Pragati Superstore

Data Driven Decision Making (DDDM)
&
Sales Performance Dashboard

Roman Shrestha

Table of Contents

Introduction:

The Pragati Superstore is a fictional superstore which is struggling to compete with Amazen and World Mart because of a limited understanding of their sales core insights. where the grocery sales dataset (MBA, 2023) from the Kaggle datasets is assumed to as this company's dataset to develop and analyse a decision support system. To solve the problem, there are seven high-level normalised CSV-formatted datasets for four-month periods, including product categories, customer details, employee information, product details, and sales information in different cities and countries, provided as open-source datasets for exploratory data analysis. The datasets, schemas, and tables are as follows.

Database Schema

The dataset consists of seven interconnected tables:

File Name	Description
`categories.csv`	Defines the categories of the products.
`cities.csv`	Contains city-level geographic data.
`countries.csv`	Stores country-related metadata.
`customers.csv`	Contains information about the customers who make purchases.
`employees.csv`	Stores details of employees handling sales transactions.
`products.csv`	Stores details about the products being sold.
`sales.csv`	Contains transactional data for each sale.

Figure 1: Interconnected Tables (MBA, 2023)

This study uses seven CSV datasets for exploratory data analysis and dashboard building, whose schema and description are mentioned in Figure 1.

Table Descriptions

1. categories

Key	Column Name	Data Type	Description
PK	`CategoryID`	`INT`	Unique identifier for each product category.
	`CategoryName`	`VARCHAR(45)`	Name of the product category.

Figure 2: Interconnected Tables (MBA, 2023)

Cateogeris datasets have 11 rows and 2 columns, which provide a normalised table for category names with unique identifiers. Data type, attributes and a brief description are provided in Figure 2 in detail.

2. cities

Key	Column Name	Data Type	Description
PK	`CityID`	`INT`	Unique identifier for each city.
	`CityName`	`VARCHAR(45)`	Name of the city.
	`Zipcode`	`DECIMAL(5,0)`	Population of the city.
FK	`CountryID`	`INT`	Reference to the corresponding country.

Figure 3: Interconnected Tables (MBA, 2023)

Cities datasets have 96 rows and 4 columns. This table stores the city name with a unique identifier, zipcode, and country ID as a foreign key, details shown in Figure 3.

3. countries

Key	Column Name	Data Type	Description
PK	`CountryID`	`INT`	Unique identifier for each country.
	`CountryName`	`VARCHAR(45)`	Name of the country.
	`CountryCode`	`VARCHAR(2)`	Two-letter country code.

Figure 4: Interconnected Tables (MBA, 2023)

The countries table stores country names and codes in 205 rows and 3 columns, the detail shown in Figure 4.

4. customers

Key	Column Name	Data Type	Description
PK	`CustomerID`	`INT`	Unique identifier for each customer.
	`FirstName`	`VARCHAR(45)`	First name of the customer.
	`MiddleInitial`	`VARCHAR(1)`	Middle initial of the customer.
	`LastName`	`VARCHAR(45)`	Last name of the customer.
FK	`cityID`	`INT`	City of the customer.
	`Address`	`VARCHAR(90)`	Residential address of the customer.

Figure 5: Interconnected Tables (MBA, 2023)

The customer's table contains the customer's demographic data in 97782 rows and 6 columns, and details are shown in Figure 5.

5. employees

Key	Column Name	Data Type	Description
PK	EmployeeID	INT	Unique identifier for each employee.
	FirstName	VARCHAR(45)	First name of the employee.
	MiddleInitial	VARCHAR(1)	Middle initial of the employee.
	LastName	VARCHAR(45)	Last name of the employee.
	BirthDate	DATE	Date of birth of the employee.
	Gender	VARCHAR(10)	Gender of the employee.
FK	CityID	INT	unique identifier for city
	HireDate	DATE	Date when the employee was hired.

Figure 6: Interconnected Tables (MBA, 2023)

The employee table contains employee demographic data in 23 rows and 8 columns, and details are provided in Figure 6.

6. products

Key	Column Name	Data Type	Description
PK	`ProductID`	`INT`	Unique identifier for each product.
	`ProductName`	`VARCHAR(45)`	Name of the product.
	`Price`	`DECIMAL(4,0)`	Price per unit of the product.
	`CategoryID`	`INT`	unique category identifier
	`Class`	`VARCHAR(15)`	Classification of the product.
	`ModifyDate`	`DATE`	Last modified date.
	`Resistant`	`VARCHAR(15)`	Product resistance category.
	`IsAllergic`	`VARCHAR`	indicates whether the item is an allergen
	`VitalityDays`	`DECIMAL(3,0)`	Product vital type classification.

Figure 7: Interconnected Tables (MBA, 2023)

The product table stores all product data in 452 rows and 9 columns and all informations are provided in Figure 7.

7. sales

Key	Column Name	Data Type	Description
PK	SalesID	INT	Unique identifier for each sale.
FK	SalesPersonID	INT	Employee responsible for the sale.
FK	CustomerID	INT	Customer making the purchase.
FK	ProductID	INT	Product being sold.
	Quantity	INT	Number of units sold.
	Discount	DECIMAL(10,2)	Discount applied to the sale.
	TotalPrice	DECIMAL(10,2)	Final sale price after discounts.
	SalesDate	DATETIME	Date and time of the sale.
	TransactionNumber	VARCHAR(25)	Unique identifier for the transaction.

Figure 7: Interconnected Tables (MBA, 2023)

The sales table contains all transactional data within 6,690,599 rows and 9 columns, and details attributes are provided in Figure 7.

The primary objective of this analysis report is to understand the components that affect sales and track them in real-time. As an international retail business, the following are the objectives of this analysis.

- Breaking down total revenue to its core components, like quarterly, monthly, daily, customers, products, regions, categories, etc.

- Identifying the total sales, the most selling category, the most selling product, the best performing city, and the best performing employee.
- Understand the sales trend, customers' purchase behaviours, product class and resistance's influence on sales, and sales volume in different cities.
- Develop and deploy an artificial intelligence-powered dashboard to track sales core components.
- Providing actionable recommendations to the organisation.

This report will help improve sales by understanding, tracking and taking control measures of the core influencers of sales. On top of that, this report will also provide adequate insights for the sales strategy development and implementation.

Theoretical Frameworks:

Data-driven decision-making is a crucial part of institutional decision-making, ensuring the effectiveness of the decision. Before making a decision backed by data, institutions need to develop proper decision support systems and infrastructure. In terms of decision support system development and deployment, multiple interrelated theories and frameworks have been developed in different periods and needs.

Resource-Based View (RBV) is one of the theories which believes that data and its analytics capabilities are an essential sustainable resource for the organisation, which provides an opportunity for competitive advantages to the organisation (Chong et al., 2022). For example, internal sales transaction data of the Pragati superstores provides crucial intelligence about customers, products, markets, demands, employees, etc, which can be leveraged against competition. So, this view believes that the data and its analytics mechanism are a sustainable resource for the company.

Only storing data in a database might cost more than generating opportunities and benefits. Thus, the Dynamic Capabilities Framework (DCF) is introduced, which emphasises the firms' ability to integrate, reconfigure, and adopt resources in response to a changing environment (Teece, Pisano and Shuen, 1997). Analytics-driven insights are most valuable where the organisation can react quickly and readjust the process and decisions. DCF provide opportunities to sense customer behaviours, market trends, so companies can redirect their efforts, aligning with new environments (Cao, Duan and El Banna, 2019). For example, Pragati Superstore can develop its information management system (IMS) and hire a data analytics team to build capabilities to quickly understand the data behaviours.

When databases become big and complex, only human-led insight analytics becomes insufficient for big data processing and decision making. Then, the Data-Driven Decision Making (DDDM) framework was introduced, which provides automation in the data-driven decision-making process by using empirical evidence, statistical models, and predictive analytics as the foundation for organisational choices (Mucci, 2024). DDDM is an operationally very efficient data-driven decision-making system because this framework allows pre-trained statistical models to make decisions based on data; however, it comes with both opportunities and challenges. It provides opportunities to take analytical and informed decisions based on data, but it requires a high level of cultural and technical skills, such as adequate analytical literacy among managers and high-quality data for input (Prelicz, 2025). By adapting DDDM, Pragati Superstore could formalise its decision-making process by emphasising measurable outcomes from its data insights. Pragati can develop and deploy statistical models to make automatic decisions regarding customers, promotions, pricing, branding, supply chain management, etc.

Implementing these theoretical frameworks, the Data Analytical Life Cycle (DALC) has been introduced, which provides a structured framework for data analytics projects and their deployments (Beneath Analytics, 2020). DALC follows multiple stages, such as discovery, data preparation, model planning, model building, communicating results, and operationalising to ensure that the analytics projects are aligned with appropriate methodologies and business objectives.

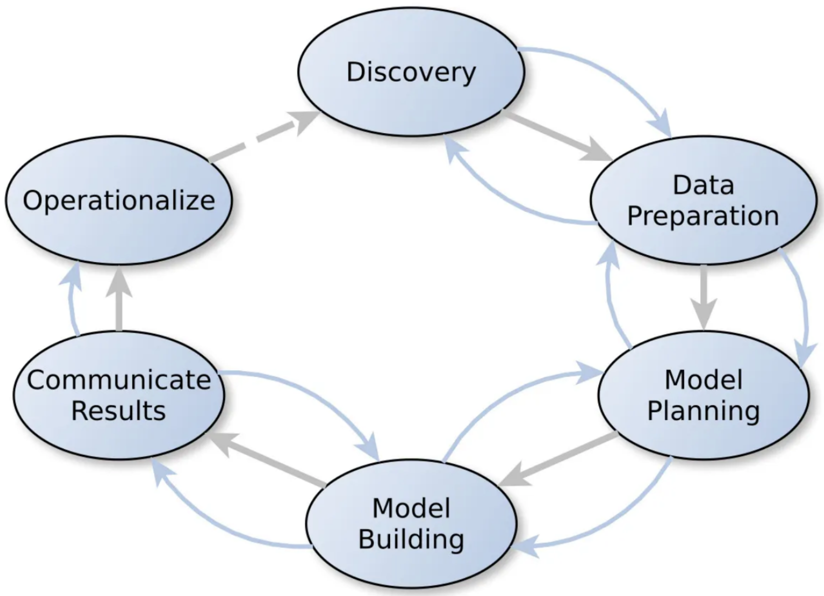


Figure 1: Different Phases of Data Analytics Life Cycle (Menon, 2021)

Figure 1 shows the process of DALC, where the cycle starts from discovery and ends with operationalising the model, then operationalising data feed for discovery as feedback, and this cycle runs continuously (Menon, 2021). In the discovery phase, orginazations define the specific challenges and opportunities, for example, Pragati Superstore identified that their past data could deliver rich insight about components that affect their sales, which helps to improve sales in the coming days. The data preparation phase reliability and completeness of input data. Moreover, model planning and model building phases extract insights from the data. Furthermore, communicating results and operationalising phases validate model effectiveness, deployment operationalises the results, feeds insights to the decision-making process, and the entire process feeds back, allowing continuous refinement of the iterative improvement of the cycle. The DALC deliver a bridge framework to implement RBV, DCF and DDDM theories in a real-life business environment.

Together with these frameworks, create a comprehensive perspective of data analytics in the organisational cases. RBV identifies the strategic resources for successful analysis (Chong et al., 2022), DCF provide capabilities to act with insights in the organisation

(Teece, Pisano and Shuen, 1997), DDDM ensure that the decisions are evidence-based and properly aligned with objectives (Mucci, 2024), and DALC provide a well-structured methodology to convert analytics projects from concepts to real-life deployments (Menon, 2021).

In conclusion, by adapting this framework, Pragati Superstore can systematically address the challenges of identifying a data-driven value and cope with implementation barriers. These theoretical frameworks not only identify problems and opportunities but also provide comprehensive ideas about how data analytics can be harnessed to achieve a competitive advantage sustainably.

Dashboard:

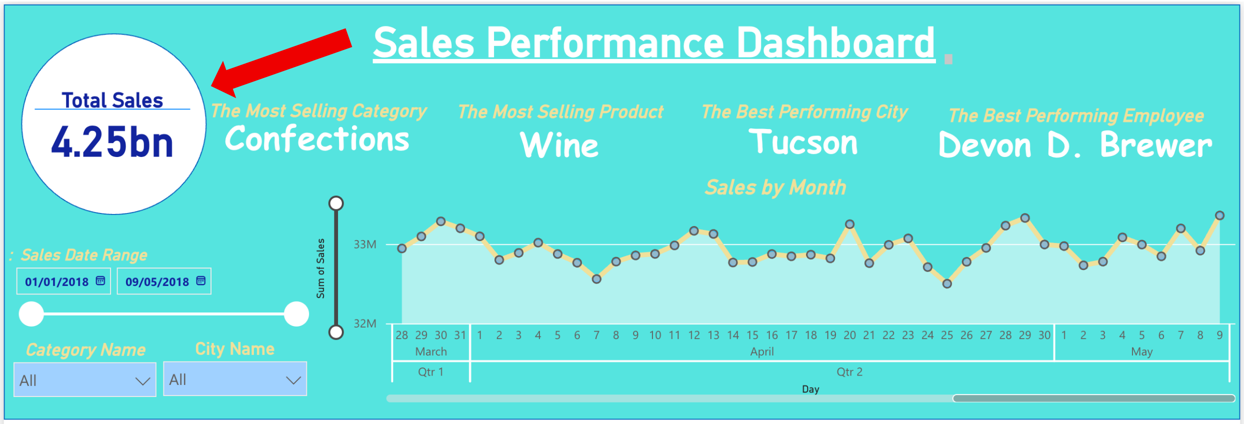
The dashboard is separately uploaded to the Git repository.

Justification of Dashboard Solution:

This dashboard is designed to analyse overall sales performance and components which influence the sales. There are five Key Performance Indicators (KPI), three slicers, one line chart, one tree map chart, one distribution bar chart, one stacked bar chart, one Azure geographical world map with bubble marker, and one AI-assisted Q&A search area in the dashboard interface. The dashboard is currently connected to a CSV file; however, it is designed for connecting live SQL database for real-time analysis. The usability and operational technique of each graph are provided as follows.

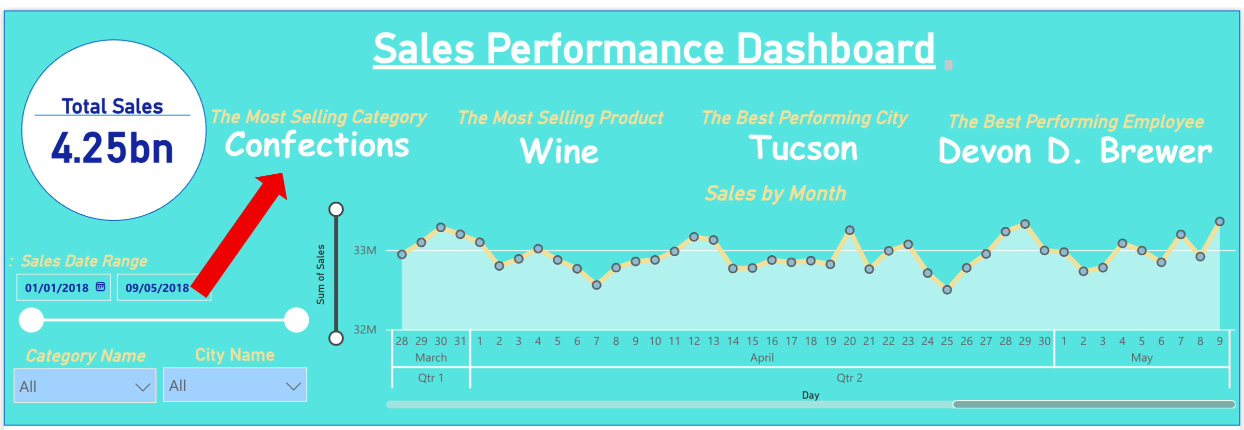
1. Total Sales (KPI):

The Total Sales indicatore is located in the top left corner with a round shape design, which shows the total sales of the Pragtati superstores during a particular period, e.g. Total sales up to now from the beginning of the year. This KPI helps to know the live total sales across the world.



2. The Most Selling Category (KPI):

The most selling category shows the most popular category, which contributes the largest percentage of sales in total sales during a certain period, e.g. Confections category contributes the highest sales up to now from the beginning of the year. Which helps to inform about the customer's most preferred category.

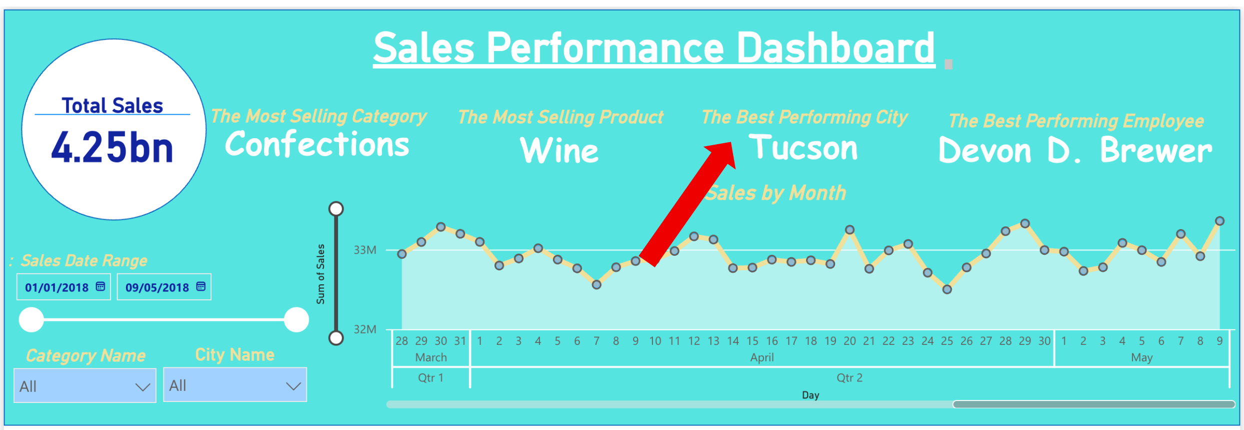


3. The Most Selling Product (KPI):

The best-selling product shows the product name, which generates the highest revenue for the company across all products. These analytics help to understand the most precious or popular product among customers in terms of sales amount, e.g. wine is the most popular product during the four months.

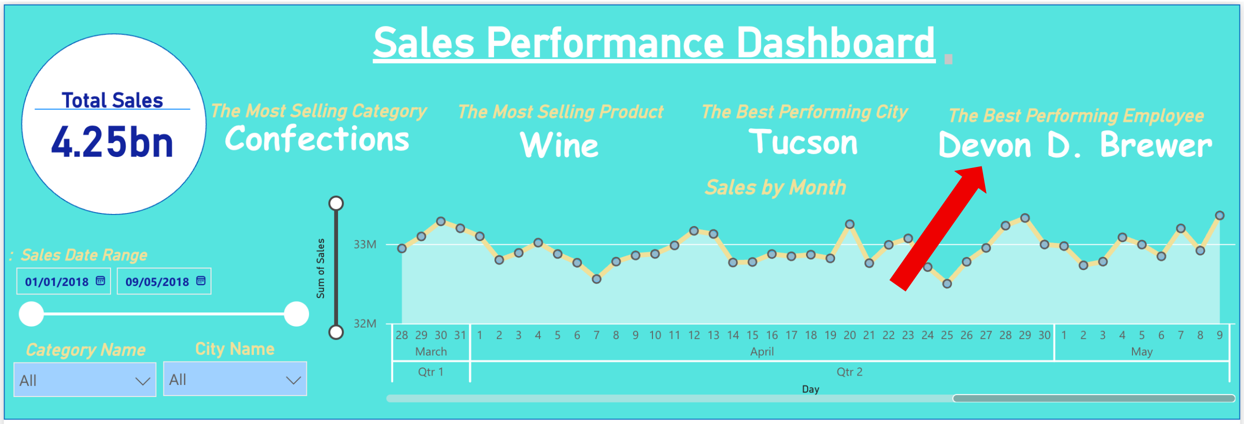
4. The Best Performing City (KPI):

The best performing city is the city which generates the most revenue for the company during the four months. This analysis helps to understand the most suitable city for further expansion because of the highest revenue record.



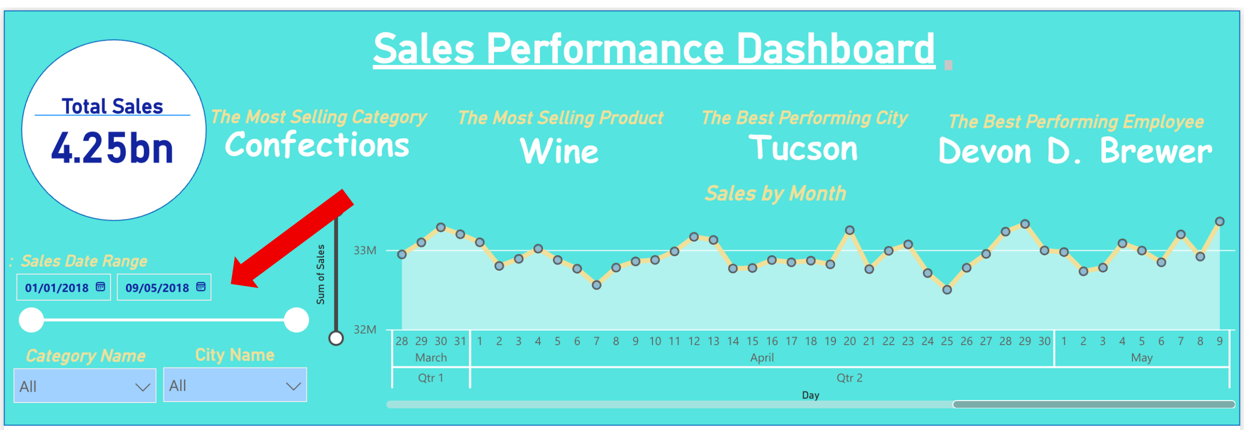
5. The Best Performing Employee (KPI):

The best performing employee is the one who sells the most. This KPI helps to know the best seller during a certain period. These kinds of KPIs help to know the best in the sector, and the company can provide awards and incentives to those people and sectors.



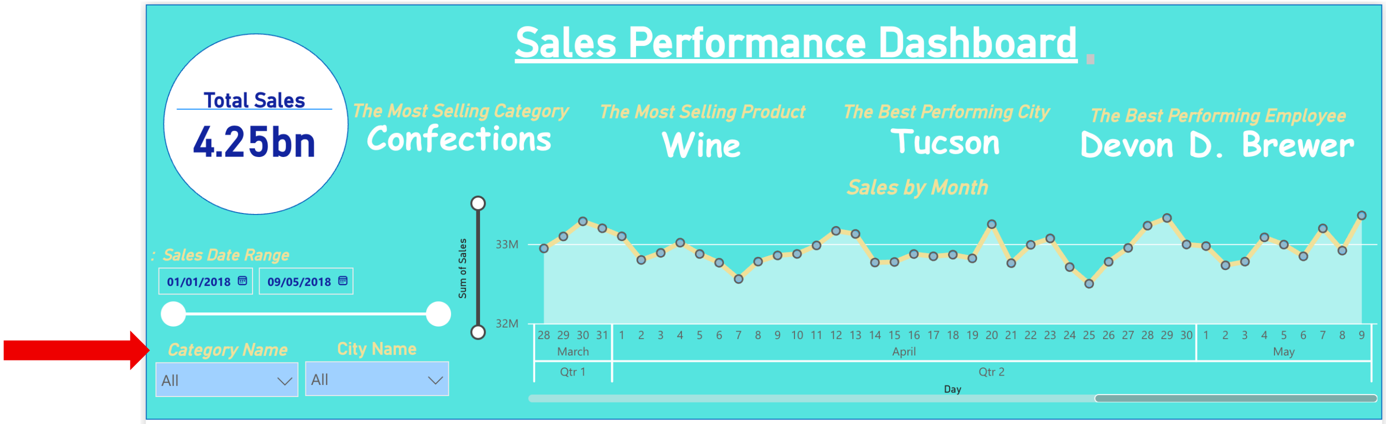
6. Sales Date Range (Slicer):

Sales date range slicer allows the user to select a specific start to end date as per requirement, which is responsive to all charts and KPIs. This helps to find out a specific period report from this dashboard, which enhances the dynamism and usability of the dashboard.



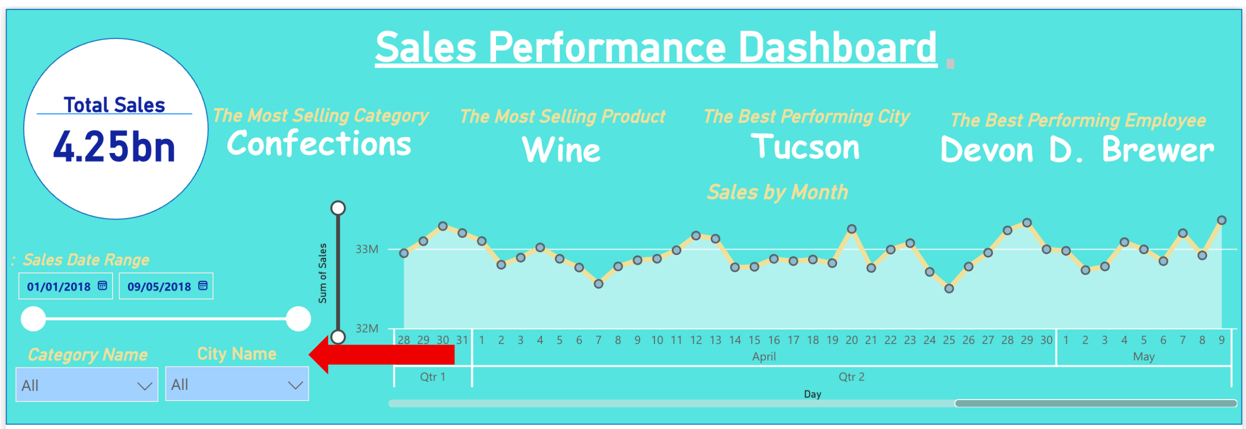
7. Category Name (Slicer):

The category name slicer allows the user to select a particular category, which filters the entire dashboard with the selected category, except sales by category name treemap chart, because there is no point in showing only one category portion in the graph. Users can select single, multiple or all categories at once.



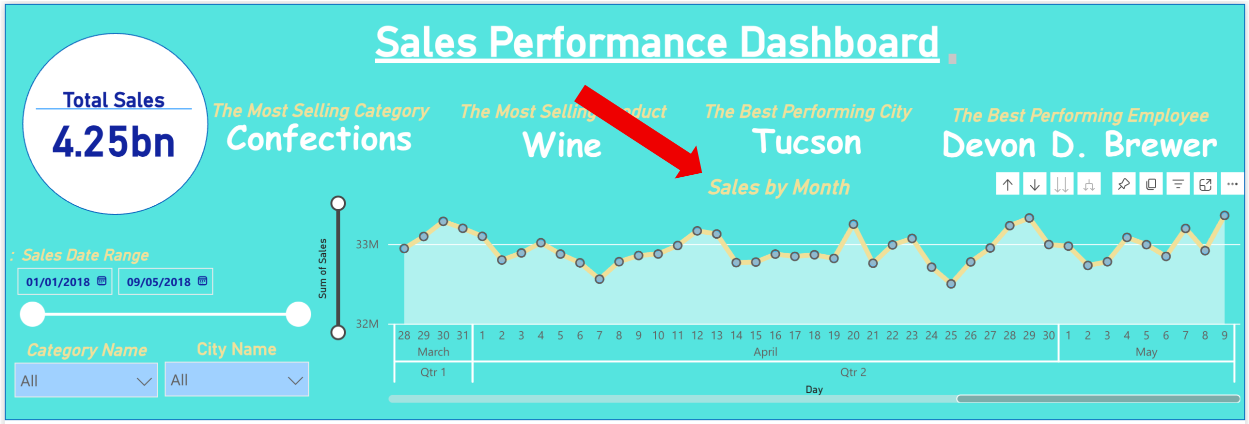
8. City Name (Slicer):

City name slicer allows users to filter the dashboard by city name, which applies to all outputs of the dashboard. Users can select single, multiple or all cities at once.



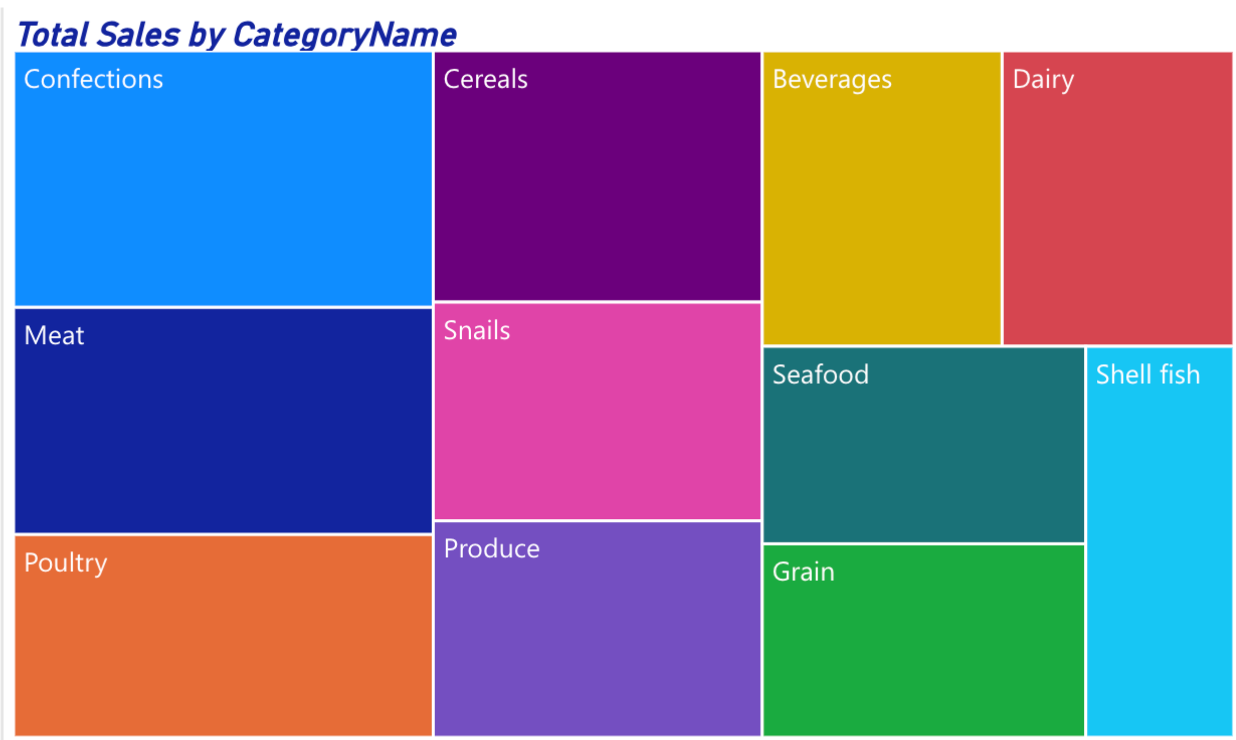
9. Sales by Month (Line Chart):

The sales by month line chart provides a live sales trend with drill up and down option to view for different period formats such as: yearly trend, quarterly trend, monthly trend or daily trend. This output is suitable for tracking live sales to take measures if any deviation occurs in the sales.



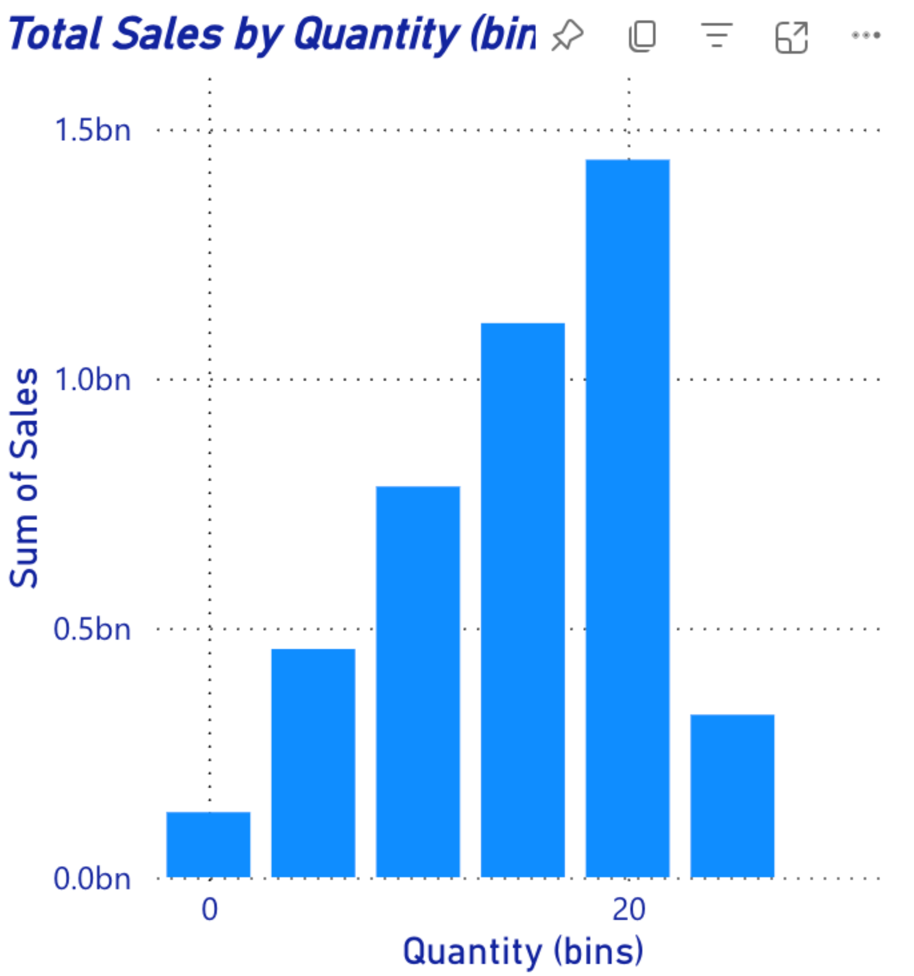
10. Total Sales by Category Name (Treemap Chart):

The total sales by category name treemap chart presents the category-wise sales, and the size of the tile represents the volume of sales, eg. The Largest size of the category presents the highest sales among all categories.



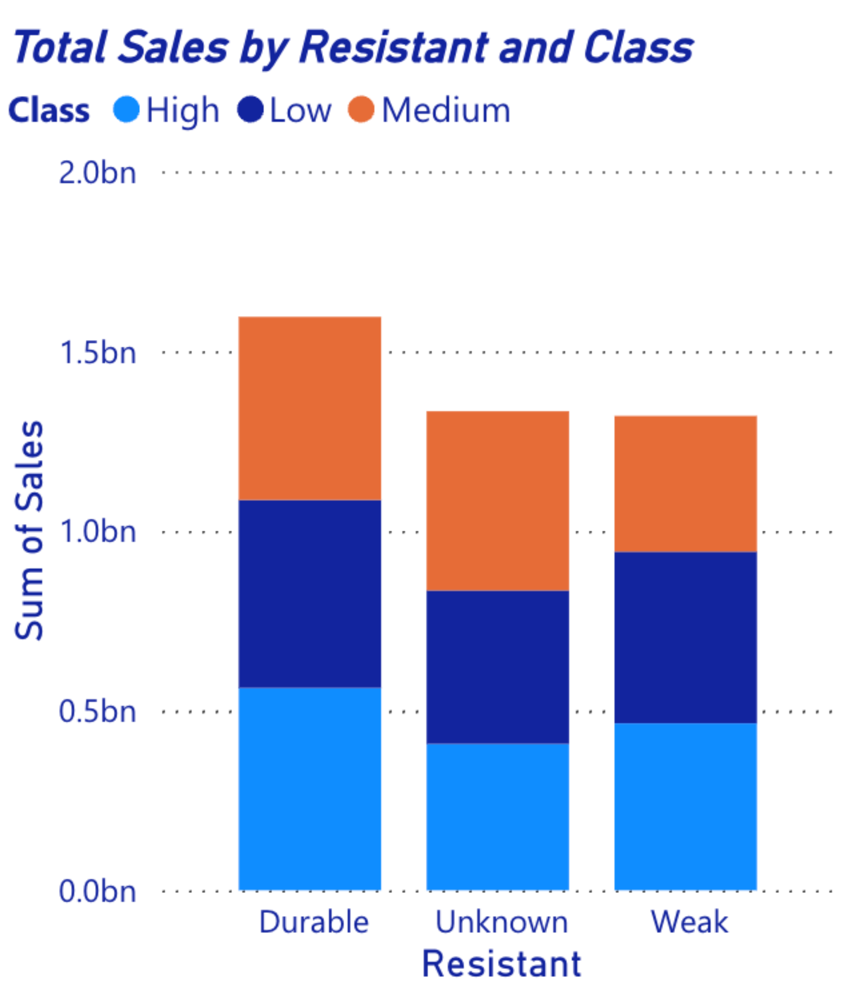
11. Total Sales by Quantity (Distribution Chart):

Total sales by quantity is a distribution chart of the frequency of sales quantity sizes, which is arranged in bins and displays total sales by sales quantity size per customer. For example, the most common quantity bin is 20-25 items per customer.



12. Total Sales by Resistance and Class (Stacked Barchart):

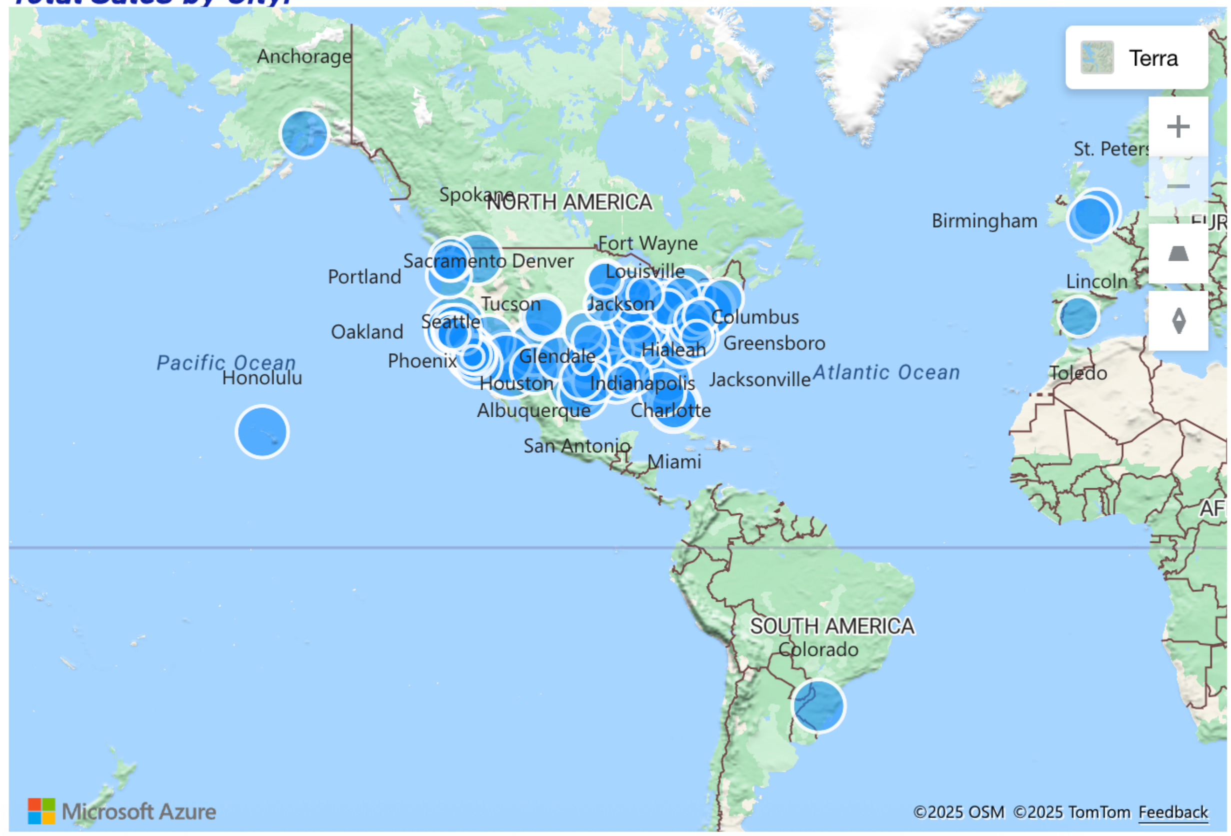
The total sales by resistance and class stacked bar chart shows the total sales value with respect to resistance with whole bars, and the value of class with different colours and sizes of colored bars.



13. Total Sales by City (Map Chart):

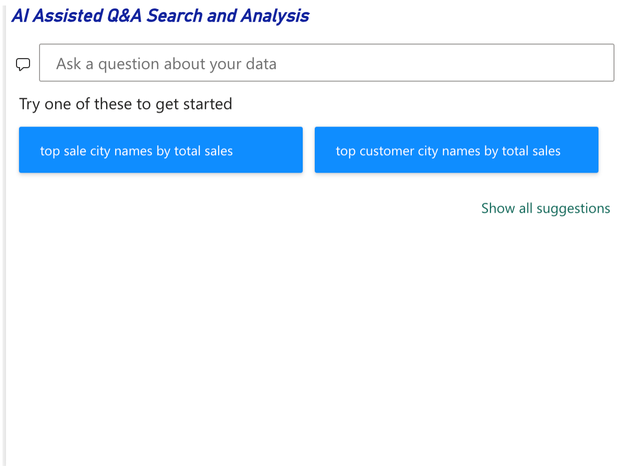
The total sales by city map chart shows the total sales size, which is presented by the size of the marker on the world map. Users can zoom in and zoom out on the map for better and comprehensive views. This helps to understand city-wise sales performance.

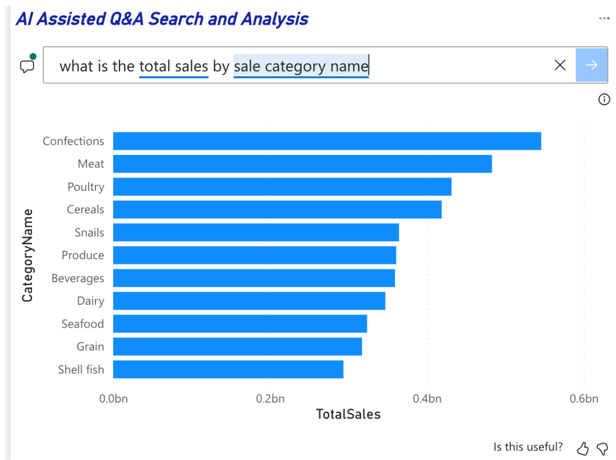
Total Sales by City:



14. AI-Assisted Q&A Search and Analysis (Q&A Chart):

This Q&A analysis chart provides functionalities to users for extracting desired insights from the datasets. In this area, users will get answers from the Power BI natural language processing system and AI-analysed insights, which enhance the capability of the dashboard by providing flexibility and customisation features in the insight extraction.





Conclusion and Recommendation:

The analysis demonstrates the successful integration of business intelligence (BI) and analytics systems in the Pragati Superstores to cope with challenges and unlock new opportunities. By adapting the multiple theories and frameworks, such as RBV, DCF, DDDM, and DALC, the analytics provides more than technical functionality; it becomes a strategic capability. RBV highlights Pragati's data assets and its analytics expertise as valuable resources. DCF explain the agility to convert insights into competitive advantages. DDDM ensure the managerial decisions are based on facts, and DACL provides a well-structured process for deploying data analytics solutions.

A combination of these perspectives in the business intelligence or analytics system can improve operational efficiency, better customer insights and better evidence-based strategies and planning. However, Pragati Superstores must address different barriers, such as the poverty of data, limited data quality, limited analytical capabilities, and resistance to implementation.

Recommendations:

To maximise the benefits of the BI/Analytics system, the following are the recommendations for Pragati Superstores.

- Develop a robust data infrastructure:

Pragati Superstores should invest in a centralised, scalable, secure, integrable and reliable data platform to ensure data quality, analytics and reporting. Currently available data is not adequate for in-depth analysis of profitability, HR, and financial performances.

- Implements data analytics into the decision-making process:

Pragati Supersotres should develop a culture to use their data insights through dashboards in every decision-making process at different levels. Higher-level management should implement and encourage data-driven decision-making.

- Investing in data analytics capabilities and skills:

Prgati Superstores should train their staff about data analytics and tools like Power BI, Tableau, Python, and R to develop in-house expertise in data analytics, which significantly reduces the dependency on external consultants.

- Adopt an iterative implementation approach:

Pragati Superstores should implement the Data Analytics Life Cycle (DALC) approach in their data analysis ecosystem because DALC provide opportunities for refinement and continuous improvement in the analytics system.

- Ensure strategic alignment and change management:

A business intelligence (BI) system is only effective if it closely aligns with organisational objectives. Just collecting data and producing reports is not a BI system. So, Pragati Superstores should support goals such as customer satisfaction, improving decision-making, reducing costs or streamlining operations.

BI system may change the habits of how decisions are used to be made. So, Pragati Superstores may face resistance from the organisational structure. Thus, Pragati Supersotres should implement a change management program to cope with these resistances.

References:

Beneath Analytics (2020). *Understanding the Data Analytics Life Cycle (DALC)*. [online] Medium. Available at: https://beneathanalytics.medium.com/understanding-the-data-analytics-life-cycle-dalc-e36b930dcb58 [Accessed 19 Aug. 2025].

Cao, G., Duan, Y. and El Banna, A. (2019). A dynamic capability view of marketing analytics: Evidence from UK firms. *Industrial Marketing Management*, 76, pp.72–83. Available at: https://doi.org/10.1016/j.indmarman.2018.08.002 [Accessed 23 Jul. 2025].

Chong, C.-L., Rasid, S.Z.A., Khalid, H.B. and Ramayah, T. (2022). Big Data Analytics Capability for Competitive Advantage and Firm Performance in Malaysian Manufacturing Firms. *SSRN Electronic Journal*. Available at: https://doi.org/10.2139/ssrn.4264007 [Accessed 23 Jul. 2025].

MBA, A.I. (2023). *Grocery Sales Database*. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/andrexibiza/grocery-sales-dataset/data [Accessed 19 Aug. 2025].

Menon, A. (2021). *Phases of Data Analytics Life Cycle*. [online] Medium. Available at: https://medium.com/@akankshamenon27/phases-of-data-analytics-life-cycle-16aa722497a1 [Accessed 19 Aug. 2025].

Mucci, T. (2024). *What Is Data-Driven Decision-Making? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/think/topics/data-driven-decision-making [Accessed 17 Aug. 2025].

Prelicz, A. (2025). *Ei Square®*. [online] Ei Square®. Available at: https://www.eisquare.co.uk/blogs/dddm-business-intelligence [Accessed 19 Aug. 2025].

Teece, D.J., Pisano, G. and Shuen, A. (1997). Dynamic Capabilities and Strategic Management. *Strategic Management Journal*, 18(7), pp.509–533.

Appendix:

Appendix 1: Data Cleaning Validation Snapshots:

Data Vaidation

```
[2]: # Import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[4]: # Import datasets
sales = pd.read_csv('sales.csv')
products = pd.read_csv('products.csv')
employees = pd.read_csv('employees.csv')
customers = pd.read_csv('customers.csv')
countries = pd.read_csv('countries.csv')
cities = pd.read_csv('cities.csv')
categories = pd.read_csv('categories.csv')
```

```
[5]: # Load all datasets in a single dictionary
datasets = ['sales', 'products', 'employees', 'customers', 'countries', 'cities', 'categories']
```

```
[6]: # Show datasets
for name in datasets:
    df = globals()[name]
    print(f'Dataset: {name}')
    print(df.head())  #
    print('\n' + '-' * 70 + '\n')
```

```
Dataset: sales
   SalesID  SalesPersonID  CustomerID  ProductID  Quantity  Discount  \
0        1              6       27039        381         7       0.0
1        2             16       25011         61         7       0.0
2        3             13       94024         23        24       0.0
3        4              8       73966        176        19       0.2
4        5             10       32653        310         9       0.0
```

```
[7]: # Check null values
for dataset in datasets:
    data = globals()[dataset]
    print(f'{dataset}: {data.isnull().sum()})')
    print('\n' + '-' * 40 + '\n')
```

```
sales: SalesID             0
SalesPersonID       0
CustomerID          0
ProductID           0
Quantity            0
Discount            0
TotalPrice          0
SalesDate           0
TransactionNumber   0
dtype: int64)

----------------------------------------
```

```
]: # Understand the proportion of null values
for shape in datasets:
    data = globals()[shape]
    print(f'shape: {shape}')
    print(f'{shape}: {data.shape[0]} Rows & {data.shape[1]} Columns')

    missing_percent = (data.isnull().sum()/len(data)) * 100
    print('Missing Values (%):')
    print(missing_percent)
    print('\n' + '-' * 40 + '\n')
```

```
shape: sales
sales: 6690599 Rows & 9 Columns
Missing Values (%):
SalesID             0.0
SalesPersonID       0.0
CustomerID          0.0
ProductID           0.0
Quantity            0.0
Discount            0.0
```

```
]: # Proportion of missing values is less than one percent
   # Let's remove all missioning values
   for value in datasets:
       data = globals()[value]
       globals()[value] = data.dropna()
```

```
]: # Check if nulls are gone
   for value in datasets:
       data = globals()[value]
       print(f'{value}: {data.isnull().sum()}')
       print('\n' + '-' * 30 + '\n')
```

```
sales: SalesID              0
SalesPersonID       0
CustomerID          0
ProductID           0
Quantity            0
Discount            0
TotalPrice          0
SalesDate           0
TransactionNumber   0
dtype: int64

------------------------------

products: ProductID       0
ProductName     0
Price           0
CategoryID      0
Class           0
ModifyDate      0
Resistant       0
IsAllergic      0
VitalityDays    0
dtype: int64

------------------------------

employees: EmployeeID       0
FirstName         0
MiddleInitial     0
```

```
1]: # Check duplicates
    for duplicate in datasets:
        data = globals()[duplicate]
        duplicate_count = df.duplicated().sum()
        print(f'{duplicate}: {duplicate_count}')
```

```
sales: 0
products: 0
employees: 0
customers: 0
countries: 0
cities: 0
categories: 0
```

```
2]: # Datasets are cleaned
    # Download the validated datasets
    for name in datasets:
        data = globals()[name]
        data.to_csv(f'{name}.csv', index = False)
        print(f'{name}.csv saved successfully')
```

```
sales.csv saved successfully
products.csv saved successfully
employees.csv saved successfully
customers.csv saved successfully
countries.csv saved successfully
cities.csv saved successfully
categories.csv saved successfully
```

```
3]: # Data visualization and dashbord will be developed in power bi
```

Appendix 2: Snapshot of Dashboard: