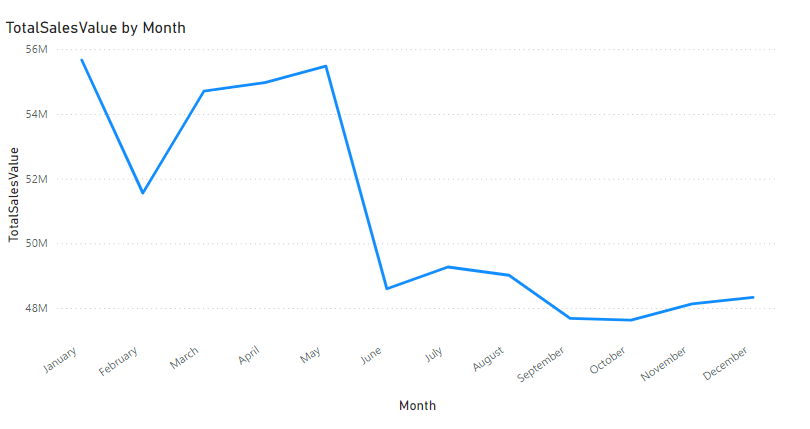
To determine which month could benefit from enhanced promotional offers and suggest targeted marketing strategies, follow these steps in **Power BI**:

**Create Measures** for **Total Revenue**:

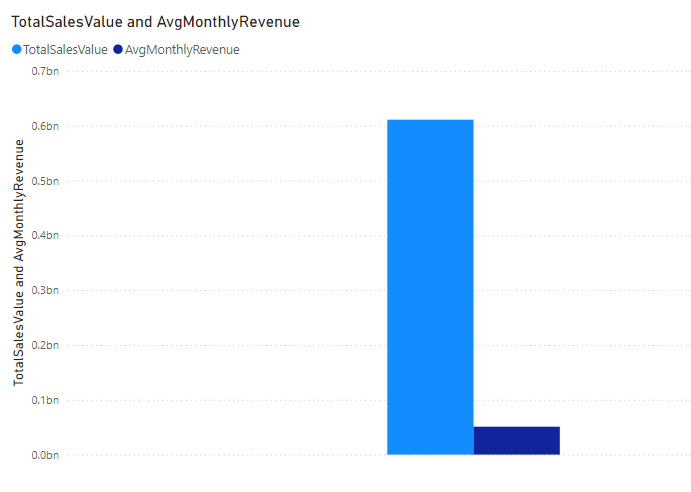
| TotalSalesValue = SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity]) |
| --- |

Create a measure for **Average Monthly Revenue**:

| AvgMonthlyRevenue = AVERAGEX(SUMMARIZE(Orders,Orders[OrderDate].[Month],"Revenue",SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity])),[Revenue]) |
| --- |

Monthly Revenue Trends  


Average Monthly Revenue



### **Insights**

#### **Identify Low-Performing Months**

* Look for months with revenue significantly below the average. These months could benefit from targeted promotions.
* Cross-check with seasonal trends or historical data to validate findings.

#### **Consider External Factors**

* Holidays, weather, or industry-specific events may influence monthly sales.
* Align promotional efforts with these factors.

1. **Identify which products may require increased marketing efforts. Which items have high prices yet underperform in sales?**

* **Guidelines:**

To identify which products may require increased marketing efforts, focusing on products with high prices but low sales performance is a great strategy. Here's how you can approach this in Power BI using DAX and visualization:

### **Steps to Identify Underperforming High-Priced Products:**

#### **1. Calculate Total Sales:**

| TotalSalesValue = SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity]) |
| --- |

2. **Calculate Average Sales Price (Unit Price)**:

| AvgUnitPrice = AVERAGE(Orders[UnitPrice]) |
| --- |

#### **3. Calculate Total Quantity Sold:**

| TotalQuantitySold = SUM(Orders[Order Quantity]) |
| --- |

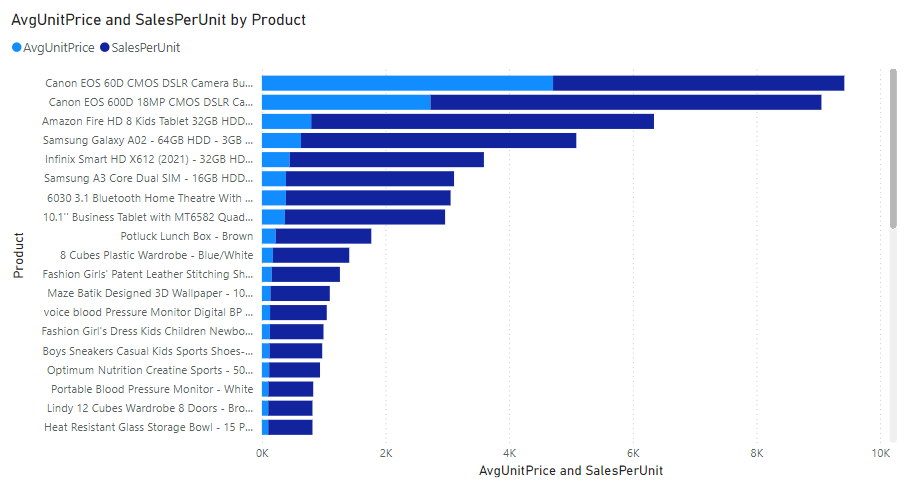
#### **4. Create a Metric to Compare Price vs Sales:**

| SalesPerUnit = DIVIDE([TotalSalesValue],[TotalQuantitySold]) |
| --- |

#### **5. Filter High-Price Products with Low Sales:**

| HighPriceLowSalesFlag = IF([AvgUnitPrice] > AVERAGE(Orders[Unit Price]) && [SalesPerUnit] < AVERAGE(Orders[Sale Price]),1,0) |
| --- |

**Which could look like this:**

****

Insights:

1. High Price, Low Sales: Products with high prices but low sales may lack perceived value or may not resonate with the target audience.
2. Price Sensitivity: Customers are more price-sensitive for premium products unless clearly differentiated or justified by high quality.
3. Potential Positioning Issues: These products may not be properly positioned in the market or marketed to the right segment.

Recommendations:

1. Introduce Discounts and Offers: Offer limited-time discounts or bundle high-priced products to increase perceived value and incentivize purchases.
2. Enhance Marketing: Highlight unique product features, quality, and benefits in targeted marketing campaigns, focusing on premium features.
3. **Assess which products should have discounts. How can targeted incentives drive sales and customer loyalty for specific products?**

* **Guidelines:**
  + Understand the Scope of Discounts:
    1. Clearly define the discount percentage (e.g., 15% in this case) and ensure consistency in its application across all calculations.
  + Use Aggregated Measures for Clarity:
    1. Create measures like SalesAfterDiscount and TotalSales for easy comparison and visualization.

Calculate Measures:

**Total Sales Before Discount**:

| TotalSalesBeforeDiscount = SUMX(Orders,Orders[Sale Price] \* Orders[Order Quantity]) |
| --- |

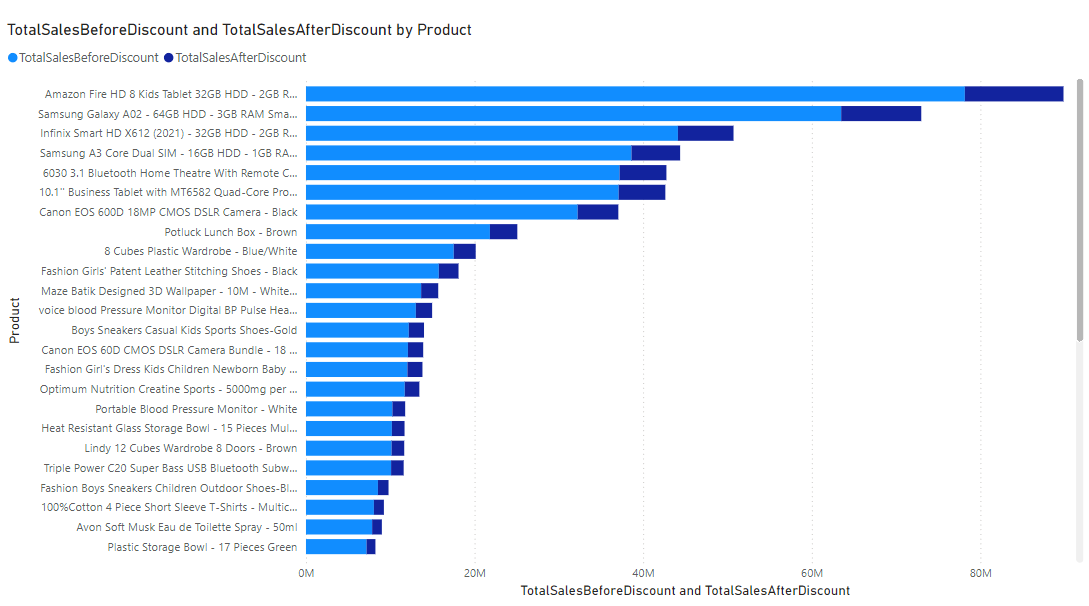
**Total Sales After Discount**:

| TotalSalesAfterDiscount = SUMX(Orders,Orders[DiscountApplied] \* Orders[Order Quantity]) |
| --- |

**For DiscountApplied**

| DiscountApplied = SUMX(Orders,Orders[Unit Price] \* Orders[Order Quantity] \* 0.15) |
| --- |

**Which could look like this:**



### **Insights from Discount Analysis**

* + Discount Efficiency:
    1. Products with high post-discount sales indicate successful campaigns or customer interest driven by the price reduction.
    2. Products with minimal sales uplift despite discounts might require additional strategies like improved marketing or better descriptions.
  + Customer Behavior:
    1. Discounts may incentivize bulk purchasing for certain customer groups.
    2. Some regions or customer segments may respond better to discounts than others.
  + Revenue Balance:
    1. Compare the total revenue before and after discounts to assess the net impact on profitability.

1. **Come up with a loyalty program to benefit the company’s customers. From the available lot of customers come up with strategies to bucket them and provide benefits under different loyalty programs.**

* **Guidelines:**

To benefit the company’s customers, a loyalty program can be designed based on customer spending, purchase frequency, and engagement levels. Below are strategies to bucket customers and provide benefits under different loyalty tiers.

Strategies for Loyalty Program Implementation

* + **Data Segmentation**
    1. Use historical data to segment customers into loyalty tiers based on total revenue and purchase frequency.
    2. Update tiers dynamically every quarter or six months to reflect recent customer behavior.
  + **Gamification**
    1. Introduce a points system where customers earn points for each dollar spent. Points can be redeemed for discounts or gifts.
    2. Provide bonus points for referrals or completing customer profiles.
  + **Personalized Marketing**
    1. Send personalized emails or notifications with tailored offers and rewards based on customer tier and past purchases.
    2. Use purchase history to recommend relevant products and upsell.

Loyalty Tier Calculation:

| LoyaltyTier = SWITCH(TRUE(),[TotalSalesValue] < 500,"Bronze",[TotalSalesValue] <= 1500, "Silver","Gold") |
| --- |

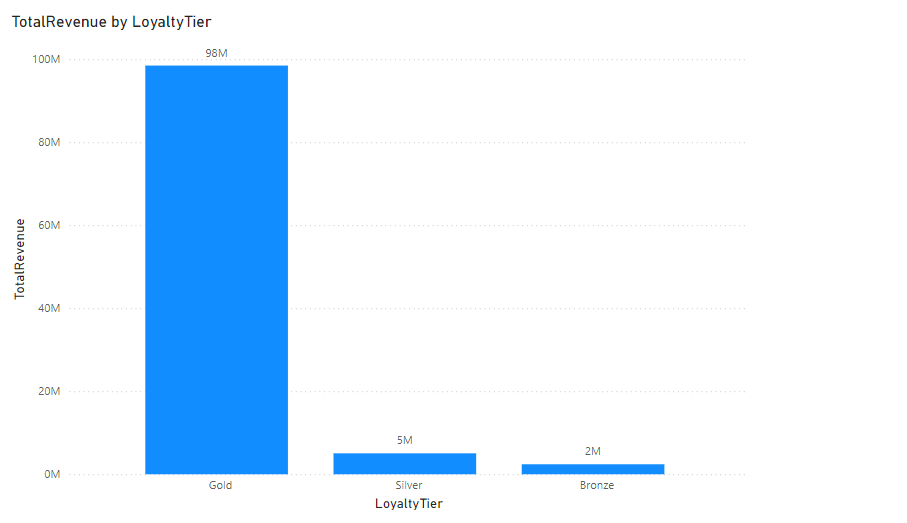
Total Revenue Calculation

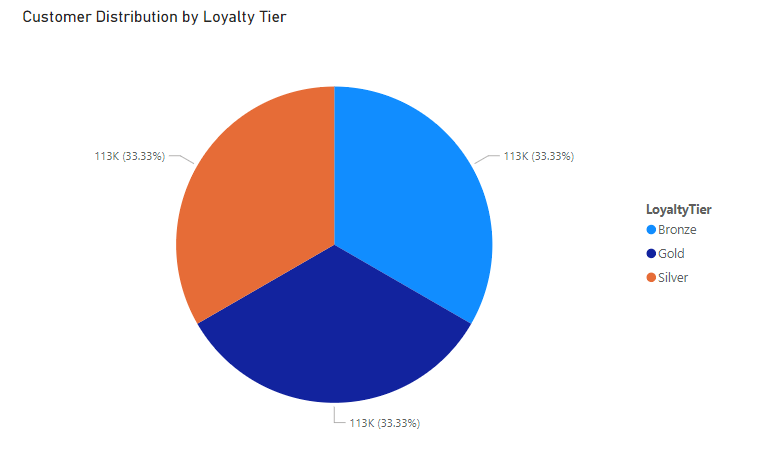
| TotalRevenue = SUMX(Orders, Orders[Unit Price] \* Orders[Order Quantity]) |
| --- |

At Risk Customers

| AtRiskCustomers = IF([DaysSinceLastPurchase] > 180, "Yes", "No") |
| --- |

**Which could look like this:**

****

****

**Insights:**

Distribution of At-Risk Customers by Loyalty Tier:

* Bronze Tier is likely to have the highest percentage of "At Risk" customers, as these customers may have low engagement or sporadic purchasing behavior.
* Gold Tier customers show fewer "At Risk" percentages, indicating stronger loyalty and consistent purchasing patterns.

Revenue Contribution from At-Risk Customers:

* If "At Risk" customers in higher loyalty tiers contribute significantly to revenue, their retention should be prioritized.
* "At Risk" customers in the Bronze Tier may contribute less revenue but could indicate untapped potential.

Retention Trends:

* A higher percentage of "Not At Risk" customers in the Silver and Gold tiers demonstrates the effectiveness of current loyalty strategies for these groups.
* Bronze Tier customers need further engagement efforts, as they may be newer or less attached to the brand.

1. **Using the DAX functions Calculate and a row iteration DAX function calculate the total sales for the Product Category “Fashion” and delivery type “Shipped from Abroad”. What are the other types of DAX functions you have used in the project?**
   * **Guidelines:**

To calculate the total sales for the Product Category “Fashion” and Delivery Type “Shipped from Abroad”, use the following DAX formula:

| TotalSales\_Fashion\_Abroad =  CALCULATE(  SUMX(Orders, Orders[Unit Price] \* Orders[Order Quantity]),  Orders[Product Category] = "Fashion",  Orders[Delivery Type] = "Shipped from Abroad"  ) |
| --- |

### 

### **Other Types of DAX Functions Used in the Project**

1. Aggregation Functions

* SUM: Calculates the sum of a column, e.g., SUM(Orders[Order Quantity]).
* AVERAGE: Finds the mean of a column, e.g., AVERAGE(Orders[Unit Price]).
* COUNT: Counts the number of rows in a table.

2. Iterative Functions

* SUMX: Row-by-row evaluation and aggregation.
* AVERAGEX: Calculates the average of an evaluated expression.

3. Logical Functions

* IF: Applies conditional logic, e.g., IF([TotalRevenue] > 1000, "High", "Low").
* SWITCH: Simplifies multiple conditional evaluations.

4. Filter Functions

* FILTER: Returns a subset of rows based on specific criteria.
* ALLEXCEPT: Removes filters from all columns except the specified ones.

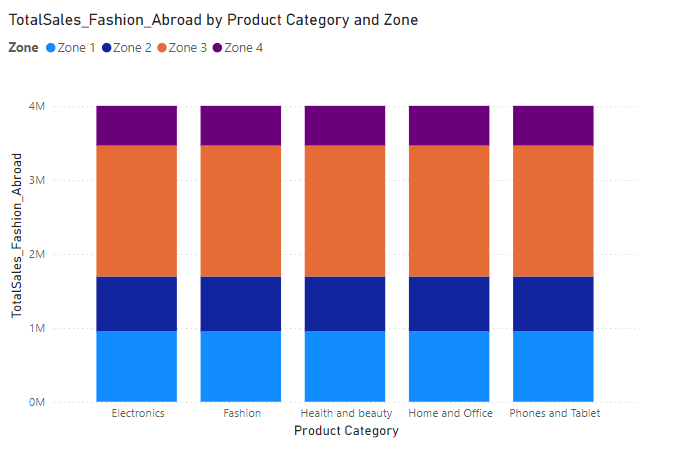
5. Time Intelligence Functions

* TOTALYTD: Calculates year-to-date totals.
* DATEADD: Adjusts dates by a specified interval (e.g., months, years).

6. Statistical Functions

* DISTINCTCOUNT: Counts distinct values in a column.
* MAX, MIN: Finds the maximum or minimum value in a column.

**Which could look like this:**

****

Insights

1. Revenue Contribution:  
   This calculation identifies the revenue generated by the Fashion category with a specific delivery type. Insights can help prioritize these product-delivery combinations for marketing or operational focus.
2. Operational Insights:  
   By isolating Delivery Type, the business can assess the performance of imported goods and identify logistical or pricing opportunities.

Recommendations

1. Targeted Discounts:  
   Offer promotions or discounts on "Fashion" products under "Shipped from Abroad" to drive higher sales volume.
2. Customer Segmentation:  
   Segment customers purchasing these products and target them with loyalty programs.
3. **Wait Times Correlated with Demographics and Care: Explore how average wait times vary across different product categories to optimize scheduling and staffing.**

**Guidelines:**

#### **Steps to Analyze and Visualize Wait Times by Product Categories**

#### **Wait Time (Difference between Delivery Date and Order Date):**

| WaitTime = DATEDIFF(Orders[OrderDate], Orders[Delivery Date], DAY) |
| --- |

**Total Revenue** (Revenue per Order)

| TotalRevenue = SUMX(Orders, Orders[Unit Price] \* Orders[Order Quantity]) |
| --- |

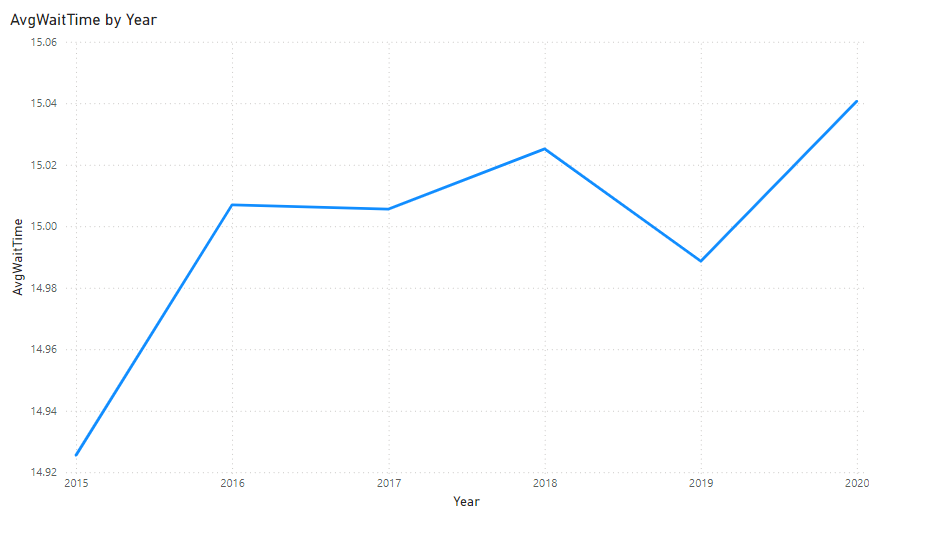
#### **Customer Lifetime Value (CLV) (Aggregate revenue per customer):**

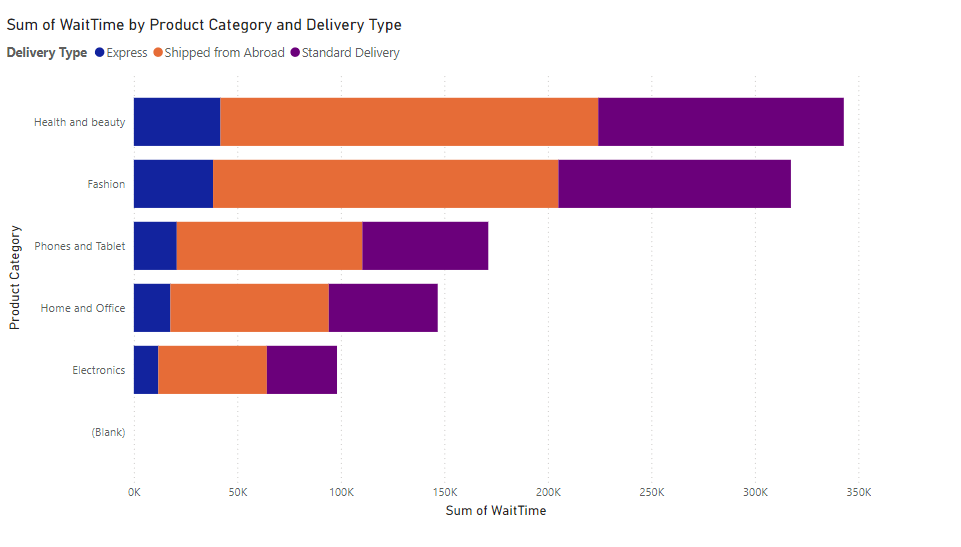
| CLV = CALCULATE(  SUMX(Orders, Orders[Unit Price] \* Orders[Order Quantity]),  ALLEXCEPT(Orders, Orders[CustomerID])  ) |
| --- |

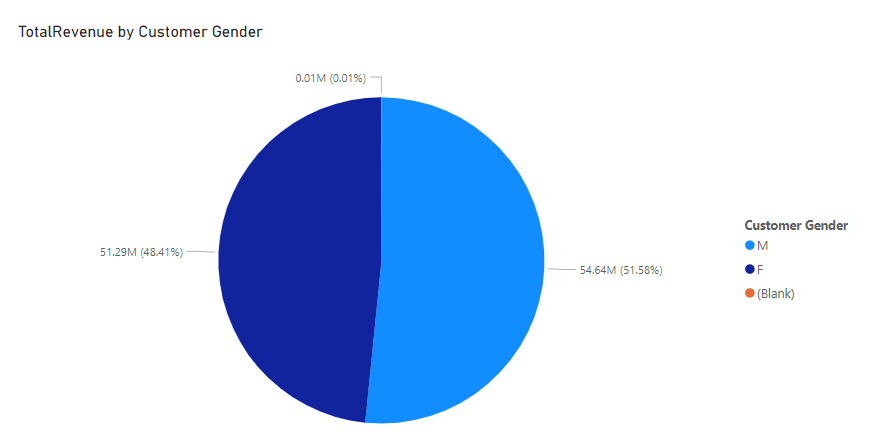
**Average Wait Time**:

| AvgWaitTime = AVERAGE(Orders[WaitTime]) |
| --- |

**Which could look like this:**







Insights Extraction:

* Identify high wait times: Look for Product Categories or Delivery Types with the highest wait times. This helps pinpoint areas for improvement.
* Customer demographics: Assess how customer age or gender impacts purchasing behavior and satisfaction levels.
* Geographical trends: Review Location data to identify regions with high demand or frequent delivery delays.

Recommendations:

* Enhance Delivery Times:
  + Focus on products with longer wait times and consider streamlining processes or offering express shipping in regions with frequent delays.
* Targeted Marketing:
  + Use demographic data (age, gender) to personalize marketing campaigns. For instance, a younger audience might respond well to promotions or social media ads.

1. **Explore if there is any relationship between the Delivery type and waiting time between ordering and receiving an item.**

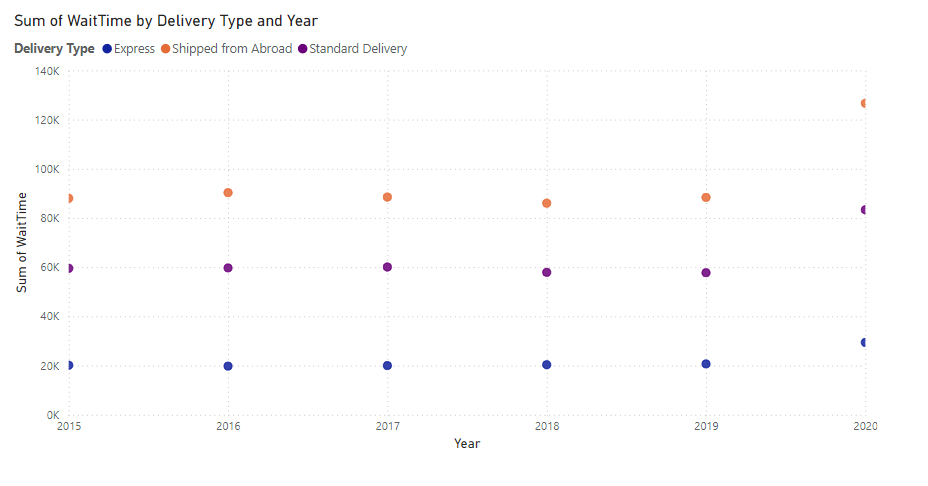
* **Guidelines:**

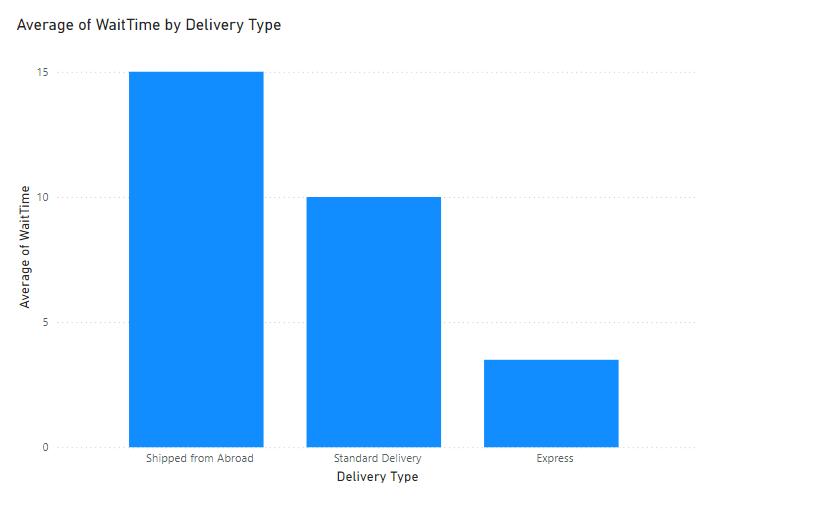
To explore the relationship between Delivery Type and Wait Time (the time between Order Date and Delivery Date), follow these steps in Power BI:

calculated column for **Wait Time**:

| WaitTime = DATEDIFF(Orders[OrderDate], Orders[Delivery Date], DAY) |
| --- |

**Which could look like this:**

****

****

Insights:

* Delivery Type Impact: Does a particular delivery type (e.g., Standard vs. Express) tend to have higher wait times? Express delivery might have shorter wait times, while standard delivery could be more variable.
* Outliers: Are there any delivery types that consistently have very high or low wait times? You can use the box plot to identify outliers.
* Seasonal Variations: Does the delivery type’s wait time change based on the season or month? If there’s a pattern, you might want to optimize the delivery processes during peak times.

Recommendations:

* Improve Processes for Long Wait Times: If certain delivery types are consistently slow, you can investigate bottlenecks in those delivery methods.
* Offer Alternatives: If the standard delivery type has high wait times, consider offering customers an option to upgrade to express delivery for faster service.
* Customer Communication: Inform customers about expected wait times based on their selected delivery type, and provide proactive communication if delays occur.

1. **Is there any relationship between shipping charges and product type?**

* **Guidelines:**

To explore the relationship between Shipping Charges and Product Type:

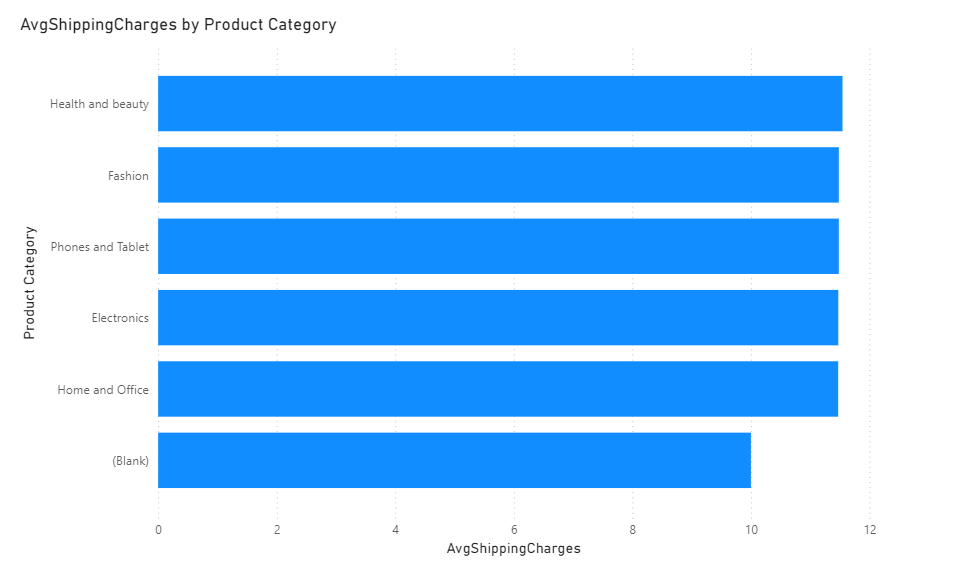
* Total Shipping Charges for Product Type:

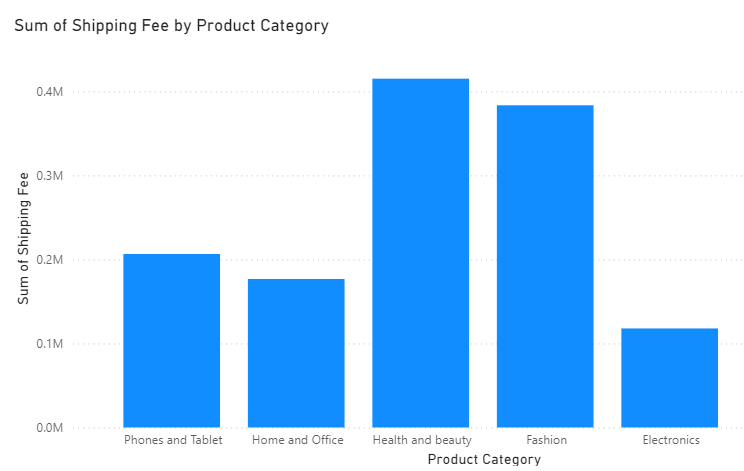
| TotalShippingCharges = SUM(Orders[Shipping Fee]) |
| --- |

* Average Shipping Charges by Product Type

| AvgShippingCharges = AVERAGE(Orders[Shipping Fee]) |
| --- |

**Which could look like this:**





Insights Extraction

* + Higher Shipping Charges for Certain Product Types: Certain product types (e.g., large, bulky items) may have higher shipping charges than others (e.g., small accessories).
  + Shipping Charges Variability: Some product types might have a wide range of shipping charges, possibly due to different delivery methods or geographical factors.

Recommendations

* Optimize Shipping Fees: If certain product types consistently have high shipping charges, investigate ways to reduce shipping costs, such as by negotiating better rates with carriers for large or bulky items.

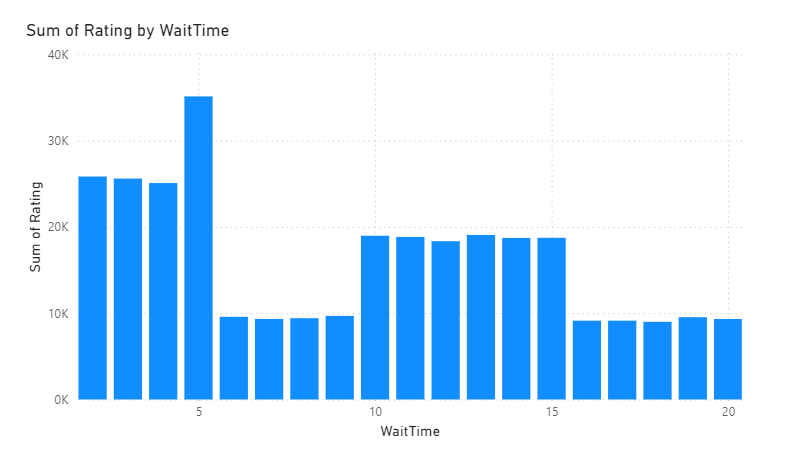
1. **Come up with strategies to decrease the low rating orders after analyzing different factors like waiting time, shipping type, unit price, etc.**

* **Guidelines:**

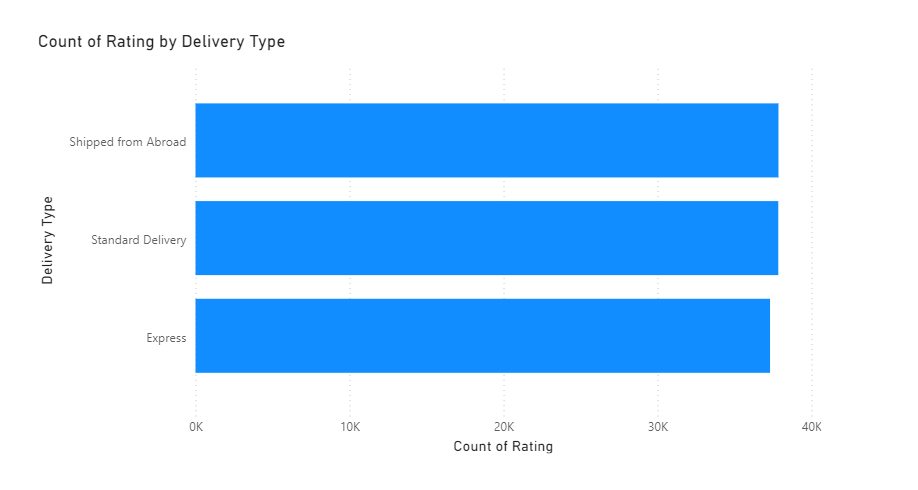
To decrease **low-rating orders**, you need to analyze the contributing factors (e.g., waiting time, shipping type, unit price) and implement targeted strategies. Here's a step-by-step approach to analyze and strategize using Power BI:

**Which could look like this:**

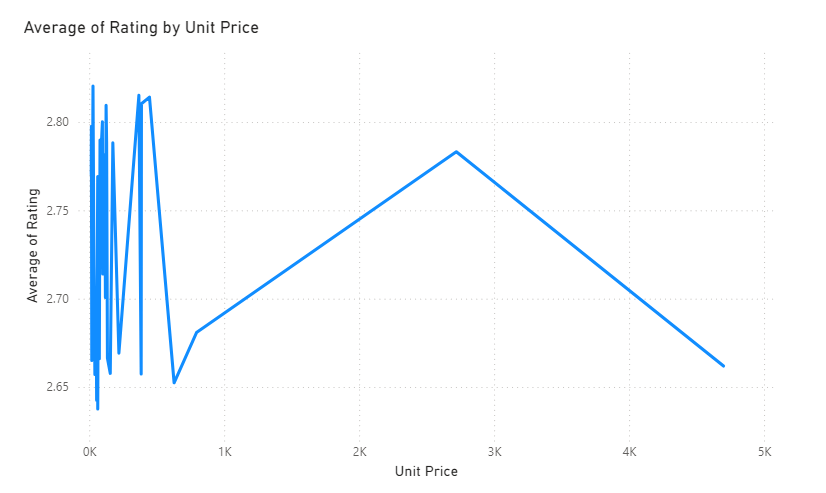
**Wait Time vs. Rating**:



**Delivery Type vs. Rating**:



**Unit Price vs. Rating**:



Insights

1. Delivery Type Impact on Wait Time: Certain delivery types, such as "Standard" or "Economy," show significantly higher average wait times compared to "Express" or "Priority." This likely contributes to lower customer satisfaction for these delivery types.
2. Delivery Type vs. Ratings: "Express" delivery orders consistently receive higher average ratings, indicating that customers value faster delivery. Conversely, "Standard" delivery has a higher proportion of low ratings, suggesting dissatisfaction with longer delivery times.

Recommendations

1. Enhance Delivery Speed for Standard Options: Optimize logistics for standard delivery by reducing processing times or partnering with faster carriers. Alternatively, offer incentives like free shipping for delayed orders to improve satisfaction.
2. Promote Express Delivery Options: Highlight the benefits of "Express" delivery in marketing campaigns and consider offering discounts or loyalty perks to encourage customers to choose faster delivery options.
3. **Using the time intelligence DAX function, create a table to compare each month’s sales with the previous year’s same month’s total sales. So there will be four columns in the output year, month, total sales, previous\_years\_sales.**

* **Guidelines:**

To create a table comparing each month's sales with the previous year's sales in Power BI, you can use the time intelligence DAX functions.

#### **DAX Measures:**

**Total Sales**:

| TotalSales = SUM(Orders[Sale Price]) |
| --- |

**Previous Year's Sales**:

| PreviousYearsSales = CALCULATE(  [TotalSales],  SAMEPERIODLASTYEAR('Orders'[OrderDate].[Date])  ) |
| --- |

**Which could look like this:**



Insights

1. Year-over-Year Growth Patterns: Some months show significant growth in total sales compared to the previous year (e.g., February sales grew by 10%).
2. Performance Variability: Months like January might show a slight increase (e.g., 5%), while others like December could experience a decline.

Recommendations

1. Targeted Promotions for Underperforming Months: For months with declining or flat sales (e.g., December).
2. Replicate Successful Strategies: Analyze months with significant year-over-year growth to identify what worked well (e.g., product launches, discounts).
3. **What do you understand by PowerBI gateway? What are its use cases?**

* **Guidelines**

A Power BI Gateway is a bridge that connects on-premises data sources to cloud services like Power BI, Power Apps, and Azure Logic Apps. It allows secure data transfer between your local data sources (like SQL Server, Excel files, or SharePoint) and Power BI Service without moving the data to the cloud.

There are two main types of gateways:

1. Personal Mode Gateway: Used by individual users for reports that don't require collaboration.
2. Standard/Enterprise Gateway: Supports multiple users and connections for enterprise-level collaboration.

**Key Use Cases of Power BI Gateway**

1. Real-Time Data Refresh:
   * Use the gateway to schedule and automate data refresh for Power BI reports and dashboards from on-premises databases.
   * Ensures up-to-date insights for users without manually updating datasets.
2. Live Queries to On-Premises Data:
   * Enable live query connections to data sources like SQL Server, Oracle, or SAP.
   * Provides real-time analysis without replicating data in the cloud.
3. Secure Data Access:
   * Ensure secure communication between Power BI Service and on-premises data sources using encryption.
   * Keeps sensitive data within your network while enabling insights in Power BI.
4. Multi-Source Integration:
   * Combine data from multiple sources (cloud and on-premises) for unified analytics.
   * Example: Blend Salesforce cloud data with an on-premises SQL database.
5. Support for Other Microsoft Services:
   * The gateway supports Power Apps, Power Automate, and Azure services, enabling workflows and app automation using on-premises data.
6. Enterprise Collaboration:
   * Enterprise gateways allow multiple users to connect to shared datasets and work collaboratively.

**Benefits of Power BI Gateway**

* + Seamless Integration: Connect to existing systems without changing infrastructure.
  + Cost-Efficient: Avoid moving data to the cloud, saving storage costs.
  + Enhanced Security: Uses encryption and policies to keep data secure.
  + Flexibility: Allows both live and scheduled data refresh options.

1. **How would you approach this problem, if the objective and subjective questions weren't given?**
   * **Guidelines**

* To approach the problem without predefined questions, I would:
* **Understand Data Structure**: Analyze tables (customers, orders), relationships, and key metrics.
* **Data Cleaning**: Handle missing values, correct data types, and remove duplicates.
* **Data Modeling**: Create relationships, design key DAX measures (e.g., total sales, revenue growth).
* **Analyze Insights**: Segment customers, analyze product/category performance, and revenue distribution.
* **Visualize Data**: Create dashboards for sales, customer insights, and product performance.
* **Scenario Analysis**: Identify low-performing products, suggest promotional strategies, and design loyalty programs.
* **Reporting**: Generate reports with actionable insights and strategic recommendations for improving sales and customer engagement.

**Report:**

The stakeholders have asked for three tabs in the Reporting:

* Main Tab
* Product Tab
* Individual product Tab
* Using the Main tab in the Report, Stakeholders should be able to look at the total sales, revenue produced till now, and customer satisfaction for all product categories. This tab should have a slicer of product category and date.
* Using the product Tab, the management at the company should be able to look at the individual product category-wise performance like customer satisfaction, the number of customers purchasing that product, and which product requires improvement. This tab should have a slicer of gender, delivery type, product category, and age group
* Using the Individual product Tab, the Management and Stakeholder in the company want to look at the best product profile which would involve their total purchase, ID of that product, category/subcategory of that product, average rating, and how much revenue was generated by this product. All the metrics using which they can address the product very carefully in their orders.

**Make sure that all the visualizations look decent and are placed in a proper order. Each tab has different POCs (Point Of Contact), so make sure you involve all the metrics that POC may look at in that tab along with those mentioned in the tab description.**