

# Sales Data Analysis Documentation

## Project Overview

**Project Name:** Sales Data Analysis

**Project Data Source:** MeriSkill

**Project Creator:** Prakhar Tripathi

## Purpose

The purpose of this project is to analyze sales data in order to identify trends, top-selling products, and revenue metrics for making informed business decisions.

## Description

In this project, we dive into a large sales dataset sourced from MeriSkill to extract valuable insights. Through data analysis, we aim to accomplish the following tasks:

- Explore sales trends over time.
- Identify the best-selling products.
- Calculate revenue metrics such as total sales and profit margins.
- Create visualizations to effectively present findings.

This project showcases the ability to manipulate and derive insights from large datasets, enabling data-driven recommendations for optimizing sales strategies.

# Data Analysis Process

## 1. Importing Libraries and Dataset

```
import pandas as pd
import matplotlib.pyplot as plt
```

Importing the sales dataset

```
# Importing the sales dataset
data = pd.read_csv("C:\\Users\\prakh\\Downloads\\Practice DataSet\\Sales Data.csv")
```

## 2. Data Exploration

```
# checking the dataset
print(data.head())

# Checking the column of the dataset
print(data.columns)

# Removing the unused columns from dataset
data.drop(labels=["Unnamed: 0", "Hour", "Month", "Order ID"], axis=1, inplace=True)
print(data.columns)

# Checking the summary of data
print(data.describe())

# Checking the null values in dataset
print(data.info())
```

Output

```
"C:\Users\prakh\Downloads\Pycharm Projects\venv\Scripts\python.exe" "C:\Users\prakh\Downloads\Pycharm Projects\Sales Data Analysis\main.py"
  Unnamed: 0  Order ID      Product  ...  Sales      City Hour
0           0    295665  Macbook Pro Laptop  ...  1700.00  New York City    0
1           1    295666    LG Washing Machine  ...   600.00  New York City    7
2           2    295667  USB-C Charging Cable  ...   11.95  New York City   18
3           3    295668    27in FHD Monitor  ...  149.99  San Francisco   15
4           4    295669  USB-C Charging Cable  ...   11.95    Atlanta   12

[5 rows x 11 columns]
Index(['Unnamed: 0', 'Order ID', 'Product', 'Quantity Ordered', 'Price Each',
      'Order Date', 'Purchase Address', 'Month', 'Sales', 'City', 'Hour'],
      dtype='object')
Index(['Product', 'Quantity Ordered', 'Price Each', 'Order Date',
      'Purchase Address', 'Sales', 'City'],
      dtype='object')
      Quantity Ordered  Price Each      Sales
count  185950.000000  185950.000000  185950.000000
mean      1.124383    184.399735    185.490917
std       0.442793    332.731330    332.919771
min       1.000000     2.990000     2.990000
25%       1.000000    11.950000    11.950000
50%       1.000000    14.950000    14.950000
75%       1.000000    150.000000    150.000000
max       9.000000   1700.000000   3400.000000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
```

```
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Product                185950 non-null  object
1   Quantity Ordered       185950 non-null  int64
2   Price Each             185950 non-null  float64
3   Order Date             185950 non-null  object
4   Purchase Address       185950 non-null  object
5   Sales                  185950 non-null  float64
6   City                   185950 non-null  object
dtypes: float64(2), int64(1), object(4)
memory usage: 9.9+ MB
None
```

### 3. Data Preprocessing

```
# Changing the datatype of Order Date
data["Order Date"] = pd.to_datetime(data["Order Date"])
```

### 4. Data Visualization

```
Plotting a graph that show the count of products by cities
product_count = data.groupby("City").size()
plt.figure(figsize=(18, 7))
plt.bar(product_count.index, product_count.values, color="lightgreen")
plt.xlabel("City")
plt.ylabel("Count of Products")
plt.title("Count of products by cities")
plt.show()
```

```
# Sales of all Products
unique_products = data["Product"].unique()
product_price = data.groupby('Product')['Price Each'].sum()
plt.figure(figsize=(18, 7))
plt.bar(unique_products, product_price, color="lightblue")
plt.xlabel("Products")
plt.xticks(rotation='vertical')
plt.ylabel("Total amount in Millions")
plt.title("Sales of all Products")
plt.show()
```

```
# Maximum Quantity ordered by customers
quantity = data["Quantity Ordered"].value_counts()
plt.figure(figsize=(18, 7))
plt.bar(quantity.index, quantity.values, color="yellow")
plt.xlabel("Number of Quantity")
plt.ylabel("Count")
plt.title("Quantity ordered by customers")
plt.show()
```

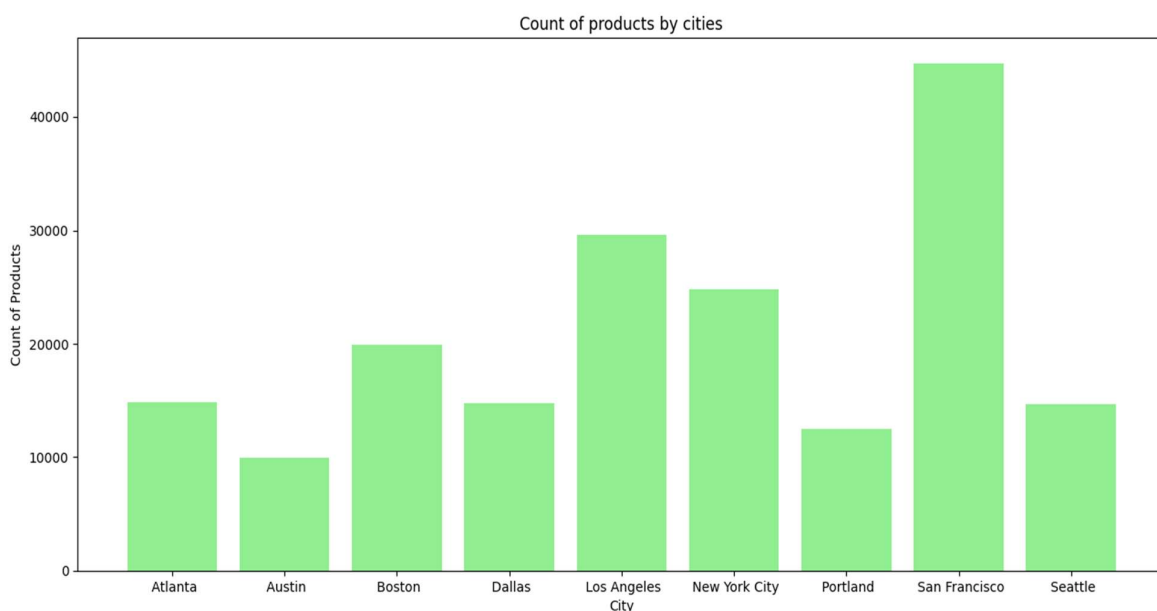
```

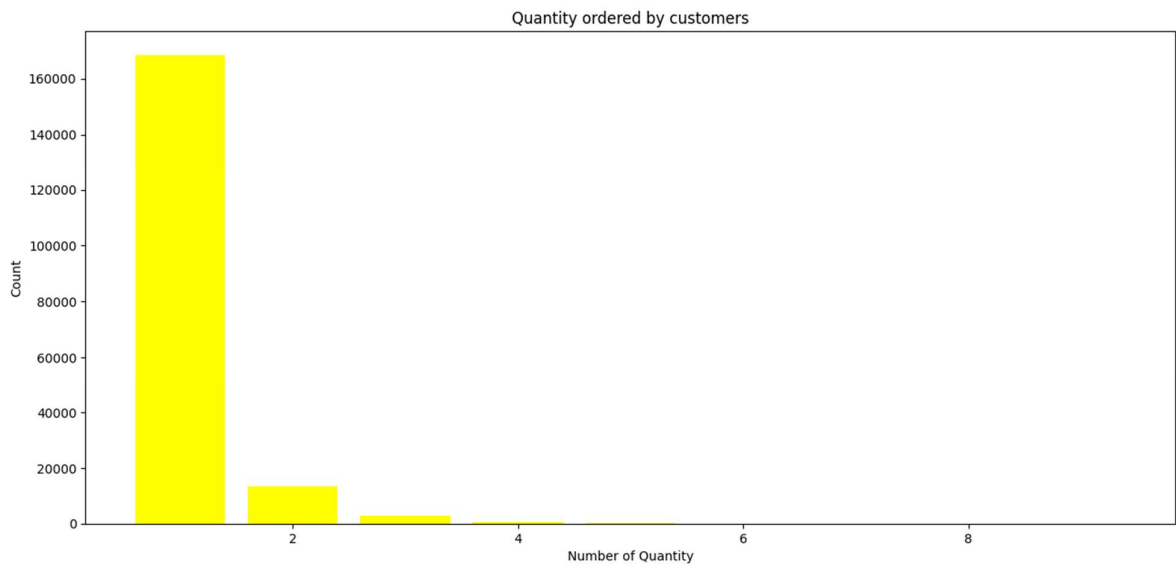
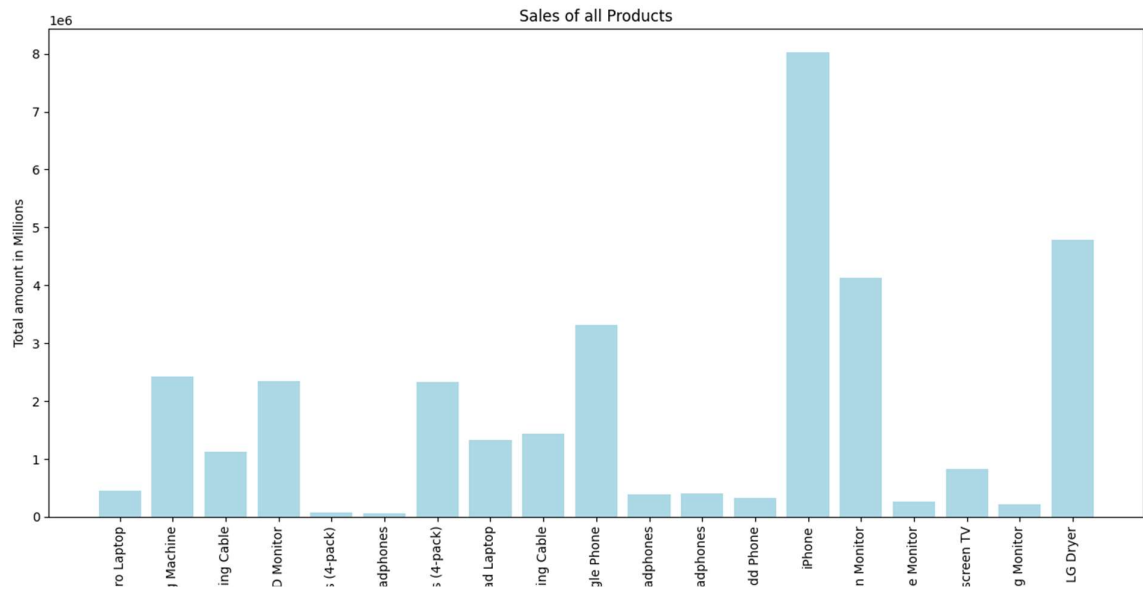
Sales in Cities
sales_count = data.groupby('City')['Sales'].size()
plt.figure(figsize=(18, 7))
plt.plot(*args: sales_count.index, sales_count.values, color="orange")
plt.xlabel("City")
plt.ylabel("Sales Count")
plt.title("Sales count per city")
plt.show()

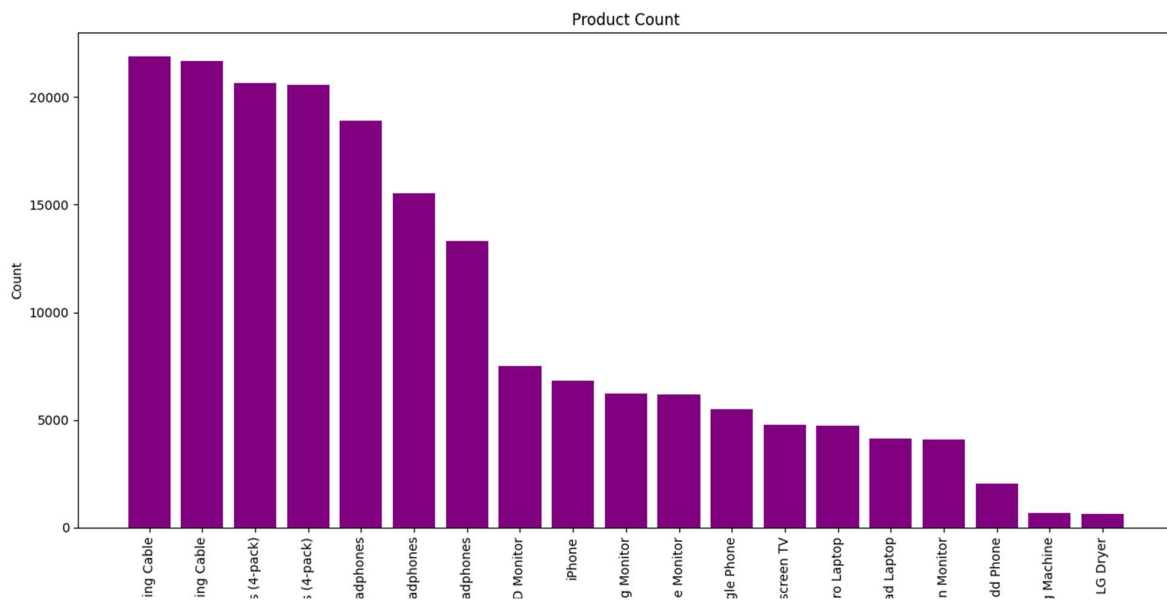
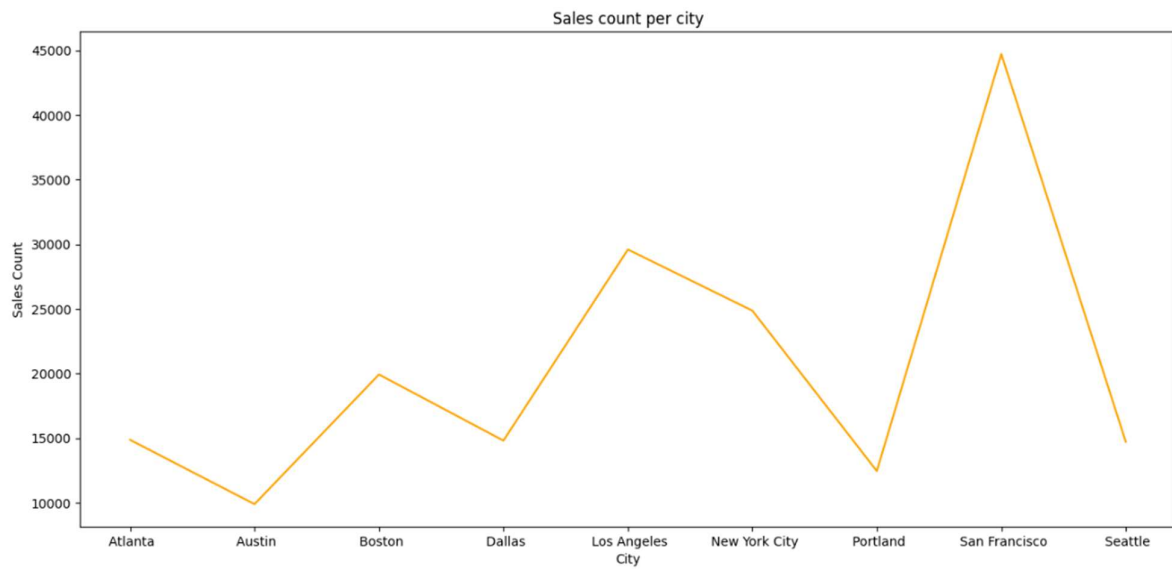
# Product Counts
products = data["Product"].value_counts()
plt.figure(figsize=(18, 7))
plt.bar(products.index, products.values, color="purple")
plt.xlabel("Products")
plt.xticks(rotation='vertical')
plt.ylabel("Count")
plt.title("Product Count")
plt.show()

```

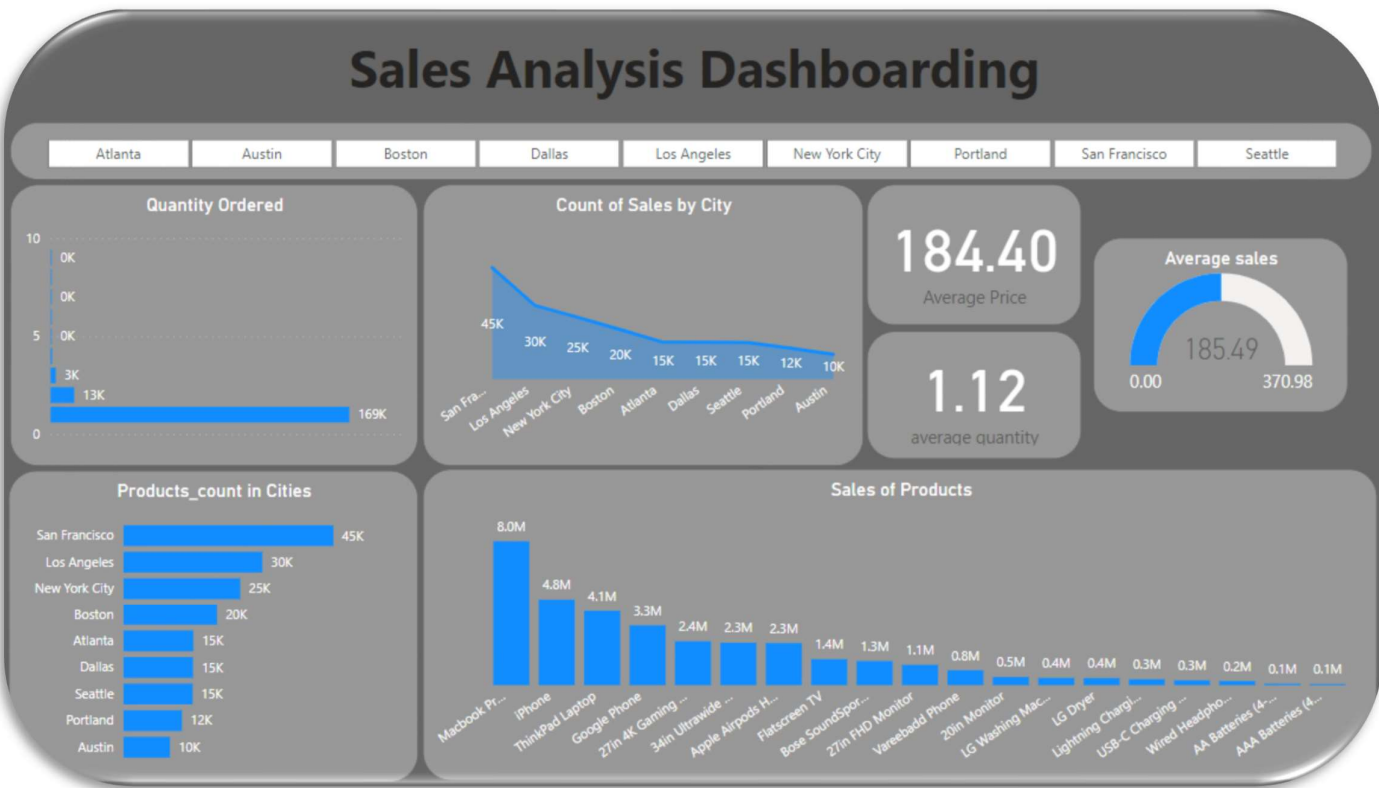
## Output







5. Dashboard



Insights

Sales Trends by City:

The bar chart depicting the count of products by cities reveals the distribution of sales across different locations.

Insight: It helps in identifying cities with higher sales activity and potential areas for targeted marketing or expansion efforts.

Top-Selling Products:

The bar chart showcasing the sales of all products provides an overview of the revenue generated by each product.

Insight: It identifies the top-selling products contributing the most to the overall revenue, enabling inventory management and marketing focus.



### **Order Quantity Distribution:**

The bar chart illustrating the quantity ordered by customers displays the distribution of order quantities.

Insight: Understanding the common order quantities helps in optimizing inventory levels and forecasting demand accurately.

### **Sales Distribution by City:**

The line plot representing sales count per city illustrates the distribution of sales across different cities.

Insight: It highlights cities with higher sales volumes, allowing for targeted marketing campaigns and resource allocation.

### **Product Popularity:**

The bar chart displaying product counts showcases the popularity of each product based on the number of orders.

Insight: It identifies products with high demand, helping in decision-making related to product promotions, pricing, and inventory management.

### **Overall Sales Performance:**

Summary statistics provide insights into the overall sales performance, including total sales, average sales, and other relevant metrics.

Insight: It offers a comprehensive understanding of the sales data, enabling comparison with targets and historical performance to assess business growth and profitability.

### **Conclusion**

This documentation outlines the process of analysing sales data to derive valuable insights for business decision-making. Through data exploration, preprocessing, and visualization techniques, we were able to identify sales trends, top-selling products, and other relevant metrics. These insights can be utilized to optimize sales strategies and enhance overall business performance.