# Supply Chain Analysis Report

Project Title: Supply chain Data Analysis and Visualization

Tools: A Comprehensive Analysis Using SQL, Pandas and Power BI

Submitted by: Mumtaj

**🔹 Introduction**

Supply chain data holds valuable information that can help companies make better decisions about product performance, supplier reliability, shipping delays, and overall profitability. This project aims to explore and analyze a supply chain dataset using three powerful tools:

* **SQL** for cleaning and preparing the data,
* **Pandas (Python)** for deep analysis and insights,
* **Power BI** for building interactive dashboards and visual storytelling.

The dataset includes records of products, suppliers, locations, shipping details, manufacturing costs, and quality checks. The focus is to uncover trends, identify inefficiencies, and provide business recommendations based on data.

**🔹 Objective**

The main aim of this project is to analyze supply chain data using SQL, Python (Pandas), and Power BI to find useful business insights.

This includes:

1. **Cleaning the data using SQL** by fixing missing values, removing duplicates, and correcting errors.
2. **Analyzing the data using Pandas** to understand product sales, supplier performance, shipping times, and defect rates.
3. **Creating a dashboard in Power BI** to show the results in a visual and easy-to-read format.

The goal is to help improve supply chain decisions by finding out what’s working well and what needs improvement.

**🔹 About the Dataset**

The dataset used in this project contains **100 rows and 24 columns** related to different parts of a company’s supply chain.

It includes information about:

* 📦 **Products** – type, price, and number of units sold
* 🏬 **Inventory** – stock levels, availability, and order quantities
* 🚚 **Shipping** – carriers, shipping time, costs, and transport modes
* 🏭 **Manufacturing** – production volumes, manufacturing costs, and lead time
* 🧪 **Quality** – inspection results and defect rates
* 🤝 **Suppliers** – supplier name and location
* 💰 **Revenue & Cost** – revenue generated, costs, and profit (calculated)

This dataset helps us understand how well the supply chain is working — from product manufacturing to delivery — and where improvements can be made.

**Importing the Dataset into SQL**

To begin the project, the dataset (in .csv format) was imported into a MySQL database. The following steps were followed:

**1. Created the Table Structure**

First, a new table named supply\_chain was created in the SQL database with the correct column names and data types using the CREATE TABLE command.

**Step 1: Create a Table in MySQL**

CREATE TABLE supply\_chain (

id INT PRIMARY KEY AUTO\_INCREMENT,

product\_type VARCHAR(100),

sku VARCHAR(100),

units\_sold INT,

price DECIMAL (10,2),

revenue\_generated DECIMAL (12,2),

manufacturing\_costs DECIMAL (12,2),

profit DECIMAL (12,2),

supplier\_name VARCHAR (100),

location VARCHAR (100),

stock\_levels INT,

shipping\_carriers VARCHAR (100),

shipping\_costs DECIMAL (10,2),

shipping\_time INT,

transportation\_mode VARCHAR (50),

production\_volume INT,

quality\_inspection VARCHAR (100),

defect\_rates DECIMAL (5,2),

return\_rates DECIMAL (5,2),

lead\_time INT,

demand\_fluctuations VARCHAR (100),

customer\_demographics VARCHAR (100),

marketing\_channels VARCHAR (100),

order\_quantities INT

);

**Step 2: Import the CSV into MySQL**

1. Right-click your database → **Table Data Import Wizard**
2. Select your supply\_chain\_data.csv
3. Choose the supply\_chain table
4. Finish import

✅ Make sure the CSV headers match the column names.

**🔹 Data Cleaning Using SQL**

Before starting the analysis, the raw supply chain dataset was cleaned using SQL to ensure the data was accurate, consistent, and ready for analysis. The following steps were performed:

1. **Checked Table Structure**

* The DESCRIBE command was used to review the table's column names and data types.
* Ensured that numerical fields such as revenue, costs, and defect rates were stored in appropriate formats like DECIMAL.
* Query:

DESCRIBE supply\_chain;

1. **Handled Missing Values**

* Used SELECT queries to identify rows with NULL values in important columns like revenue\_generated, shipping\_costs, and manufacturing\_costs.
* Query:

SELECT \* FROM supply\_chain WHERE revenue\_generated IS NULL;

* Replaced NULLs with 0 or a default value using UPDATE statements to prevent errors during analysis.
* Query:

UPDATE supply\_chain SET revenue\_generated = 0 WHERE revenue\_generated IS NULL;

1. **Cleaned and Standardized Text**

* Applied TRIM() and LOWER() functions to remove extra spaces and standardize text in fields like:
  + supplier\_name
  + location
  + shipping\_carriers
  + customer\_demographics
* Query:

UPDATE supply\_chain

SET supplier\_name = TRIM(LOWER(supplier\_name)),

location = TRIM(LOWER(location)),

shipping\_carriers = TRIM(LOWER(shipping\_carriers));

1. **Removed Duplicates**

* Checked for duplicate sku entries using GROUP BY and HAVING.
* Query:

SELECT sku, COUNT(\*) FROM supply\_chain GROUP BY sku HAVING COUNT(\*) > 1;

* Deleted duplicate rows by comparing IDs and keeping only the first record.
* Query:

DELETE sc1 FROM supply\_chain sc1

JOIN supply\_chain sc2

ON sc1.sku = sc2.sku AND sc1.id > sc2.id;

1. **Treated Outliers**

* Identified extremely high values (e.g., shipping\_costs > 10000) that could distort results.
* Query:

UPDATE supply\_chain SET shipping\_costs = NULL WHERE shipping\_costs > 10000;

* Set such values to NULL or adjusted them as appropriate.
* Query:

UPDATE supply\_chain SET shipping\_costs = NULL WHERE shipping\_costs > 10000;

**6. Created Derived Columns**

* Added a profit column calculated as:  
  profit = revenue\_generated - shipping\_costs - manufacturing\_costs
* Query:

ALTER TABLE supply\_chain ADD COLUMN profit DECIMAL(12,2);

UPDATE supply\_chain

SET profit = revenue\_generated - shipping\_costs - manufacturing\_costs;

* Created a defect\_percentage column by converting defect rates to percentage format.
* Query:

ALTER TABLE supply\_chain ADD COLUMN defect\_percentage DECIMAL(5,2);

UPDATE supply\_chain SET defect\_percentage = defect\_rates / 100;

**7. Renamed Columns**

* Renamed columns to remove spaces and make them SQL-friendly.  
  For example:  
  Product type → product\_type,  
  Revenue generated → revenue\_generated.
* Query:

ALTER TABLE supply\_chain

CHANGE `Product type` product\_type VARCHAR(100),

CHANGE `Revenue generated` revenue\_generated DECIMAL(12,2);

**8. Final Validation**

* Verified the cleaned dataset using SELECT queries and checked row count consistency.
* Query:

SELECT \* FROM supply\_chain LIMIT 10;

SELECT COUNT(\*) FROM supply\_chain;

This cleaned dataset was then used for further analysis in Python (Pandas) and Power BI.

**Exploratory Data Analysis (EDA):**

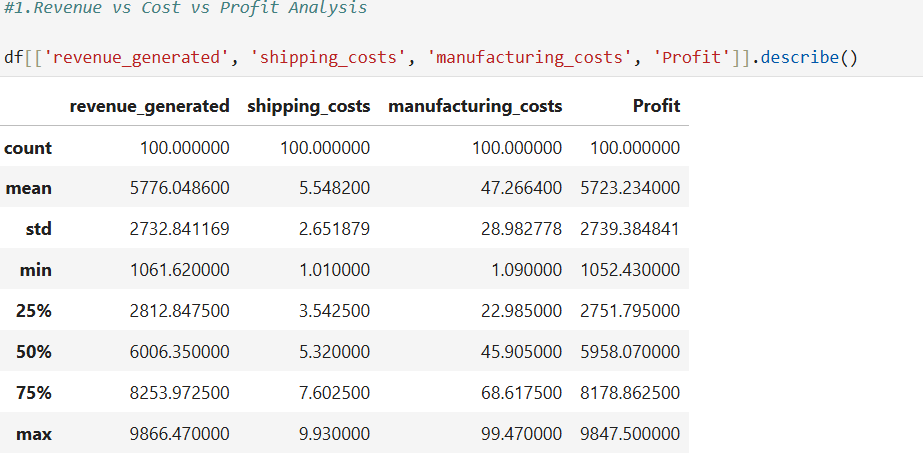
After cleaning the supply chain dataset using SQL and importing it into Pandas:

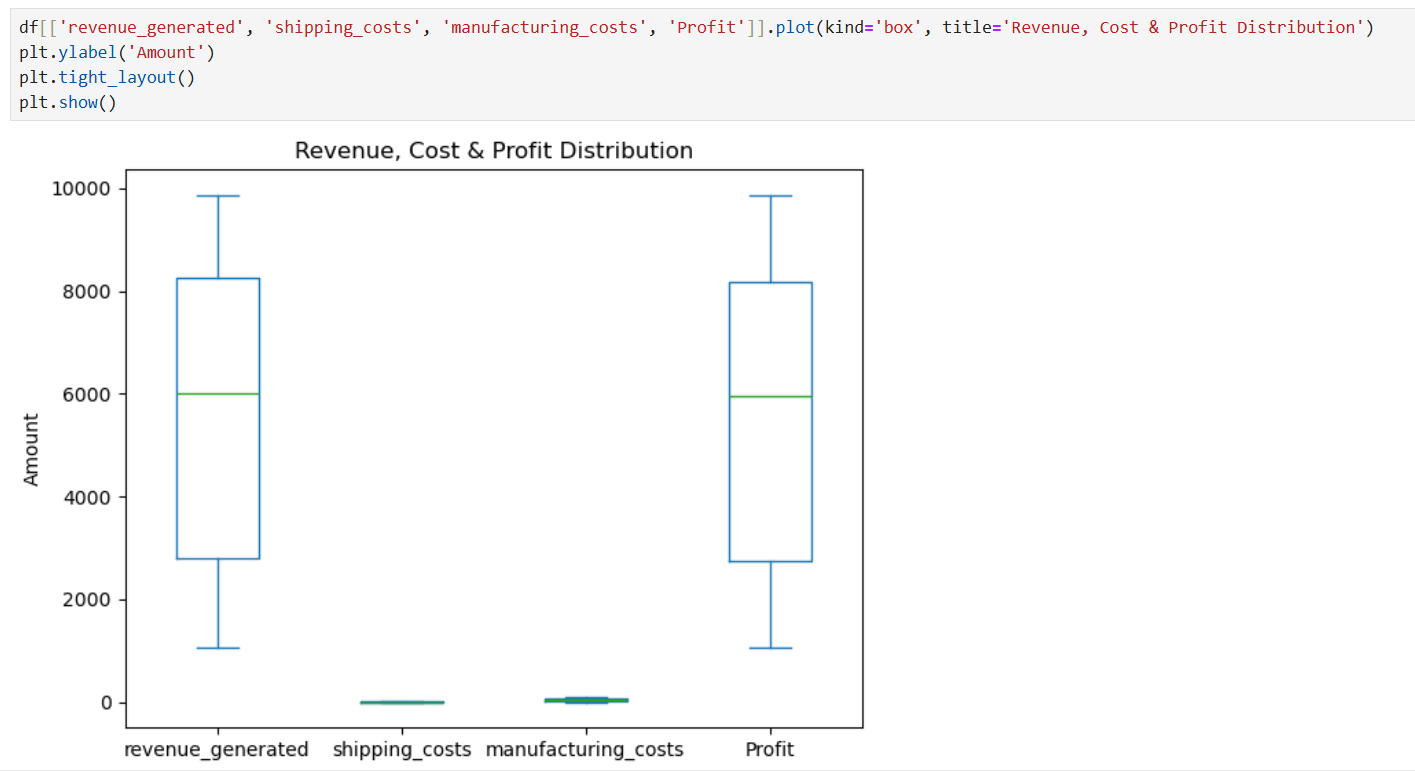




**1️⃣ Revenue vs Cost vs Profit Analysis**

**Objective**: Evaluate overall performance  
**Analysis**:

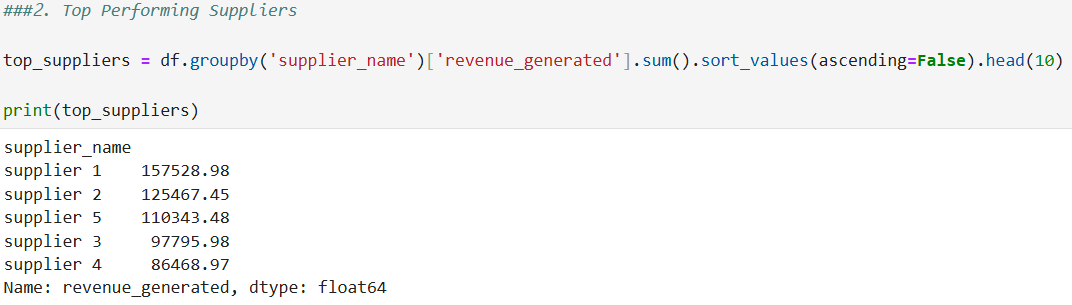


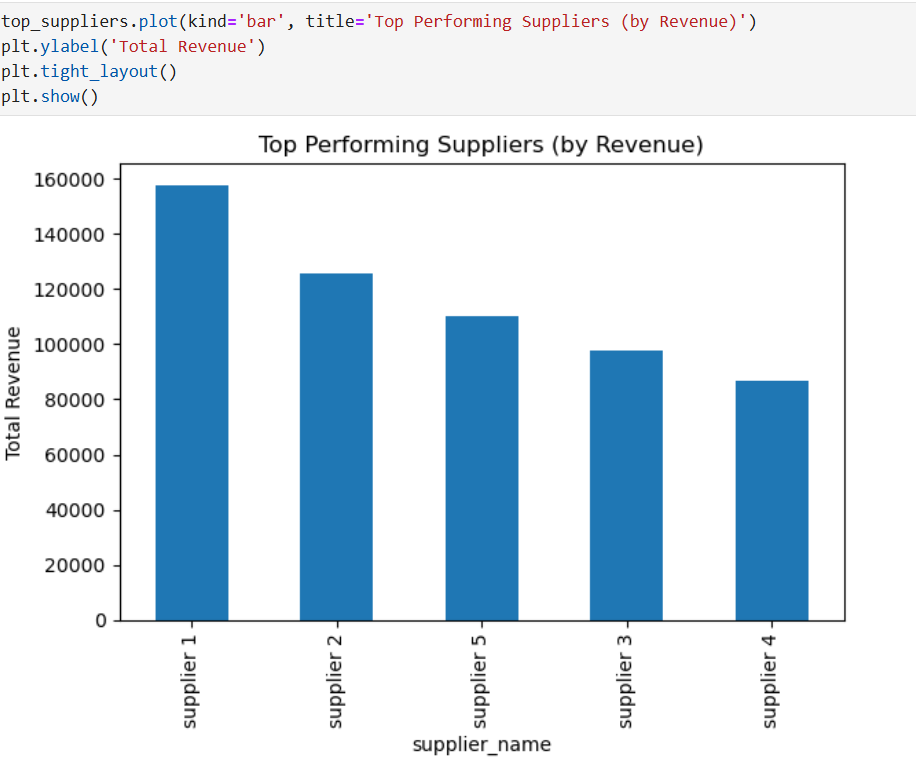


**Insight**: Find if revenue is sufficient to cover shipping + manufacturing costs, and what the profit margins are.

**2️⃣ Top Performing Suppliers**

**Objective**: Identify most valuable suppliers  
**Analysis**:

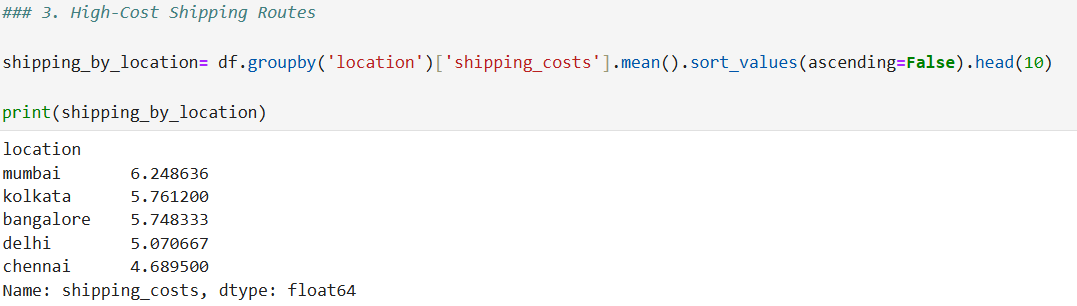


****

**Insight**: Which suppliers contribute most to revenue. Helps in vendor prioritization.

**3️⃣ High-Cost Shipping Routes**

**Objective**: Spot inefficiencies in shipping  
**Analysis**:

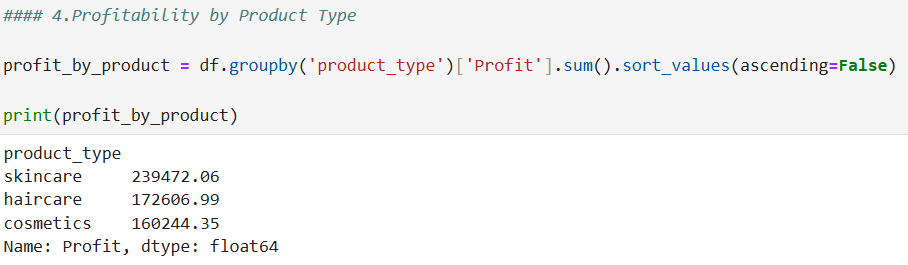


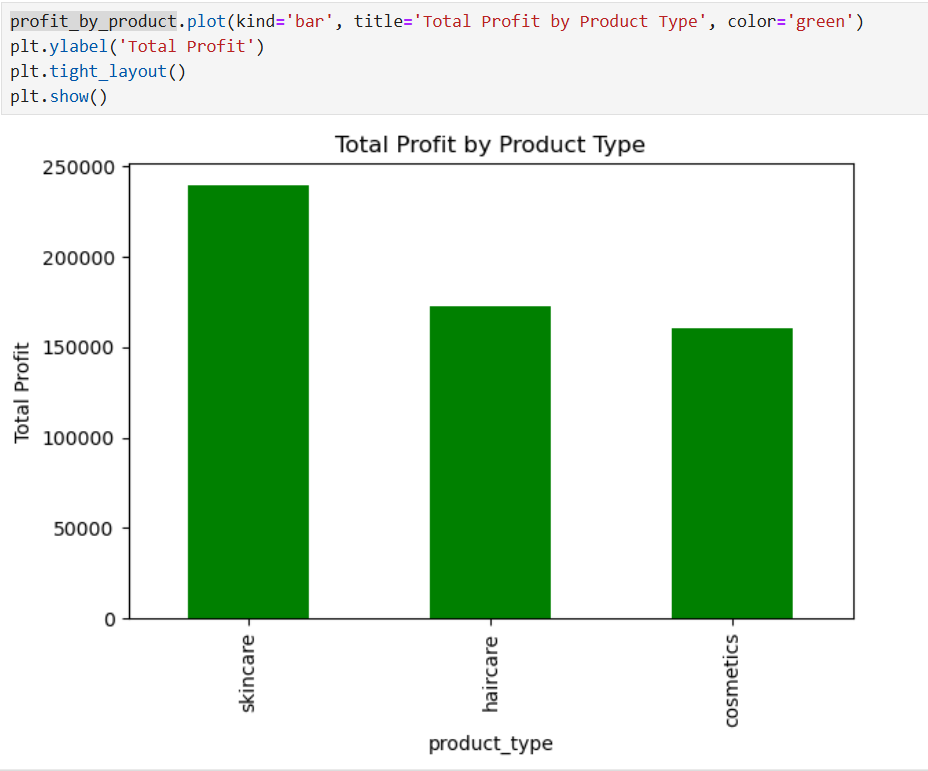
****

**Insight**: Find locations with high shipping cost, optimize routes or partners.

**4️⃣ Profitability by Product Type**

**Objective**: Know which products bring profit  
**Analysis**:

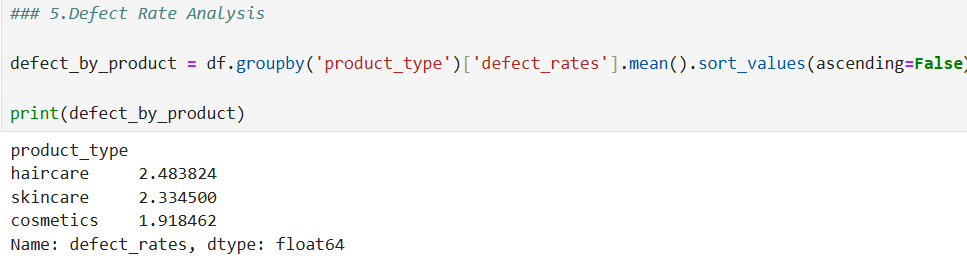


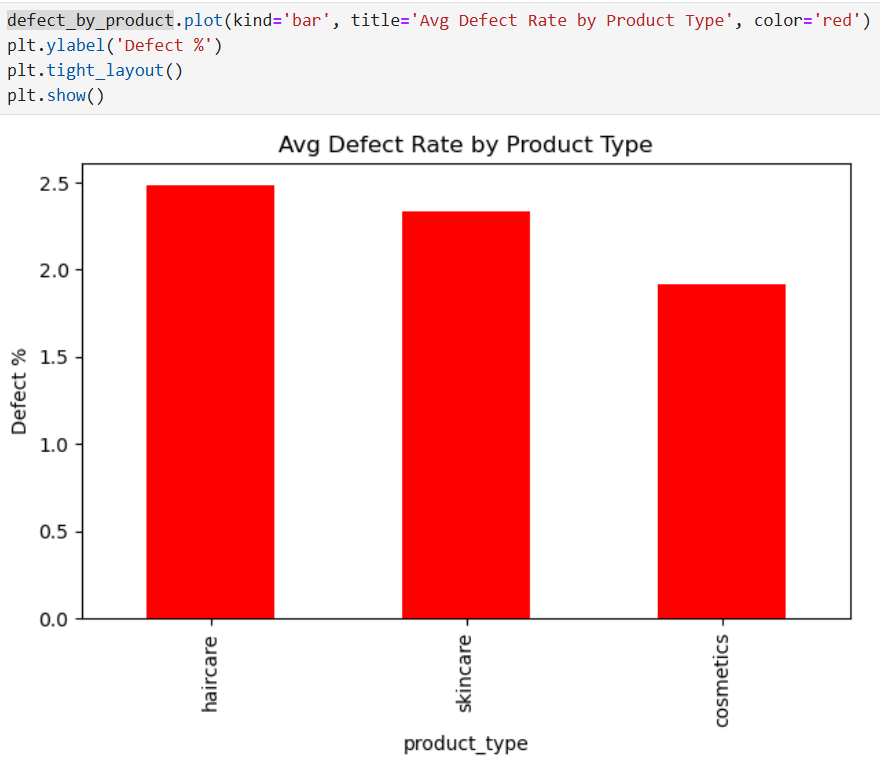
****

**Insight**: Focus on high-profit products, reduce or optimize low-profit ones.

**5️⃣ Defect Rate Analysis**

**Objective**: Identify quality issues  
**Analysis**:

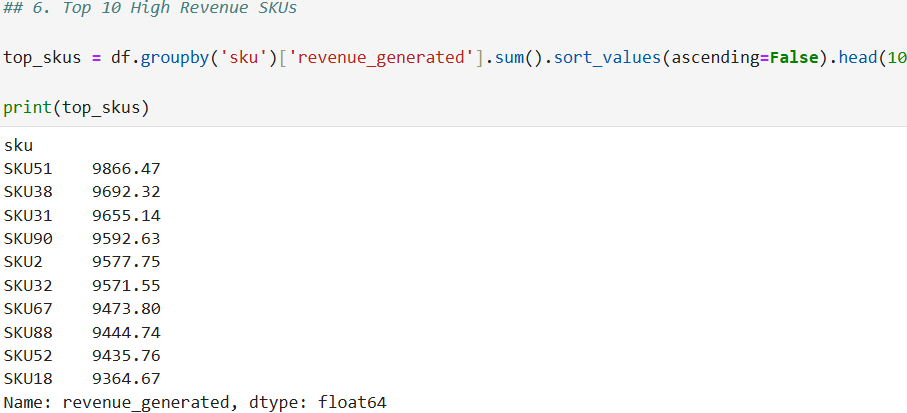


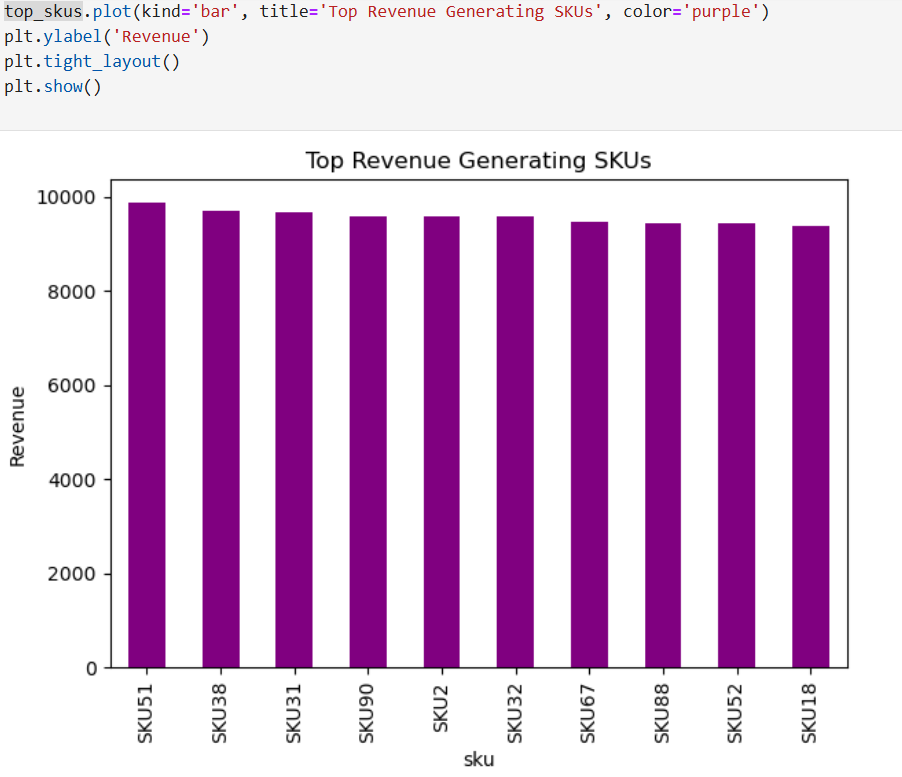
****

**Insight**: Pinpoint products with high defect rates → quality control needed.

**6️⃣ Top 10 High Revenue SKUs**

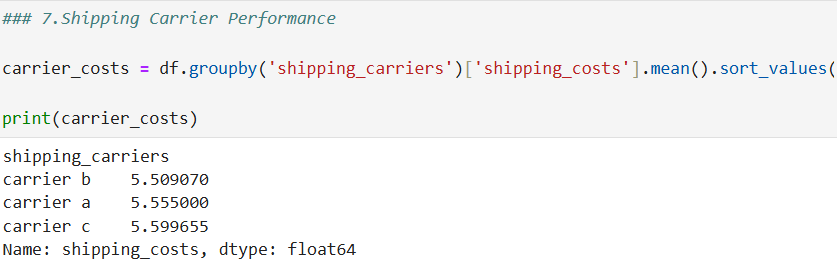
**Objective: Identify best-selling products  
Analysis:**

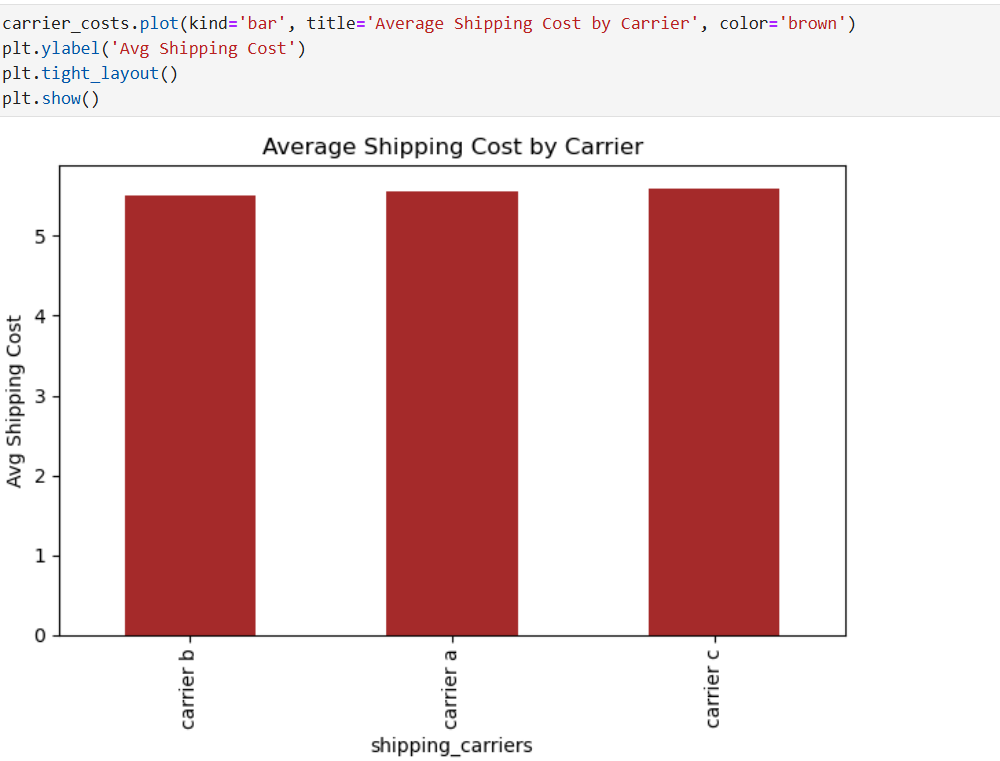
****

****

**Insight: Know which SKUs drive your business → focus on marketing/supply.**

**7️⃣ Shipping Carrier Performance**

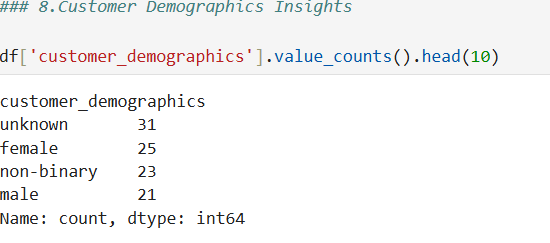
**Objective**: Evaluate cost-efficiency of carriers  
****



**Insight**: Choose the most cost-efficient shipping partners.

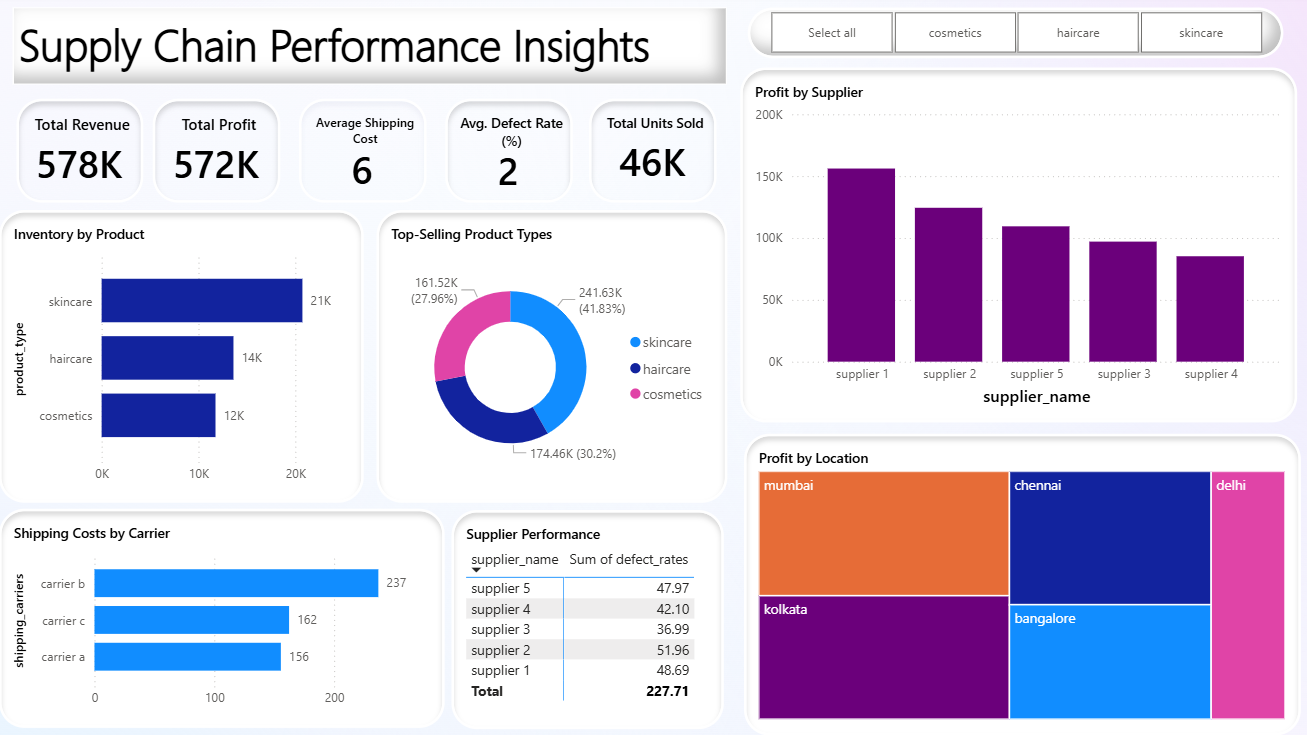
**8️⃣ Customer Demographics Insights**

**Objective**: Understand customer base (if available)  
**Analysis**:



**Insight**: Tailor products or promotions based on most frequent customer groups.

**Power BI Dashboard – Supply Chain Performance Insight**

****

This Power BI dashboard was created to analyze and monitor the performance of the supply chain. It shows important information like sales, profit, inventory, supplier performance, and shipping costs in one place. The purpose is to help make better decisions based on data.

**Data Source:**  
The data was taken from supply chain records. It was cleaned and organized before using in Power BI. Some extra columns and calculations were added to find things like profit range and defect rate.

**Main Parts of the Dashboard:**

1. **KPIs** – Show total revenue, total profit, average shipping cost, average defect rate, and total units sold.
2. **Inventory by Product** – Shows stock levels for cosmetics, haircare, and skincare.
3. **Top-Selling Product Types** – Displays which product types sell the most.
4. **Profit by Supplier** – Compares profit from different suppliers.
5. **Profit by Location** – Shows which cities give the highest profit.
6. **Shipping Costs by Carrier** – Compares shipping costs for different carriers.
7. **Supplier Performance** – Lists suppliers with their defect rates.
8. **Filters** – Allow users to view data by product category.

**Key Insights:**

* Skincare products bring the highest sales.
* Supplier 1 gives the most profit.
* Mumbai is the top city for profit.
* Carrier B has the highest shipping costs

### **Conclusion**

This project successfully analyzed the supply chain data using SQL, Python (Pandas), and Power BI. The data was first cleaned in SQL to remove errors, fill missing values, and standardize formats. Then, exploratory data analysis was done in Pandas to understand revenue, profit, shipping costs, supplier performance, and defect rates. Finally, an interactive Power BI dashboard was created to present these insights visually.

From the analysis, skincare emerged as the highest revenue-generating product type, Supplier 1 provided the most profit, Mumbai was the most profitable location, and Carrier B had the highest shipping cost. These findings can help the company improve supplier selection, reduce logistics costs, and optimize inventory.

Overall, the combination of SQL, Pandas, and Power BI provided a complete workflow — from data cleaning to analysis and visualization — making it easier for decision-makers to track performance and take timely actions.