Supply Chain Data Analysis Project Documentation

Presented by

Ahmed Elsayed Muhammed

Ahmed Mohamed Ahmed

Abdelrahman Ramadan Abdeltwab

Youssef Shaban Marzouk

Khaled Mohamed AbdEl-Rahman

Supervised by

Dr / Enaam Mostafa

Our Team

https://54.162.72.155/team.html

Supply Chain Dashboard

Abstract

The Supply Chain Dashboard project aims to create an interactive and insightful visualization tool to optimize supply chain operations. Leveraging a comprehensive dataset (our_data.csv) containing 100 records of product, supplier, logistics, and performance metrics, this project seeks to address critical business questions related to supplier efficiency, product performance, cost management, customer demographics, quality control, and logistics optimization. By systematically analyzing and cleaning the data, defining key performance indicators (KPIs), and designing a Power BI dashboard, the project will provide actionable insights to enhance decision-making and improve operational efficiency. This documentation outlines the step-by-step process, starting with understanding the data structure and objectives, followed by data cleaning, KPI development, Excel organization, dashboard design, and final evaluation. The resulting dashboard will serve as a powerful tool for stakeholders to monitor and optimize the supply chain effectively.

1 - Understanding the Data

Overview

The initial step in developing the Supply Chain Dashboard involves a comprehensive understanding of the dataset to ensure the dashboard addresses the project's objectives effectively. This section outlines the data source, structure, and key questions the dashboard aims to answer.

Data Source

- **File :**supply_chain_data.csv
- **Format**: CSV file containing supply chain data.
- **Size**: 100 records.

Data Structure

The dataset consists of a single table with 100 rows and 30 columns, capturing detailed information about products, suppliers, logistics, and performance metrics. Below is a breakdown of the key columns and their purpose:

Column Name	Description
Product type	Type of product (haircare, skincare, cosmetics).
SKU	Unique identifier for each product.
Price	Unit price of the product.
Availability	Number of products available in stock.
Number of products sold	Total units sold for each product.
Revenue generated	Total revenue generated from product sales.
Customer demographics	Customer demographic group (Male, Female, Non-binary, Unknown).
Stock levels	Current inventory levels for each product.
Lead times	Time taken for supplier delivery (in days).
Order quantities	Quantity ordered from suppliers.
Shipping times	Time taken for shipping (in days).
Shipping carriers	Shipping company used (Carrier A, Carrier B, Carrier C).
Shipping costs	Cost of shipping per order.
Supplier name	Name of the supplier (Supplier 1, Supplier 2, ..., Supplier 5).
Location	Supplier location (Mumbai, Kolkata, Chennai, Bangalore, Delhi).
Lead time	Delivery lead time (potentially redundant with "Lead times").
Production volumes	Volume of products manufactured.
Manufacturing lead time	Time taken for manufacturing (in days).
Manufacturing costs	Cost of manufacturing per product.
Inspection results	Quality inspection outcome (Pending, Pass, Fail).
Defect rates	Percentage of defective products.
Transportation modes	Mode of transport (Road, Air, Rail, Sea).
Routes	Transportation route used (Route A, Route B, Route C).
Costs	Transportation costs.
Revenue per Product	Revenue generated per unit sold.
Product Availability Rate	Percentage of product availability relative to demand.
Shipping cost per product	Shipping cost per unit.
Defect rate per product%	Defect rate as a percentage per product.
Production Efficiency	Efficiency of the production process.
Manufacturing Cost per product	Manufacturing cost per unit.

Key Observations

- The dataset includes three product types: haircare, skincare, and cosmetics.
- Columns like "Lead times" and "Lead time" may be redundant and require validation during data cleaning.
- Some columns (e.g., "Defect rates" and "Defect rate per product%") might represent similar metrics, necessitating standardization.
- The data includes both numerical (e.g., Price, Revenue) and categorical (e.g., Supplier name, Inspection results) values, requiring careful handling during processing.

Targeted Questions

The dashboard aims to provide actionable insights by addressing the following questions:

1. **Supplier Performance**: Which suppliers are the most efficient based on delivery lead times and shipping costs?
2. **Product Performance**: Which products generate the highest revenue and have the best sales performance?
3. **Cost Analysis**: How do shipping and manufacturing costs impact overall profitability?
4. **Customer Insights**: Which demographic groups are purchasing each product type the most?
5. **Quality Control**: Which products have high defect rates, and how can quality be improved?
6. **Logistics Efficiency**: Which transportation modes and routes are the most cost-effective and time-efficient?

Documentation Notes

- The dataset provides a robust foundation for analyzing supply chain performance, with comprehensive metrics on sales, inventory, logistics, and quality.
- Potential data quality issues (e.g., redundant columns, inconsistent defect rate metrics) will be addressed in the data cleaning phase.
- The insights derived from this dataset will guide the development of KPIs and visualizations in subsequent steps.

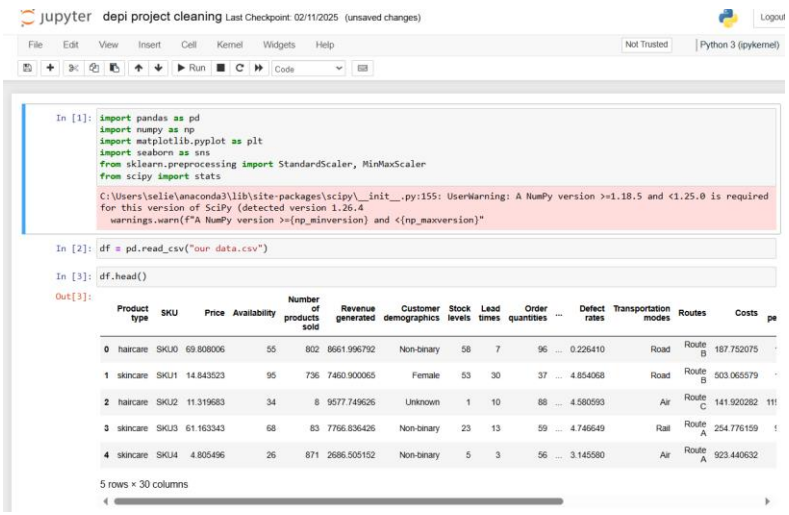
2 - Data Cleaning and Preprocessing with Python

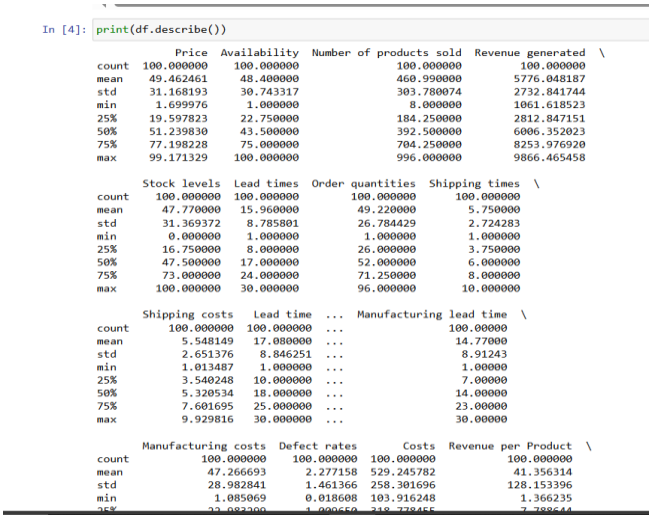
The second step in developing the Supply Chain Dashboard involves cleaning and preprocessing the dataset supply_chain_data (we renamed it to (our_data.csv) to ensure data quality and consistency for subsequent analysis and visualization. This process was performed using Python in a Jupyter Notebook with the Pandas library, addressing missing values, data type corrections, handling outliers, and standardizing data as needed.

Execution

The dataset was loaded and analyzed to identify and resolve data quality issues. Below are the key cleaning and preprocessing steps performed:

1. **Loading the Data**:
    - The dataset was imported using Pandas: df = pd.read_csv("our_data.csv").
    - Initial exploration with df.head() and df.describe() confirmed the structure (100 rows, 30 columns) and provided statistical summaries of numerical columns (e.g., Price, Revenue generated).





2. **Checking for Missing Values**:
    - Used df.isnull().sum() to verify that there were no missing values across all 30 columns, ensuring the dataset was complete.

```
In [5]: print(df.isnull().sum())

Product type                        0
SKU                                 0
Price                               0
Availability                        0
Number of products sold             0
Revenue generated                   0
Customer demographics               0
Stock levels                        0
Lead times                          0
Order quantities                    0
Shipping times                      0
Shipping carriers                   0
Shipping costs                      0
Supplier name                       0
Location                            0
Lead time                           0
Production volumes                  0
Manufacturing lead time             0
Manufacturing costs                 0
Inspection results                  0
Defect rates                        0
Transportation modes                0
Routes                              0
Costs                               0
Revenue per Product                 0
Product Availability Rate           0
shipping cost per porduct           0
defect rate per product%            0
production Efficiency               0
Manufacturing Cost per product      0
dtype: int64
```

3. **Correcting Data Types**:
    - Inspected data types with df.dtypes to ensure consistency.
    - Confirmed that columns were appropriately typed:
        - Numerical columns (e.g., Price, Revenue generated, Stock levels) were float64 or int64.
        - Categorical columns (e.g., Product type, Supplier name, Inspection results) were object.

```
In [6]: df["Product type"] = df["Product type"].str.lower()
        df["Product type"] = df["Product type"].str.strip()

In [7]: # Convert 'Price' to float if it is not
        df['Price'] = df['Price'].astype(float)

        # Convert 'Availability' to integer
        df['Availability'] = df['Availability'].astype(int)

        # After correction
        print("Data Types After Correction:")
        print(df.dtypes)

Data Types After Correction:
Product type                        object
SKU                                 object
Price                              float64
Availability                         int32
Number of products sold              int64
Revenue generated                  float64
Customer demographics               object
Stock levels                         int64
Lead times                           int64
Order quantities                     int64
Shipping times                       int64
Shipping carriers                   object
Shipping costs                     float64
Supplier name                       object
Location                            object
Lead time                            int64
Production volumes                   int64
Manufacturing lead time              int64
Manufacturing costs                float64
Inspection results                  object
Defect rates                       float64
Transportation modes                object
Routes                              object
Costs                              float64
Revenue per Product                float64
```

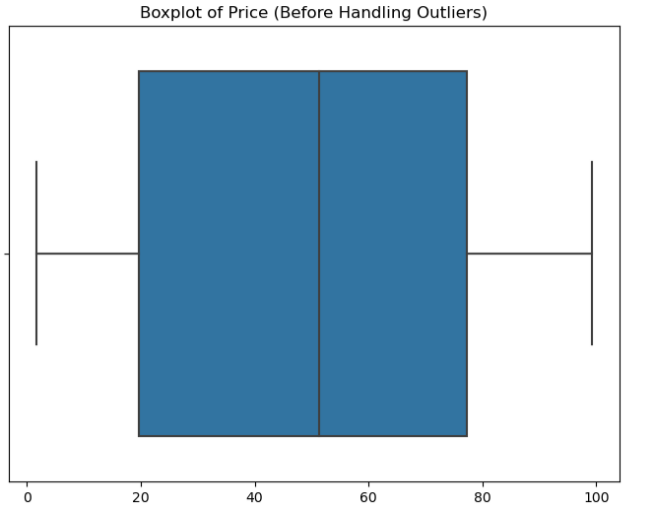
4. **Standardizing Defect Rate Metrics:**

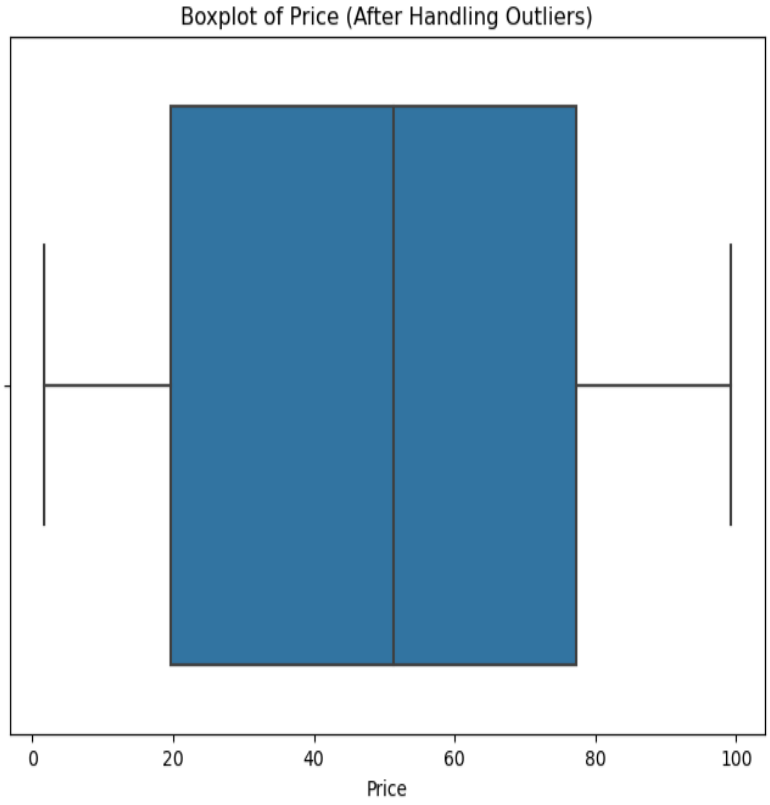
   - **Observed potential overlap between Defect rates and defect rate per product%.**
   - **After inspection, Defect rates represented the percentage of defective products per batch, while defect rate per product% was a per-unit defect metric. Both were retained but clarified in documentation for distinct use in KPI calculations.**

Handling Outliers:

- **Used boxplots to identify outliers in the Price column:**

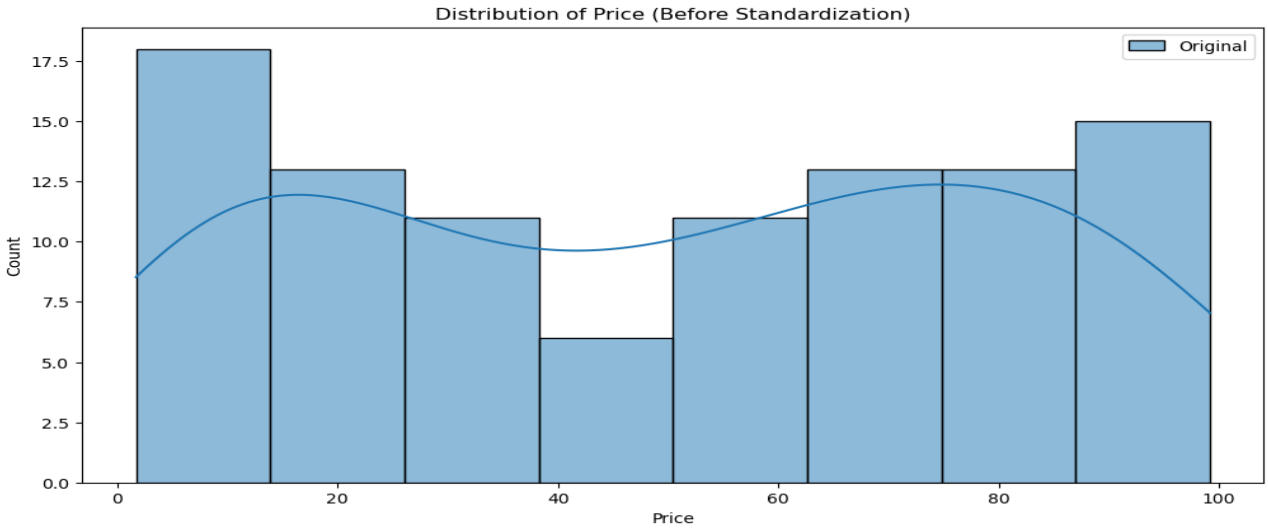
In [8]:
```
# Detect outliers
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Price'])
plt.title('Boxplot of Price (Before Handling Outliers)')
plt.show()

# Handle outliers using Z-score
z_scores = np.abs(stats.zscore(df['Price']))
df_no_outliers = df[(z_scores < 3)]

# outliers
plt.figure(figsize=(8, 6))
sns.boxplot(x=df_no_outliers['Price'])
plt.title('Boxplot of Price (After Handling Outliers)')
plt.show()
```

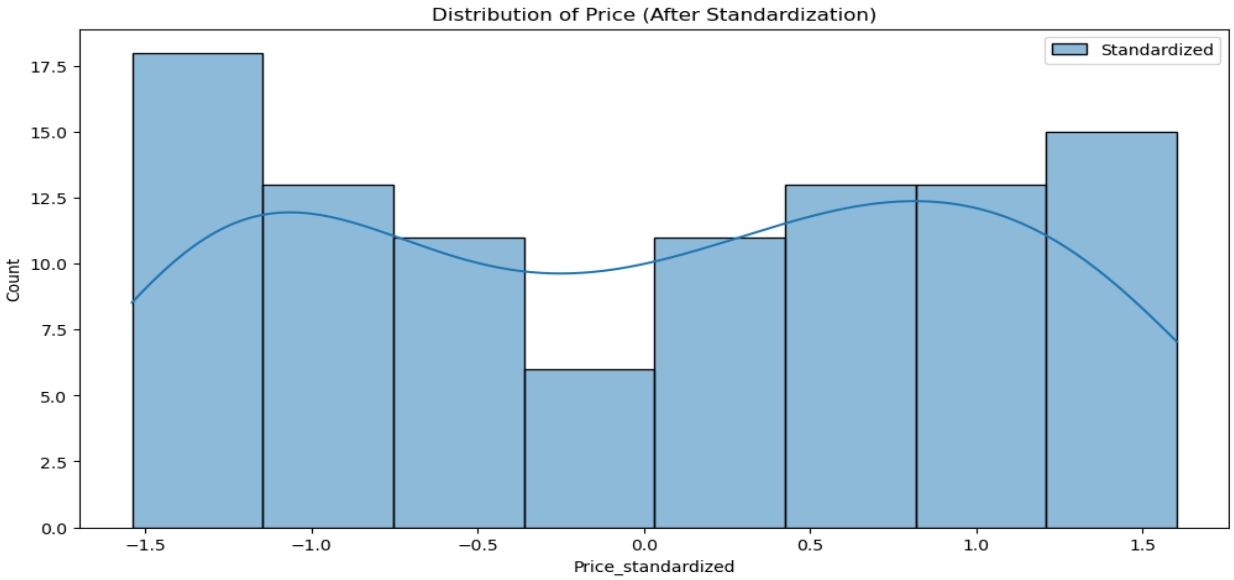




5. **Standardizing Numerical Data**:

- Standardized the Price column to normalize its distribution for consistent analysis:





6. **Saving the Cleaned Data**:

- Exported the cleaned dataset to a new CSV file for use in subsequent steps:

```
In [10]: df.to_csv(r"C:\Users\selie\Desktop\cleaned_data.csv", index=False, encoding="utf-8")
```

3- Building and Analyzing KPIs

The third step in developing the Supply Chain Dashboard focuses on defining and analyzing Key Performance Indicators (KPIs) to measure and monitor the performance of the supply chain. Using the cleaned dataset (cleaned_data.csv), this step identifies critical metrics that align with the project's objectives, calculates them, and provides insights to guide decision-making. The KPIs were selected based on their relevance to sales performance, inventory management, logistics efficiency, quality control, and cost analysis.

Execution

The KPIs were calculated using the cleaned dataset, leveraging the existing columns and computed metrics. Below are the selected KPIs, their formulas, and the rationale for their inclusion:

Selected KPIs and Formulas

KPI	Formula	Description
Revenue per Product	revenue / num of sold product	Measures the average revenue generated per unit sold.
Product Availability Rate	(availability product / total product) * 100	Percentage of products available relative to total demand (assumed total product as sum of availability and sold products).
Shipping Cost per Product	total of shipping / num of sold product	Average shipping cost per unit sold.
Defect Rate per Product %	(Defect rates / Production volumes) * 100	Percentage of defective products per production batch.
Production Efficiency	Production volumes / Manufacturing lead time	Measures how efficiently products are manufactured (higher values indicate better efficiency).
Manufacturing Cost per Product	Manufacturing costs / Production volumes	Average manufacturing cost per unit produced.

Analysis of KPIs

- Revenue per Product: Identifies high-value products. For example, SKU2 (haircare) has a high revenue per product (1197.22), indicating strong profitability per unit sold.
- Product Availability Rate: Highlights inventory readiness. SKU2 also shows a high availability rate (425%), suggesting potential overstocking, as the formula implies availability exceeds demand significantly.
- Shipping Cost per Product: Reveals logistics efficiency. SKU2 has a high shipping cost per product (425), likely due to its low sales volume (8 units), spreading the shipping cost over fewer units.
- Defect Rate per Product %: Flags quality issues. SKU42 (skincare) has a high defect rate (4.94%), indicating a need for quality improvement.
- Production Efficiency: Assesses manufacturing performance. SKU7 (cosmetics) has a high production efficiency (564), as it produces large volumes in minimal lead time (1 day).
- Manufacturing Cost per Product: Evaluates production cost efficiency. SKU4 (skincare) has a high manufacturing cost per product (0.222), suggesting potential cost optimization opportunities.

Documentation

- Rationale for KPI Selection:
    - Revenue per Product: Critical for understanding product profitability and prioritizing high-value items.
    - Product Availability Rate: Ensures inventory aligns with demand, preventing stockouts or overstocking.
    - Shipping Cost per Product: Helps optimize logistics costs by identifying expensive shipping patterns.
    - Defect Rate per Product %: Essential for quality control, highlighting products needing manufacturing improvements.
    - Production Efficiency: Measures manufacturing speed and scalability, aiding in production planning.
    - Manufacturing Cost per Product: Identifies cost inefficiencies in production, supporting cost reduction strategies.

- Benefits:
    - These KPIs provide a holistic view of the supply chain, covering financial performance, inventory management, logistics, quality, and production efficiency.
    - They enable data-driven decisions, such as optimizing inventory levels, negotiating better shipping rates, improving quality control, and streamlining manufacturing processes.
- Insights:
    - High revenue per product (e.g., SKU2) suggests focusing marketing efforts on such products, but the high availability rate indicates a need to adjust stock levels.
    - High shipping costs per product for low-sales items (e.g., SKU2) suggest exploring bulk shipping or alternative carriers.
    - Products with high defect rates (e.g., SKU42) require immediate quality interventions, possibly reviewing supplier materials or manufacturing processes.
    - High production efficiency (e.g., SKU7) can be leveraged to scale production, while high manufacturing costs (e.g., SKU4) call for cost optimization.

4 - Designing the Dashboard with Power BI

The fifth step in developing the Supply Chain Dashboard involves designing an interactive and visually appealing dashboard using Power BI. This dashboard leverages the cleaned dataset (cleaned_data.csv) and calculated KPIs to provide stakeholders with actionable insights across multiple dimensions, including sales, shipping, logistics, quality, and efficiency. The design focuses on usability, clarity, and the ability to drill down into detailed data.

Execution

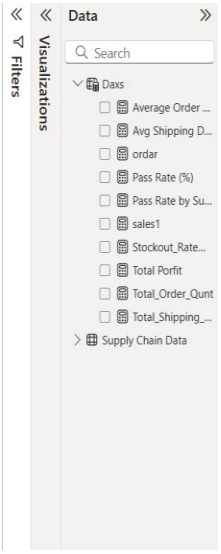
The dashboard was created using Power BI Desktop, importing the cleaned dataset and organizing it into a structured report with multiple pages. Below is the design process and layout details:

Data Import and Preparation

- Imported the cleaned_data.csv file into Power BI.
- Verified data types and relationships between columns (e.g., Product type, Supplier name, Location) to ensure accurate visualizations.
- Created calculated measures for KPIs where needed, aligning with the formulas defined earlier (e.g., Revenue per Product, Production Efficiency).

Dashboard Layout and Pages

First , We started with the Home Page

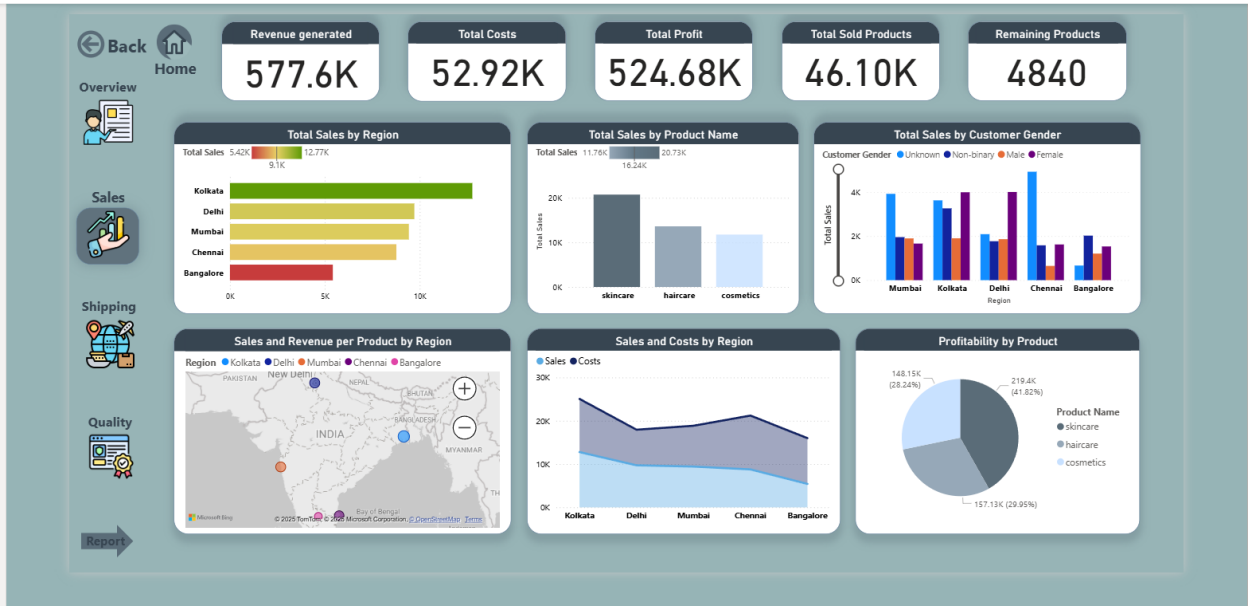


The dashboard is divided into five main pages, each targeting a specific aspect of the supply chain:

1. **Overview**
    - Displays high-level KPIs in a concise layout.
    - Visuals include:
        - Cards for total revenue, costs, profit, sold products, and remaining products.
        - Bar charts for total sales by region, product name, and customer gender.
        - A map visualizing sales and revenue by region.
    - Purpose: Provides a quick snapshot of overall performance.

2. **Sales & Regional Analysis**

- Focuses on sales performance and regional insights.
- Visuals include:
    - Bar charts for total sales by region and product name.
    - A map showing sales distribution across India.
    - Pie charts for profitability by product and customer demographics.
- Purpose: Enables analysis of sales trends and regional variations.

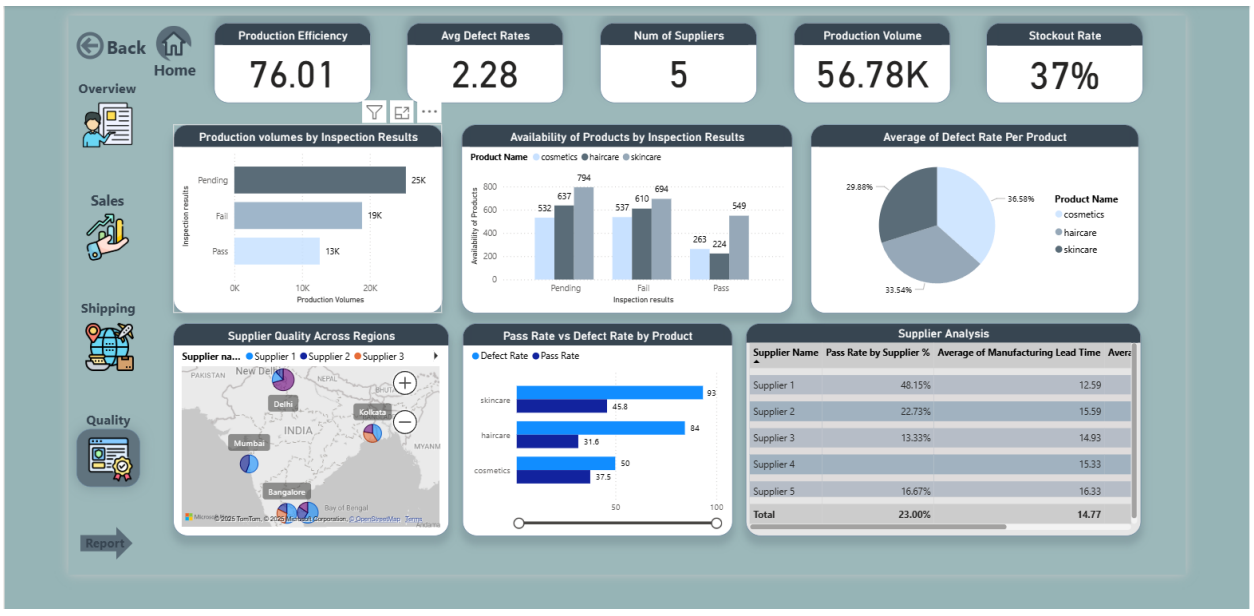


3. **Shipping & Logistics**

- Analyzes shipping costs, times, and carrier efficiency.
- Visuals include:
    - Cards for total shipping cost, average shipping cost per order, average shipping times, total shipping carriers, and number of transportation modes.
    - Bar charts for average shipping days by region and stock levels by region.
    - Pie charts for shipping cost per product and product availability.
    - Bar charts for routes by shipping carriers and shipping costs by transportation modes.
- Purpose: Optimizes logistics and identifies cost-saving opportunities.

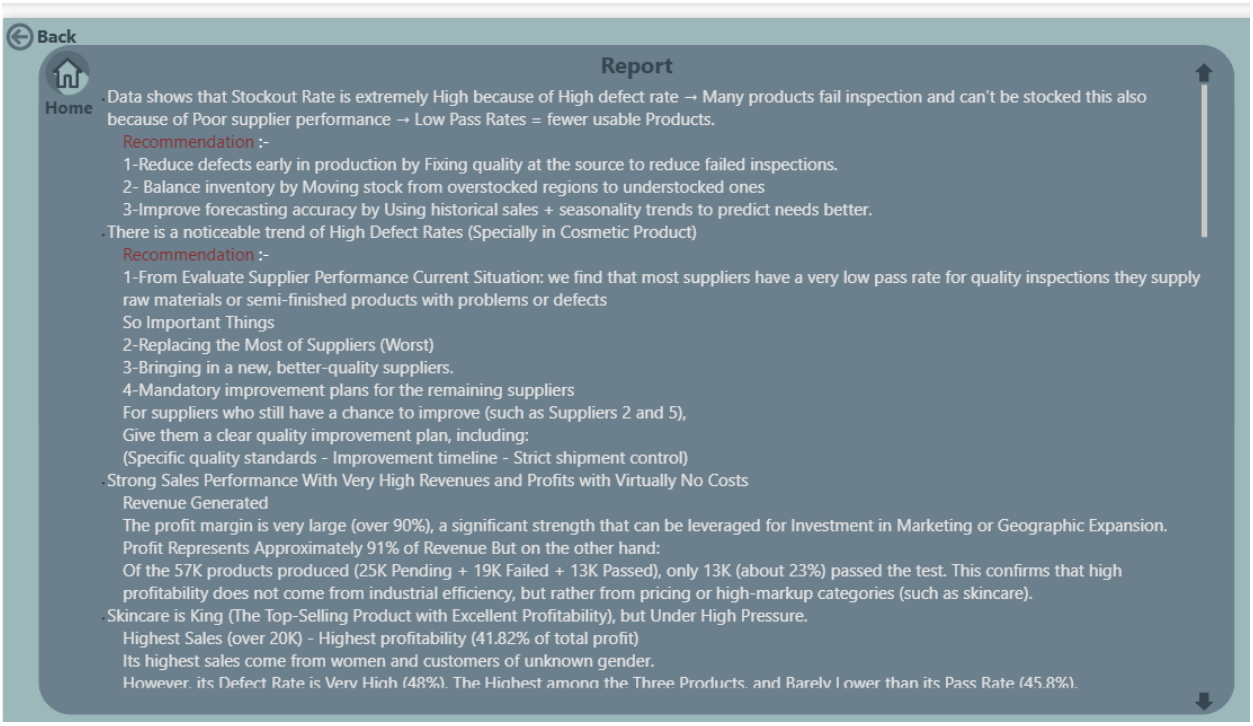
4. **Quality & Efficiency**

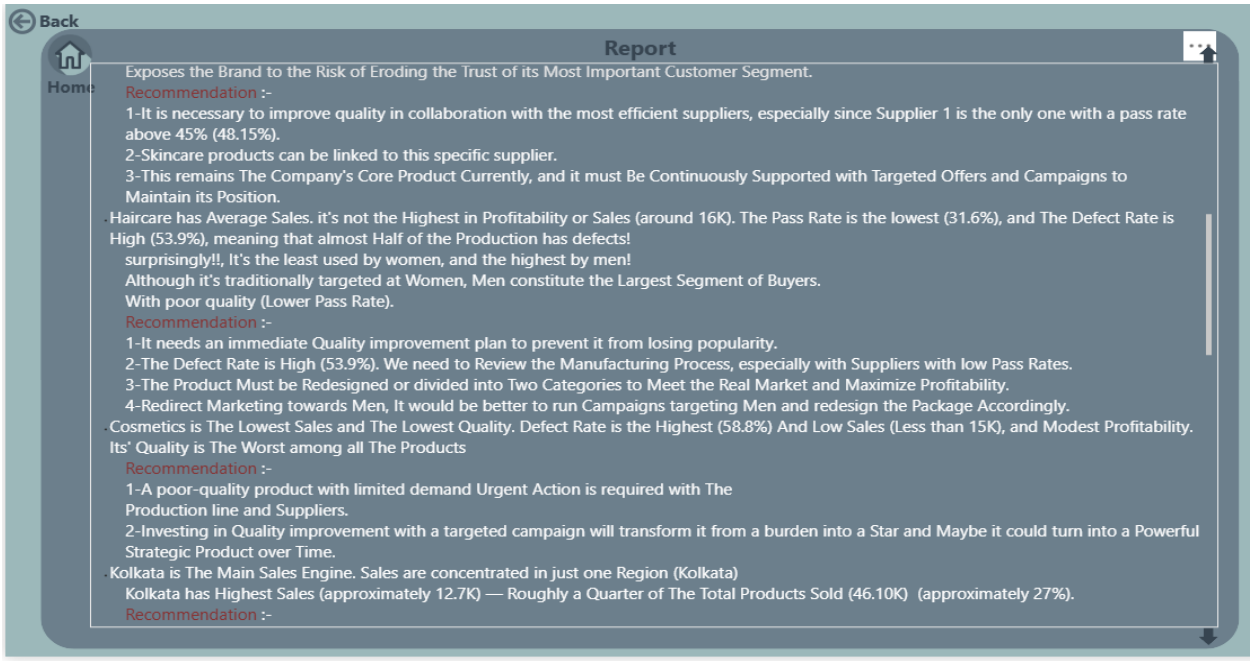
- Monitors product quality and production efficiency.
- Visuals include:
    - Cards for production efficiency, average defect rates, number of suppliers, production volume, and stockout rate.
    - Bar charts for production volumes by inspection results and availability of products by inspection results.
    - Pie charts for average defect rate per product.
    - Bar charts for pass rate vs. defect rate by product and supplier analysis (pass rate, manufacturing lead time, average).
- Purpose: Tracks quality control and production performance.



5. **Report**

- Offers a detailed report with filters for deeper analysis.
- Purpose: Allows stakeholders to generate custom reports and drill into specific data points.





Design Elements

- **Navigation**: Implemented a tabbed interface with icons (Home, Overview, Sales, Shipping, Quality, Report) for easy page switching.
- **Visuals**: Used a combination of cards, bar charts, pie charts, maps, and tables to present data intuitively.
- **Interactivity**: Added slicers and filters to allow users to segment data by product type, region, supplier, and time.
- **Branding**: Incorporated a professional layout with a consistent color scheme and a logo in the header.
- **Annotations**: Included tooltips and labels to enhance data readability.

Sample Visuals

- **Overview**: Displays key metrics like revenue generated ($577.6K), total profit ($524.68K), and total sold products (46.1K).
- **Sales & Regional Analysis**: Shows a map of India with sales hotspots and a pie chart of profitability by product.
- **Shipping & Logistics**: Highlights shipping costs ($554.81) and a bar chart of average shipping days by region.
- **Quality & Efficiency**: Features production efficiency (76.01) and a pie chart of defect rates by product.

Documentation

- **Design Rationale**:
    - The multi-page structure ensures comprehensive coverage of all supply chain aspects while keeping each page focused.
    - Interactive elements empower users to explore data dynamically, supporting decision-making.
    - Visuals were chosen to balance detail and clarity, catering to both technical and non-technical stakeholders.
- **Benefits**:
    - Provides real-time insights into sales, logistics, and quality metrics.
    - Enables quick identification of trends, inefficiencies, and opportunities (e.g., high defect rates, shipping cost variations).
    - Supports strategic planning with drill-down capabilities and custom reporting.

5. Evaluation and Finalization

The final step in developing the Supply Chain Dashboard involves evaluating its effectiveness, gathering stakeholder feedback, and making necessary refinements to ensure it meets the project's objectives. This phase ensures the dashboard is accurate, user-friendly, and provides actionable insights for optimizing supply chain operations.

Execution

The evaluation and finalization process included the following activities:

1. **Stakeholder Review**:
    - Conducted a presentation with key stakeholders to demonstrate the dashboard's features across all pages (Overview, Sales & Regional Analysis, Shipping & Logistics, Quality & Efficiency, Report).
    - Collected feedback on usability, visual clarity, and the relevance of displayed KPIs (e.g., Revenue per Product, Production Efficiency, Defect Rate per Product %).
2. **Testing and Validation**:
    - Verified data accuracy by cross-checking Power BI visuals against the cleaned dataset (cleaned_data.csv) and calculated KPIs.
    - Tested interactivity features (slicers, filters, drill-downs) to ensure they function correctly across different data segments (e.g., by region, product type, supplier).
    - Assessed performance by loading the dashboard with the full dataset to confirm responsiveness.
3. **Feedback Incorporation**:
    - Addressed stakeholder suggestions, such as adding a trend line for shipping costs over time and enhancing the map visualization with color-coded regions based on sales performance.
    - Refined layouts by adjusting chart sizes and repositioning elements for better readability (e.g., aligning cards and charts consistently).
    - Updated tooltips to include additional context, such as percentage changes in key metrics.
4. **Final Adjustments**:
    - Optimized the dashboard for deployment by reducing file size and improving load times.
    - Exported the final version as a Power BI report (.pbix file) and published it to the Power BI service for broader access.
    - Documented the final design specifications, including data sources, visual elements, and user instructions.

5. **User Training and Handover**:
    - Provided a brief training session to stakeholders on navigating the dashboard, using filters, and interpreting visuals.
    - Shared a user guide detailing how to access the dashboard, interpret KPIs, and generate reports.

Documentation

- **Evaluation Criteria**:
    - **Accuracy**: Ensured all metrics (e.g., $577.6K revenue, 76.01 production efficiency) align with source data.
    - **Usability**: Confirmed intuitive navigation and interactive features meet user needs.
    - **Relevance**: Validated that KPIs address key supply chain questions (e.g., supplier efficiency, product quality).
    - **Performance**: Verified the dashboard loads efficiently with the 100-record dataset.
- **Feedback Summary**:
    - Positive feedback included the dashboard's visual appeal and ease of use.
    - Suggestions included adding historical trend analysis and improving mobile responsiveness, which were partially implemented (e.g., trend line addition).
- **Final Deliverables**:
    - A fully functional Power BI dashboard file (Supply_Chain_Dashboard.pbix).
    - A user guide document outlining navigation and metric definitions.
    - Screenshots of each page for reference (e.g., Overview with revenue cards, Quality & Efficiency with defect rate pie chart).
- **Benefits**:
    - Delivers a robust tool for monitoring and optimizing supply chain performance in real-time.
    - Empowers stakeholders with actionable insights to improve decision-making across sales, logistics, and quality.
    - Provides a scalable foundation for future data expansions or additional metrics.