TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

January
2025

Analyzing Walmart Sales

Prepared By

SAIF CHOUAYA

EYA HAFSI

MOHAMED RAYEN FADHLAOUI

SAYDA OUADDAR

Table of Contents

1) Company Introduction

2) Requirement Analysis

3) Data Gathering

4) ETL Process
   - Data Extraction
   - Data Transformation
   - Data Loading

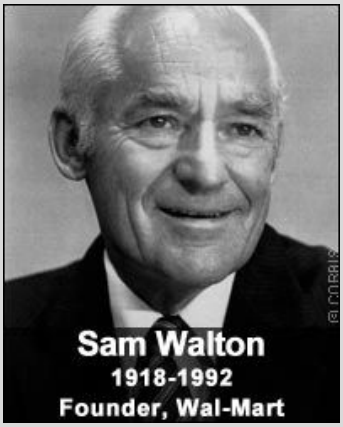
5) Data Warehousing

6) Data Modeling

7) Data Visualization and Analysis

TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

01 COMPANY INTRODUCTION



One of the largest multinational retail corporations in the world. Walmart was founded by Sam Walton in 1962 and is headquartered in Bentonville, Arkansas, USA.

Walmart operates a chain of hypermarkets, discount department stores, and grocery stores, offering a wide variety of products at competitive prices.

VISION

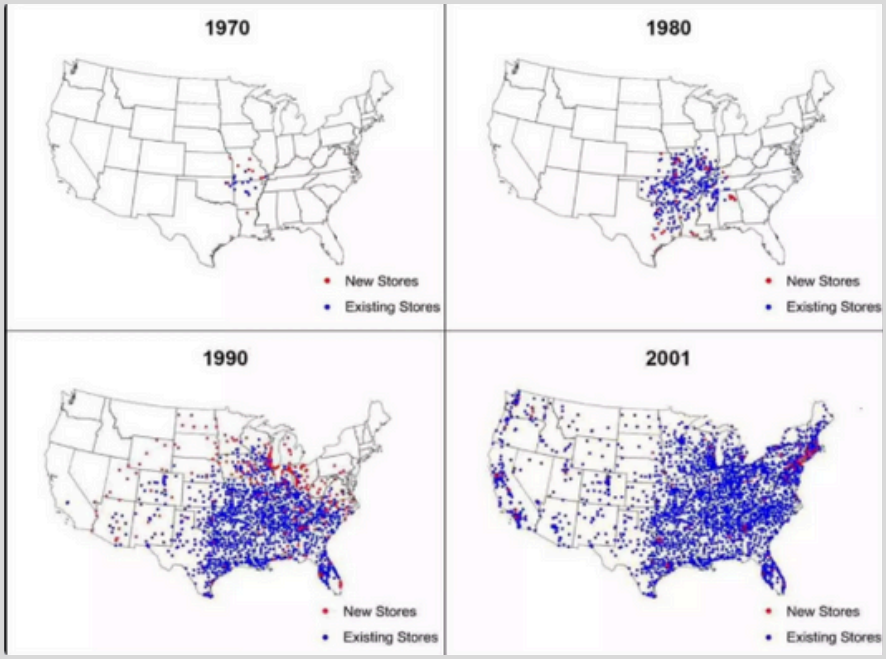
01 Walmart envisions being the go-to store for affordable and accessible products for everyday needs.

02 Walmart aims to transform retail by adopting cutting-edge technology and practices to enhance customer experiences.

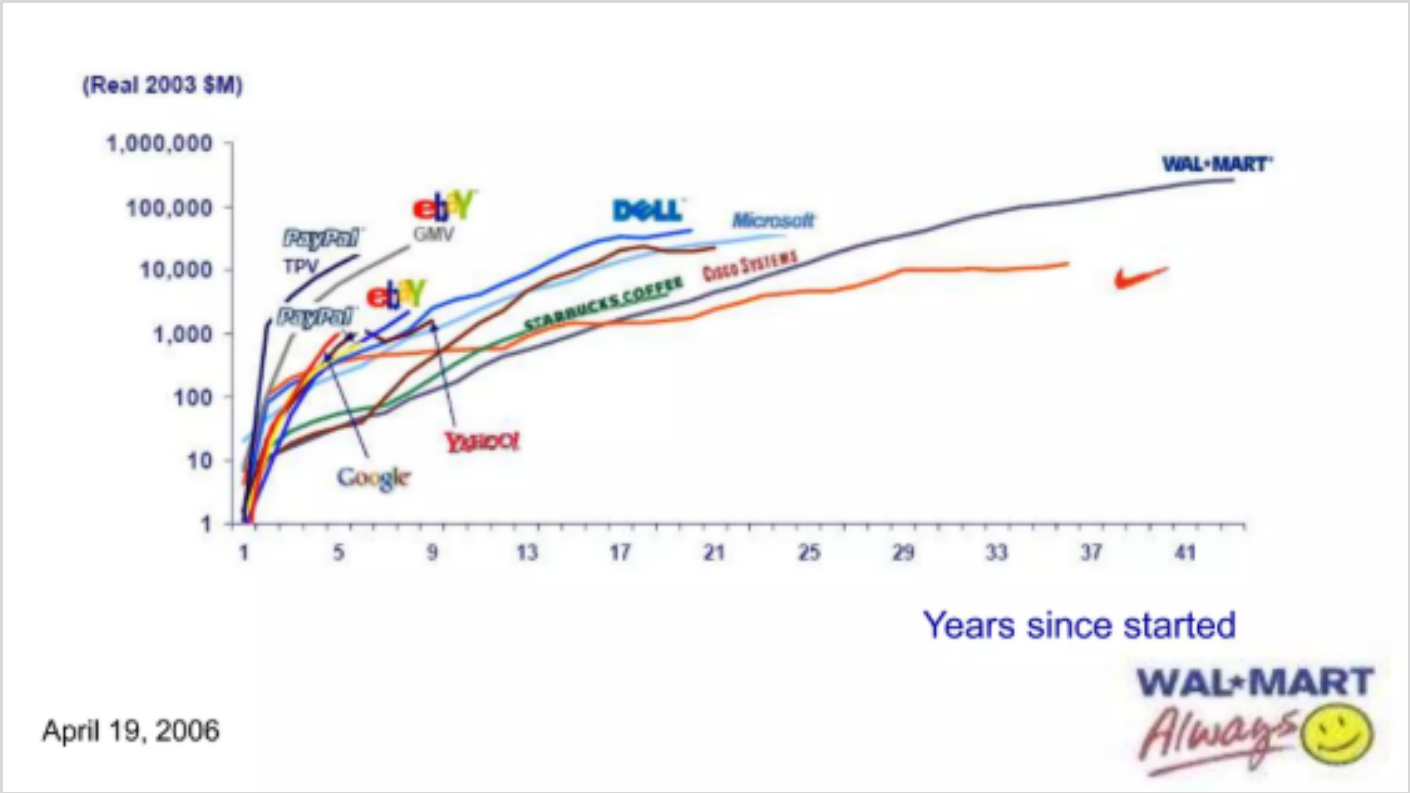
MISSION

01 Walmart focuses on providing quality products at low prices to improve customers' lives.

02 Walmart is committed to offering a seamless shopping experience in-store and online.



Over the years, Walmart expanded aggressively, opening new stores while strengthening its existing network.



Walmart's financial journey, showcases its remarkable growth over the years. Unlike tech giants like Google or Dell, Walmart's consistent revenue scaling has solidified its position as a retail powerhouse.

02 REQUIREMENT ANALYSIS

Walmart Sales

01 Which store locations generates the highest profit based on Walmart data?

02 What can be inferred about the sales performance across different periods at Walmart?

03 Which product category has the highest sales percentage at Walmart?

04 Which customer segment generates the highest proportion of orders at Walmart?

05 What is the range of shipping costs across different locations at Walmart?

03 DATA GATHERING

Walmart Sales

WE HAVE EXTRACTED DATA FROM 4 DISTINCT SOURCES, INCLUDING CSV FILES, EXCEL FILES, JSON FILES, AND APIS.

1) CSV Files

- **Grocery Dataset**
  - FILE: WMT_Grocery_202209.csv
  - DESCRIPTION: This dataset contains information about grocery products available at Walmart. It includes details such as the shipping location, department, category, subcategory, product name, brand, retail price, current price, product size, and promotion status. The data is useful for analyzing product offerings, pricing strategies, and inventory management in the grocery department.
  - SHAPE: 568534*16
- **Product Dataset**
  - FILE: product.csv
  - DESCRIPTION: This dataset provides detailed information about various products sold at Walmart. It includes attributes such as the product URL, final price, SKU, currency, GTIN, specifications, image URLs, customer reviews, rating stars, brand, breadcrumbs, category IDs, review count, product description, and more. This dataset is valuable for understanding product details, customer feedback, and product categorization.
  - SHAPE: 1095*44
- **Store Status Public Dataset**
  - FILE: STORE_STATUS_PUBLIC_VIEW_-8827503165297490716.csv
  - DESCRIPTION: This dataset contains information about the status of Walmart stores, including their names, numbers, descriptions, types, addresses, cities, counties, states, postal codes, and operation status. It also includes geographic coordinates (x, y) for each store. This dataset is useful for analyzing store locations, operational status, and geographic distribution.
  - SHAPE: 5234*14

- SKU (Stock Keeping Unit): is a unique identifier assigned to a specific product or item in a retailer's inventory, used to track and manage stock levels, sales, and inventory efficiently.

2) EXCEL Files

- Clients Dataset
  - FILE: Clients.xlsx
  - DESCRIPTION: This dataset contains customer-related information for Walmart, including demographic and financial details. It can be used to analyze customer behavior, attrition, and segmentation.
  - SHAPE: 10127*9
- Walmart Sales Dataset
  - FILE: Walmart-Sales.xlsx
  - DESCRIPTION: This dataset contains transactional data for Walmart sales, including details about products, branches, and payment methods. It can be used to analyze sales performance, customer preferences, and payment trends.
  - SHAPE: 1000*12
- Walmart US Retail (2012-2015) Dataset
  - FILE: walmart-US-Retail-Data-2012-2015.xlsx
  - DESCRIPTION: This dataset contains retail data for Walmart in the US from 2012 to 2015. It includes information about orders, customers, sales, and shipping. It can be used to analyze sales trends, regional performance, and profitability.
  - SHAPE: 8399*25

3) JSON File

- Walmart Store Locations
  - FILE: json.geojson
  - DESCRIPTION: This dataset contains geographic information about Walmart store locations in the United States. It is stored in GeoJSON format, which is a standard format for encoding geographic data structures. The dataset includes details about each store's location, business unit information, and operational status.
  - SHAPE: 5234*13

4) APIs

- From HuggingFace API:
  - URL: dataset_url
  - NAME: Walmart Store Information
  - DESCRIPTION: This dataset contains detailed information about Walmart stores, including their opening dates, locations, and types. It provides historical and geographic data that can be used to analyze store distribution, growth trends, and regional presence.
  - SHAPE: 100*16

HUGGING FACE

Dataset Viewer | Auto-converted to Parquet | API | Embed | Full Screen Viewer

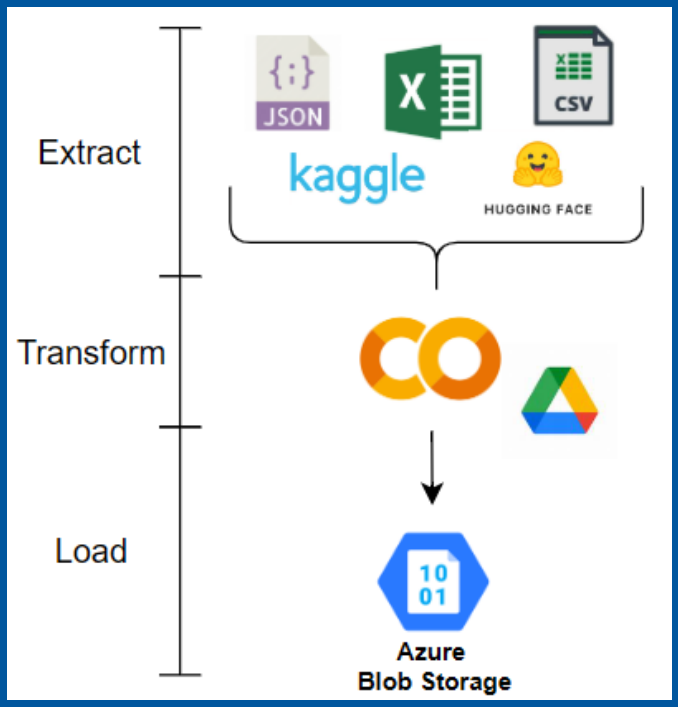
Split (1)
train · 2.99k rows

Search this dataset | SQL Console

storenum int64	OPENDATE string · lengths	date_super string · lengths	conversion float64
8	10/1/67	3/1/93	1
7	10/1/67	null	null
10	7/1/68	3/1/98	1
13	11/1/68	3/1/96	1
12	7/1/68	3/1/94	1
11	3/1/68	2/20/02	1

- From Kaggle API:
  - PATH to URL: /root/.cache/kagglehub/datasets/yasserh/walmart-dataset/versions/1
  - NAME: Walmart Weekly Sales
  - DESCRIPTION: This dataset contains weekly sales data for Walmart stores, along with additional contextual information such as holiday flags, temperature, fuel prices, economic indicators (CPI), and unemployment rates. It is useful for analyzing sales trends, the impact of external factors on sales, and forecasting future performance.
  - SHAPE: 6435*8

04 ETL PROCESS



- After gathering the necessary data to address the questions, we processed it through an ETL (Extract, Transform, Load) pipeline. The transformed data was saved in Google Drive as a staging area for further use.

4.1 Data Extraction:

- Logical Extraction: Conducted a full extraction of data from the sources (complete dataset extraction).
- Physical Extraction: Extracted data from external sources, as direct access to the company's internal systems was unavailable (offline extraction).

4.2 Data Transforamtion::

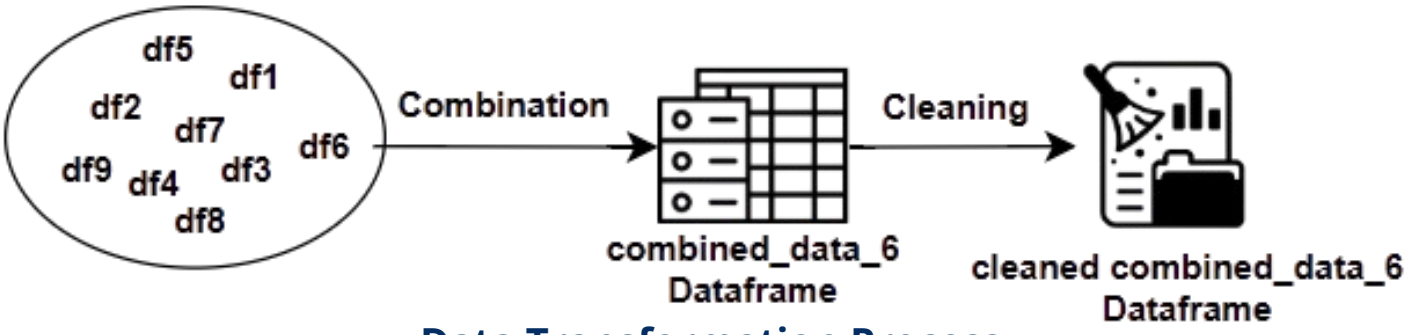
- After loading all the data from different sources into Google Drive for use in Google Colab, we proceeded to the transformation phase, which was divided into 2 parts: combination and cleaning.
- 1) Combination:
  - We began by combining the first two datasets: the Grocery dataset and the Product dataset.
  - We identified common columns between them: {'product_name', 'brand', 'sku', 'breadcrumbs'}.

- The datasets were merged based on the SKU column, ensuring a logical connection.
- Additional dataframes were added one by one, and incompatible dataframes (those without shared columns) were discarded.
- This process resulted in a final combined dataset named combined_data_6, which includes data from df1, df2, df7, and df8.
- Resulting Dataset:
  - The combined_data_6 dataset contains 94 columns and 1,606 rows.
  - Key columns include:
    - Product-related: product_name, brand, sku, price_retail, price_current, product_size, rating, ingredients, etc.
    - Transaction-related: invoice id, quantity, date, payment, sales, profit, etc.
    - Customer-related: customer name, customer age, customer segment, gender, etc.
    - Location-related: city, state, zip code, shipping_location, etc.

- 2) Cleaning:
  - The combined_data_6 dataset will now undergo the cleaning phase, which is divided into 4 key steps:
  - 1. Handle Duplicate Columns:
    - Remove Duplicate Columns: Identical columns will be removed to avoid redundancy.
    - Merge Columns with Suffixes: For columns with similar data but different names (e.g., product_name_x and product_name_y), they will be merged into a single column.
  - 2. Check for Missing Values:
    - Identify Missing Values
    - Fill or Drop Missing Values:
      - For columns critical to analysis, missing values will be filled with default values (e.g., mean, median, or mode for numerical columns, or a placeholder like "Unknown" for categorical columns).

- For less important columns or rows with excessive missing data, they will be dropped to maintain data quality.

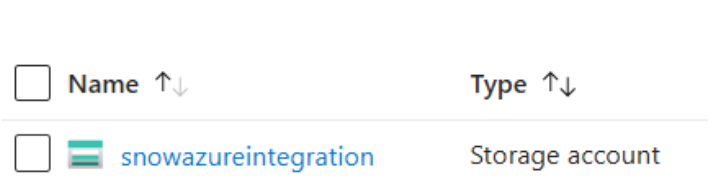
- 3. Standardize Column Names:
  - Convert all column names to lowercase for consistency.
  - Remove any unwanted spaces, special characters, or symbols to ensure uniformity and ease of use in analysis.
- 4. Convert Data Types:
  - Convert Date Columns to Datetime
  - Convert Numeric Columns
  - Convert Categorical Columns
  - Check for Outliers or Invalid Data

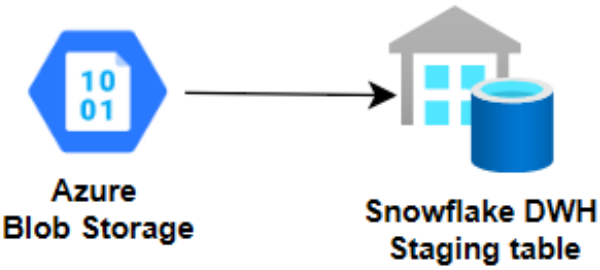


Data Transformation Process

4.3 Data Loading:

- Then, the transformaed data is loaded into a blob container within the Azure Storage Account named snowazureintegration. Additionaly, it will be loaded into a pre-defined schema table named staging_area in Snowflake for datawarehousing and modeling.





05 DATA WAREHOUSING

- We have chosen Snowflake as the cloud-based Data Warehouse Solution, because of its scalability, performance, and cloud-native architecture to handle data warehousing needs.
- First, we configured a connection to load data from Azure Blob Storage into Snowflake.

```
-- Create a storage integration object
-- so that it can read and write data present in the file in azure container instance
CREATE STORAGE INTEGRATION snow_azure_int
    TYPE = EXTERNAL_STAGE
    STORAGE_PROVIDER = AZURE
    ENABLED = TRUE
    AZURE_TENANT_ID = 'dbd6664d-4eb9-46eb-99d8-5c43ba153c61'
    STORAGE_ALLOWED_LOCATIONS = ('azure://snowazureintegration.blob.core.windows.net/snowflakeazurefile');

    -- Describe integration object
DESC STORAGE INTEGRATION snow_azure_int;


    // Create database and
schemaSNOWFLAKE.ALERT.GET_CONDITION_QUERY_UUIDINVENTORY_WH.INFORMATION_SCHEMA.APPLICABLE_ROLES
CREATE DATABASE IF NOT EXISTS MYDB;
CREATE SCHEMA IF NOT EXISTS MYDB.file_formats;
CREATE SCHEMA IF NOT EXISTS MYDB.external_stages;

// Create file format object
CREATE OR REPLACE file format mydb.file_formats.csv_fileformat
    type = csv
    field_delimiter = '|'
    skip_header = 1
    empty_field_as_null = TRUE;

// Create stage object with integration object & file format object
CREATE OR REPLACE STAGE mydb.external_stages.stg_azure_cont
    URL = 'azure://snowazureintegration.blob.core.windows.net/snowflakeazurefile'
    STORAGE_INTEGRATION = snow_azure_int
    FILE_FORMAT = mydb.file_formats.csv_fileformat ;


//Listing files under your azure containers
list @mydb.external_stages.stg_azure_cont;
```

- Then, we created a Staging Table (Sales_Staging) defined with a comprehensive schema to temporarily store raw data.

```
// create a staging table (or an intermediary table)
// that loads all columns from the file, and then
// assign or map the relevant columns from that table
// to your fact and dimension tables.
CREATE OR REPLACE TABLE Sales_Staging (
    index STRING,
    shipping_location STRING,
    department STRING,
    category STRING,
    sku STRING,
    product_url STRING,
    brand_x STRING,
    price_retail FLOAT,
    price_current FLOAT,
    product_size STRING,
    promotion STRING,
    rundate DATE,
    tid STRING,
    timestamp TIMESTAMP,
    url STRING,
    final_price FLOAT,
    currency STRING,
    gtin STRING,
    specifications STRING,
    image_urls STRING,
    top_reviews STRING,
    rating_stars INT,
    related_pages STRING,
    available_for_delivery STRING,
    available_for_pickup STRING,
    brand_y STRING,
    category_ids STRING,
    review_count INT,
    description STRING,
    product_id STRING,
    review_tags STRING,
    category_url STRING,
    category_name STRING,
    category_path STRING,
    root_category_url STRING,
    root_category_name STRING,
    upc STRING,
    tags STRING,
    main_image STRING,
    rating FLOAT,
    unit_price FLOAT,
    unit STRING,
    aisle STRING,
    free_returns STRING,
    sizes STRING,
    colors STRING,
    seller STRING,
    other_attributes STRING
    customer_reviews STRING
    ingredients STRING,
    initial_price FLOAT,
    discount_x FLOAT,
    ingredients_full STRING
    categories STRING,
    invoice_id STRING,
    branch STRING,
    city_x STRING,
    customer_type STRING,
    gender STRING,
    product_line STRING,
    quantity INT,
    date DATE,
    payment STRING,
    city_y STRING,
    customer_age INT,
    customer_name STRING,
    customer_segment STRING,
    discount_y FLOAT,
    number_of_records INT,
    order_date DATE,
    order_id STRING,
    order_priority STRING,
    order_quantity INT,
    product_base_margin FLOAT,
    product_category STRING,
    product_container STRING,
    product_name STRING,
    product_sub_category STRING,
    profit FLOAT,
    region STRING,
    row_id STRING,
    sales FLOAT,
    ship_date DATE,
    ship_mode STRING,
    shipping_cost FLOAT,
    state STRING,
    zip_code STRING
);
```

06 DATA MODELING

- Inside Snowflake, We designed a Star Schema by Divided the Sales_Staging table into 1 fact table and 5 dimension tables for efficient data organization and querying.
- We created fact and dimension tables.
- FACT table:
  - NAME: Sales_Fact
  - DESCRIPTION: Defined to store transactional data with measures like quantity, sales, initial_price, final_price, profit, shipping_cost, order_quantity, discount_x, discount_y, and number_of_records.
  - Linked to dimension tables via foreign keys (product_id, customer_name, shipping_location, promotion, order_date, ship_date).

- DIMENSION tableS:
  - Product_Dimension:
    - Stores product-related attributes like product_id, sku, product_name, category, brand_x, brand_y, description, image_urls, colors, sizes, tags, main_image, upc, gtin, category_ids, category_url, root_category_name, category_path, other_attributes, ingredients, and ingredients_full.DESCRIPTION: Defined to store transactional data with measures like quantity, sales, initial_price, final_price, profit, shipping_cost, order_quantity, discount_x, discount_y, and number_of_records.
  - Customer_Dimension
    - Stores customer-related attributes like customer_name, customer_type, customer_age, customer_segment, and gender.
  - Shipping_Dimension
    - Stores shipping-related attributes like shipping_location, state, zip_code, region, city_x, city_y, branch, invoice_id, and free_returns.
  - Time_Dimension
    - Stores time-related attributes like order_date, ship_date, timestamp, and rundate.
  - Promotion_Dimension
    - Stores promotion-related attributes like promotion_id, promotion, discount_x, discount_y, and final_price.
- We populated the fact and dimension tables with data.

```
// create customer_dimension table
CREATE OR REPLACE TABLE Customer_Dimension (
    customer_name STRING PRIMARY KEY,
    customer_type STRING,
    customer_age INT,
    customer_segment STRING,
    gender STRING
);
INSERT INTO Customer_Dimension (
    customer_name,customer_type,customer_age,customer_segment,
gender
)
SELECT
    customer_name,customer_type,
customer_age,customer_segment,gender
FROM Sales_Staging;
```

- We validated that the tables are working as expected.

```
//Validate the data
SELECT * FROM Sales_Staging LIMIT 10;

-- Validate Fact Table
SELECT * FROM Sales_Fact;

-- Validate Dimension Tables
SELECT * FROM Product_Dimension;
SELECT * FROM Customer_Dimension;
SELECT * FROM Promotion_Dimension;
SELECT * FROM Shipping_Dimension;
SELECT * FROM Time_Dimension;
```

- Finally, We performed ROLAP Queries.
- Why ROLAP?:
  - ROLAP was chosen because it allows for direct querying of relational databases (like Snowflake) without requiring pre-aggregation.
  - It leverages Snowflake's powerful SQL capabilities and scalability for large datasets.

```
// sql query 2: Average Profit by Customer Segment
SELECT
    cd.customer_segment,
    AVG(sf.profit) AS avg_profit
FROM Sales_Fact sf
JOIN Customer_Dimension cd ON sf.customer_name = cd.customer_name
GROUP BY cd.customer_segment
ORDER BY avg_profit DESC;
```

```
// Query 1: Total Sales by Product Category
SELECT
    pd.category,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY pd.category
ORDER BY total_sales DESC;
```

```
//query 3: Sales by Region and Month
SELECT
    sh.region,
    EXTRACT(MONTH FROM sf.order_date) AS month,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Shipping_Dimension sh ON sf.shipping_location =
sh.shipping_location
GROUP BY sh.region, EXTRACT(MONTH FROM sf.order_date)
ORDER BY region, month;
```

```
// query 4: Top Products by Sales Volume
SELECT
    pd.product_name,
    SUM(sf.quantity) AS total_quantity_sold
FROM Sales_Fact sf
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY pd.product_name
ORDER BY total_quantity_sold DESC
LIMIT 10;
```

```
// query 5: Sales by Promotion and Time
SELECT
    pr.promotion,
    EXTRACT(YEAR FROM sf.order_date) AS year,
    EXTRACT(MONTH FROM sf.order_date) AS month,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Promotion_Dimension pr ON sf.promotion = pr.promotion
GROUP BY pr.promotion, EXTRACT(YEAR FROM sf.order_date), EXTRACT(MONTH FROM
sf.order_date)
ORDER BY year, month, total_sales DESC;
```

```
// query6: Sales and Profit by Time Period
SELECT
    EXTRACT(MONTH FROM sf.order_date) AS month,
    SUM(sf.sales) AS total_sales,
    SUM(sf.profit) AS total_profit
FROM Sales_Fact sf
GROUP BY EXTRACT(MONTH FROM sf.order_date)
ORDER BY month;
```

```
// Quey 7: Customer Segments and Average Order Quantity
SELECT
    cd.customer_segment,
    AVG(sf.order_quantity) AS avg_order_quantity
FROM Sales_Fact sf
JOIN Customer_Dimension cd ON sf.customer_name = cd.customer_name
GROUP BY cd.customer_segment
ORDER BY avg_order_quantity DESC;
```

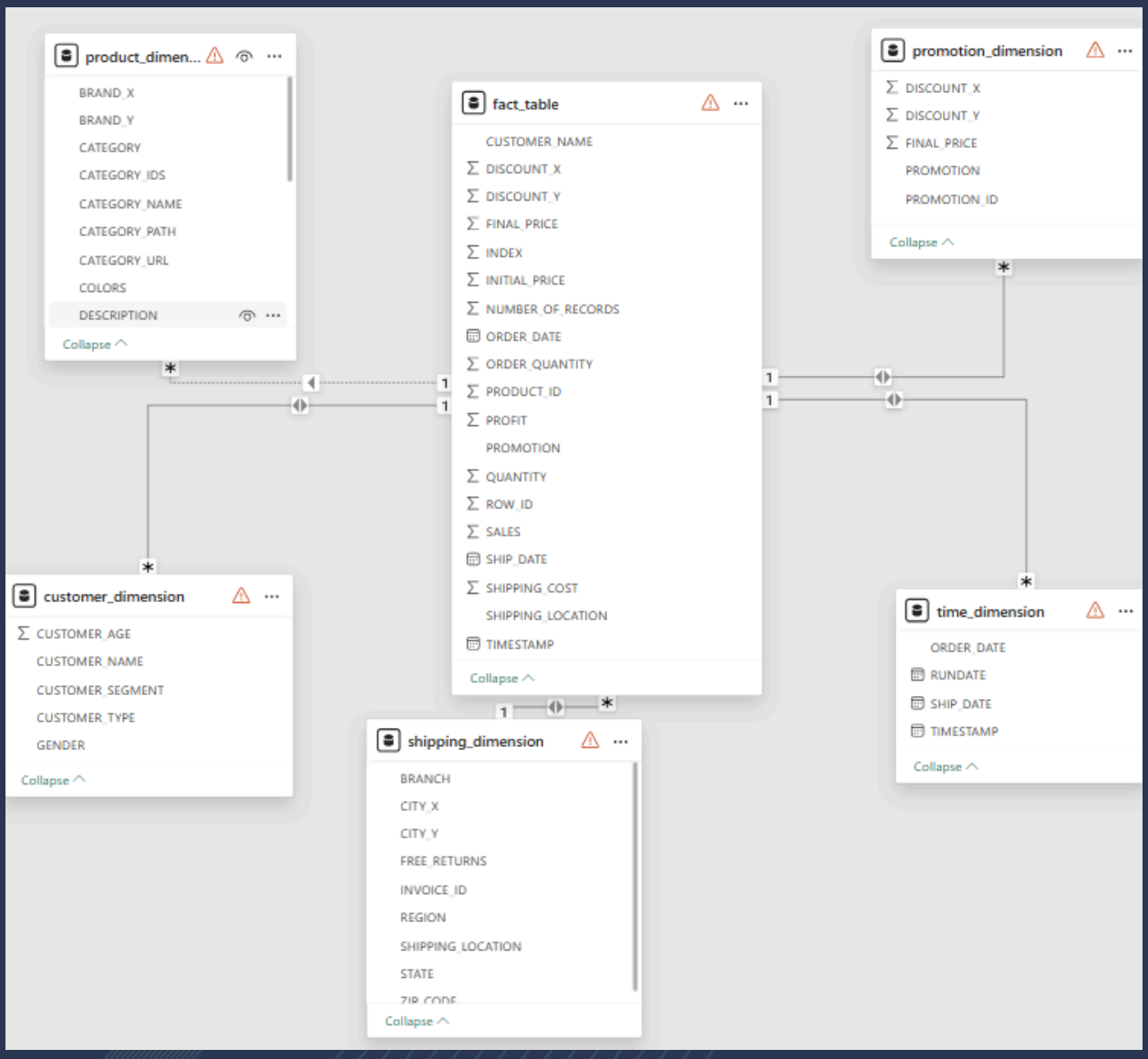
```
// Query 8: Sales by State and Product Category
SELECT
    sh.state,
    pd.category,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Shipping_Dimension sh ON sf.shipping_location = sh.shipping_location
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY sh.state, pd.category
ORDER BY total_sales DESC;
```

```
// query 9: Sales Performance by Region and Product
SELECT
    sh.region,
    pd.product_name,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Shipping_Dimension sh ON sf.shipping_location = sh.shipping_location
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY sh.region, pd.product_name
ORDER BY total_sales DESC;
```

```
// query 10: Average Shipping Cost by Product Category
SELECT
    pd.category,
    AVG(sf.shipping_cost) AS avg_shipping_cost
FROM Sales_Fact sf
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY pd.category
ORDER BY avg_shipping_cost DESC;
```

```sql
// query 11: Sales by Customer Age Group
SELECT
    CASE
        WHEN cd.customer_age < 20 THEN 'Under 20'
        WHEN cd.customer_age BETWEEN 20 AND 30 THEN '20-30'
        WHEN cd.customer_age BETWEEN 30 AND 40 THEN '30-40'
        WHEN cd.customer_age BETWEEN 40 AND 50 THEN '40-50'
        ELSE '50+'
    END AS age_group,
    SUM(sf.sales) AS total_sales
FROM Sales_Fact sf
JOIN Customer_Dimension cd ON sf.customer_name = cd.customer_name
GROUP BY age_group
ORDER BY total_sales DESC;
```

```sql
// query 12: Profit by Brand and Product Category
SELECT
    pd.brand_x,
    pd.category,
    SUM(sf.profit) AS total_profit
FROM Sales_Fact sf
JOIN Product_Dimension pd ON sf.product_id = pd.product_id
GROUP BY pd.brand_x, pd.category
ORDER BY total_profit DESC;
```

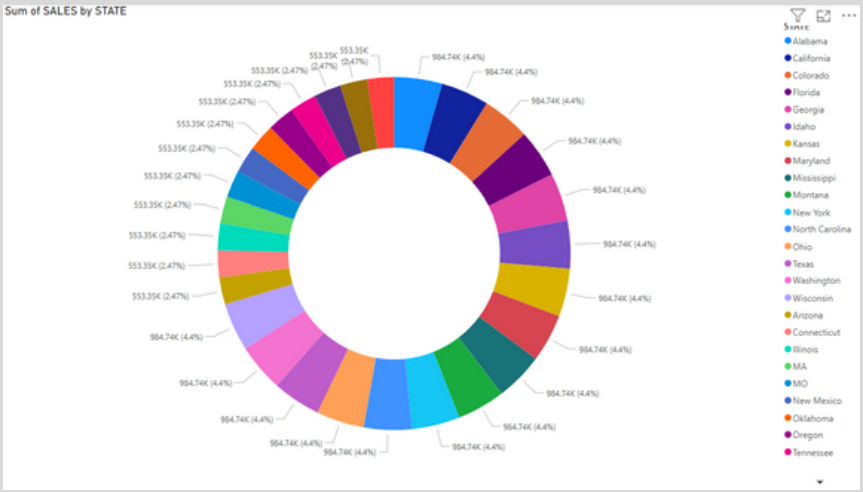


07 DATA VISUALISATION AND ANALYSIS

- Power BI, one of the most popular and user-friendly business intelligence tools for analysis, is highly compatible with all dimensional models. In our project, we will use the Desktop version of this software to conduct analysis and address the questions outlined in requirement Analysis.

Store Performance Analysis

- To decide which stores are best performing, we chose a donut chart to visualize the states with the best sales performance.

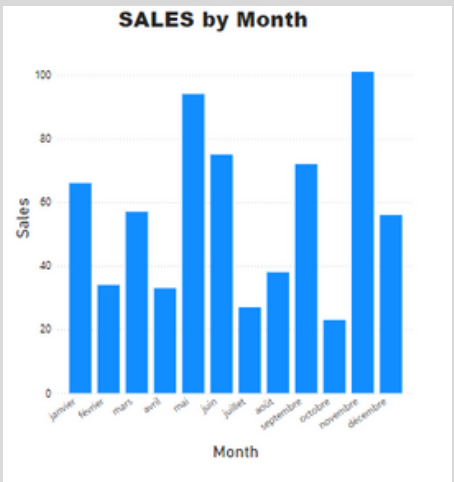


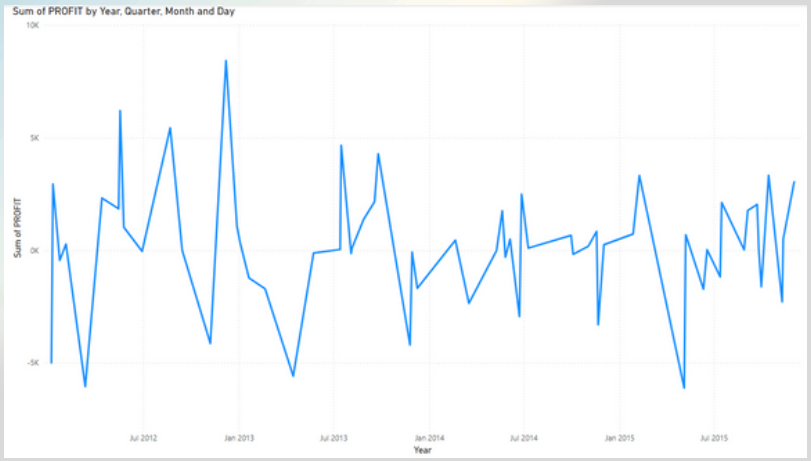
The chart shows sales are spread across states, with California, Texas, Florida, and New York each contributing 4.4%, and other states at 2.47%. Focus on strong markets while boosting sales in lower-performing states to grow overall sales.

Time Period Analysis

- TO DETERMINE WHICH ARE THE MOST PROFITABLE PERIODS FOR THE COMPANY

This graph displays monthly sales figures, with november reaching a peak of 100, while other months show lower sales.This indicates a clear variation in sales performance across months, highlighting potential seasonal trends.



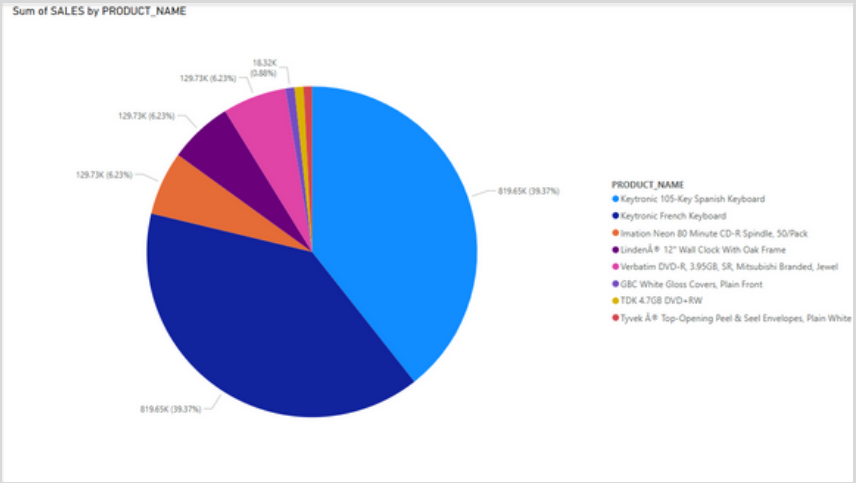


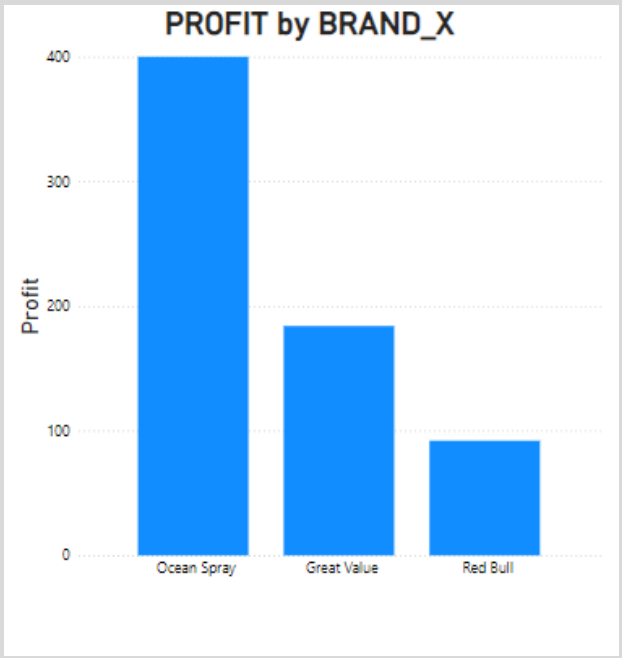
The chart shows profit going up and down over time, with some periods making money and others losing it. This could be due to market changes, costs, or demand. Studying these trends and their causes can help make profits more steady.

Product Analysis

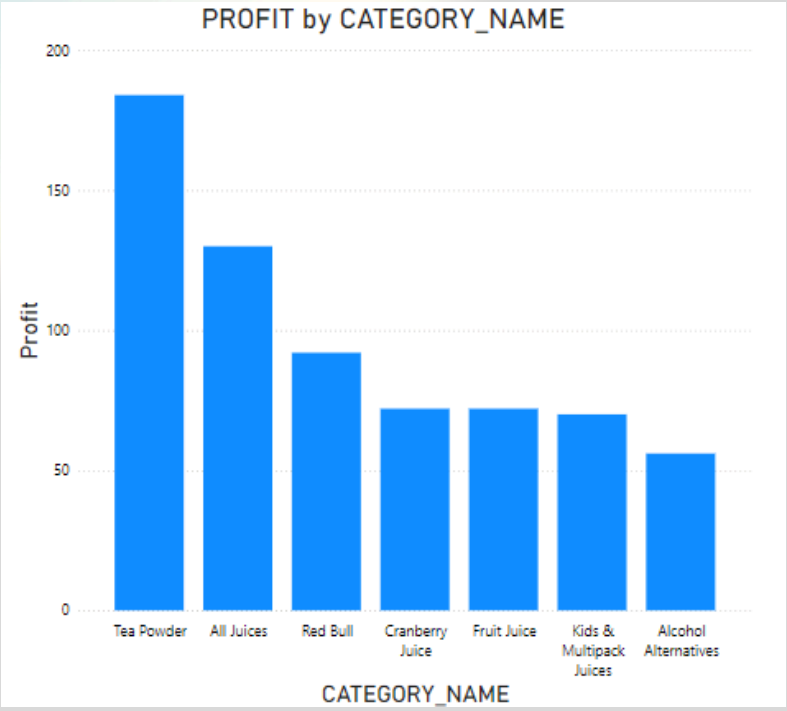
- TO DETERMINE WHICH PRODUCTS ARE THE MOST PROFITABLE FOR THE COMPANY

The chart shows that most sales come from two main products: the Keytronic 105-Key Spanish Keyboard and the Keytronic French Keyboard. The company should think about adding more products or boosting sales of less popular ones.





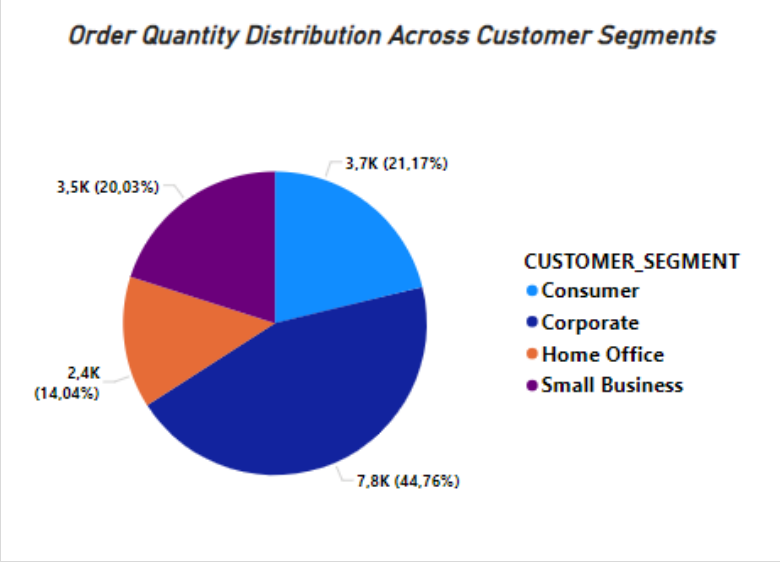
Ocean Spray leads significantly with the highest profit, nearly double that of Great Value, while Red Bull has the lowest value. This suggests Ocean Spray's strong performance compared to the other brands, the company should think about selling more or promoting other products.

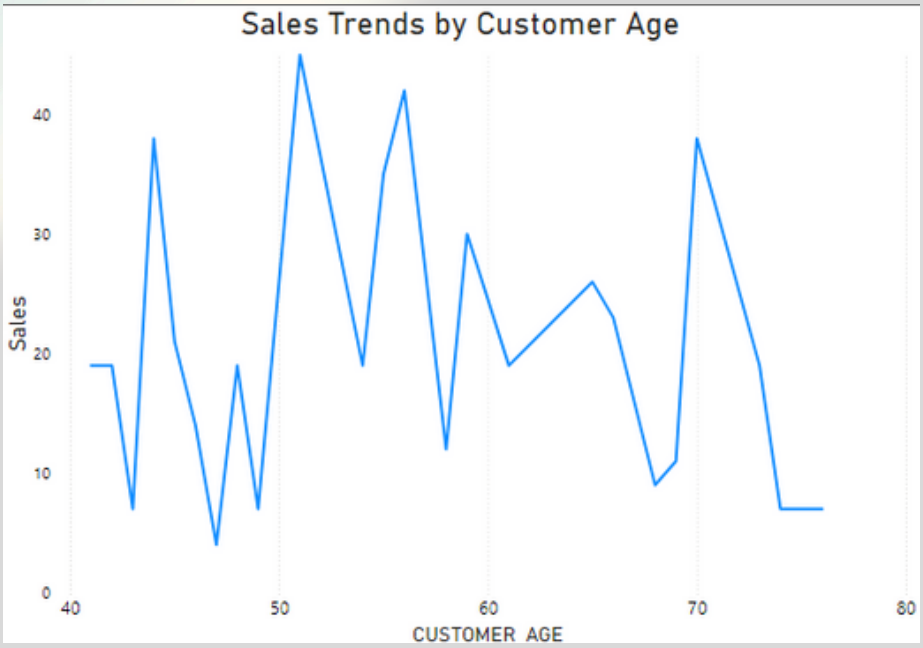


The graph displays profit across different product categories, with values ranging from 0 to 200. Categories like Tea Powder and all Juices show higher profits, while others like Kids & Akchel Juices and Multipack Alternatives have lower or minimal profits. This visualization highlights which categories are the most profitable and which may require strategic improvements or reevaluation to enhance their performance.

Customer Profile Analysis

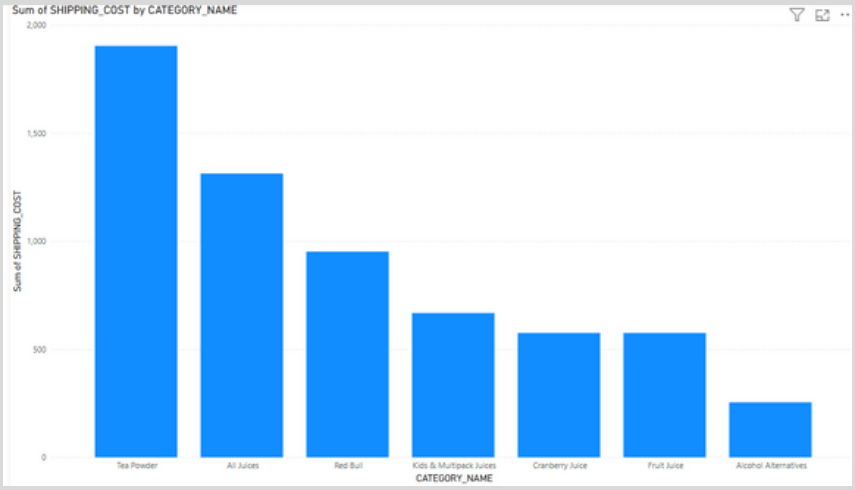
The Corporate segment dominates with 44.76% of total orders, followed by Consumer at 21.17%. Small Business and Home Office contribute 20.03% and 14.04%, respectively, indicating Corporate and Consumer are key drivers of sales.





The graph shows sales distribution across age groups. The highest sales are in the 52-53 age group, with a gradual decline as age increases. 80+ groups have the lowest sales. This suggests that younger customers drive the most sales, indicating a potential focus area for marketing efforts.

Cost Analysis



This bar chart illustrates the total shipping costs by category. "Tea Powder" incurs the highest shipping cost, exceeding 2,000 units, followed by "All Juices" and "Red Bull." Categories like "Alcohol Alternatives" have significantly lower shipping costs, indicating a disparity in shipping expenses across different product categories. The company should reduce shipping costs for "Tea Powder" by improving packaging or rates. Promoting cheaper-to-ship items can also help save money..

The graph shows shipping costs ranging from 60 to 200 across locations, highlighting significant variations.
This visualization helps identify high-cost shipping locations, which could benefit from further analysis to optimize logistics and reduce expenses.

ANY QUESTIONS?

Project Github Repo

EMAILS:
saif.chouaya@tbs.u-tunis.tn
e.hafsii@gmail.com
fadhlaouirayen@gmail.com
Saydaouaddar@gmail.com