📊 Superstore Sales Data Analysis Project

Business Intelligence & Exploratory Data Analysis using Python

---

🎯 Project Objective

This project analyzes 3 years of retail sales data to answer key business questions:

1. What is the overall sales trend?
2. Which are the Top 10 products by sales?
3. Which are the Most Selling Products (by quantity)?
4. Which is the most preferred Ship Mode?
5. Which are the Most Profitable Category and Sub-Category?

🛠 Tools & Libraries Used

- Python
- Pandas
- Matplotlib
- Seaborn
- Google Colab

STEP 1: Import Required Libraries

import pandas as pd import matplotlib.pyplot as plt import seaborn as sns
%matplotlib inline

📂 STEP 2: Upload & Load Dataset

```
from google.colab import files
uploaded = files.upload()

df = pd.read_excel(list(uploaded.keys())[0])

df.head()
```

Choose Files superstore_sales.xlsx
superstore_sales.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 6888951 bytes, last modified: 2/14/2026 - 100% done
Saving superstore_sales.xlsx to superstore_sales (2).xlsx

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	region	...	category
0	AG-2011-2040	2011-01-01	2011-01-06	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa	Africa	...	Office Supplies
1	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Office Supplies
2	HU-2011-1220	2011-01-01	2011-01-05	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA	EMEA	...	Office Supplies
3	IT-2011-3647632	2011-01-01	2011-01-05	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU	North	...	Office Supplies
4	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Furniture

5 rows × 21 columns

🧹 STEP 3: Data Cleaning & Standardization

To avoid column name errors, we standardize all column names.

```
# Standardizing column names
df.columns = (
    df.columns
    .str.strip()
```

```
    .str.lower()
    .str.replace(" ", "_")
)

print(df.columns)
```

```
Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
       'segment', 'state', 'country', 'market', 'region', 'product_id',
       'category', 'sub_category', 'product_name', 'sales', 'quantity',
       'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
      dtype='object')
```

🔍 STEP 4: Data Audit

Understanding the dataset structure before analysis.

```
df.shape
df.info()
df.describe()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   order_id        51290 non-null  object
 1   order_date      51290 non-null  datetime64[ns]
 2   ship_date       51290 non-null  datetime64[ns]
 3   ship_mode       51290 non-null  object
 4   customer_name   51290 non-null  object
 5   segment         51290 non-null  object
 6   state           51290 non-null  object
 7   country         51290 non-null  object
 8   market          51290 non-null  object
 9   region          51290 non-null  object
 10  product_id      51290 non-null  object
 11  category        51290 non-null  object
 12  sub_category    51290 non-null  object
 13  product_name    51290 non-null  object
 14  sales           51290 non-null  float64
 15  quantity        51290 non-null  int64
 16  discount        51290 non-null  float64
 17  profit          51290 non-null  float64
 18  shipping_cost   51290 non-null  float64
 19  order_priority  51290 non-null  object
 20  year            51290 non-null  int64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
memory usage: 8.2+ MB
```

	0
order_id	0
order_date	0
ship_date	0
ship_mode	0
customer_name	0
segment	0
state	0
country	0
market	0
region	0
product_id	0
category	0
sub_category	0
product_name	0
sales	0
quantity	0
discount	0
profit	0
shipping_cost	0
order_priority	0
year	0

dtype: int64

🗓️ STEP 5: Date Conversion & Feature Engineering

We extract year and month to analyze trends over time.

```
df["order_date"] = pd.to_datetime(df["order_date"])

df["year"] = df["order_date"].dt.year
df["month"] = df["order_date"].dt.month
df["month_name"] = df["order_date"].dt.month_name()

df.head()
```

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	region	...	product_na
0	AG-2011-2040	2011-01-01	2011-01-06	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa	Africa	...	Ter Lockers, Bl
1	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Acı Trimmer, Hi Spe
2	HU-2011-1220	2011-01-01	2011-01-05	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA	EMEA	...	Tenex B Single Wi
3	IT-2011-3647632	2011-01-01	2011-01-05	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU	North	...	Enermax N Car Premi
4	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Eldon Li Bulb, D Pa

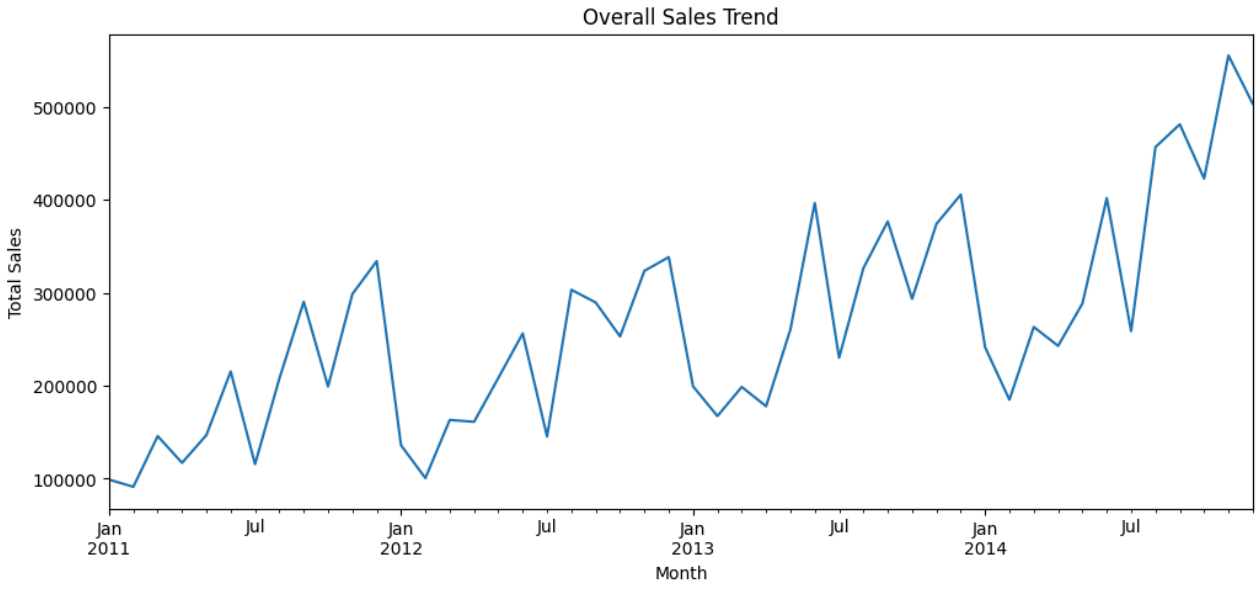
5 rows × 23 columns

📈 Business Question 1

What is the Overall Sales Trend?

```
monthly_sales = (
    df.groupby(df["order_date"].dt.to_period("M"))["sales"]
    .sum()
)

monthly_sales.plot(figsize=(12,5))
plt.title("Overall Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.show()
```



🔎 Insight:

Sales show a growing trend over the years, indicating business expansion and seasonal patterns.

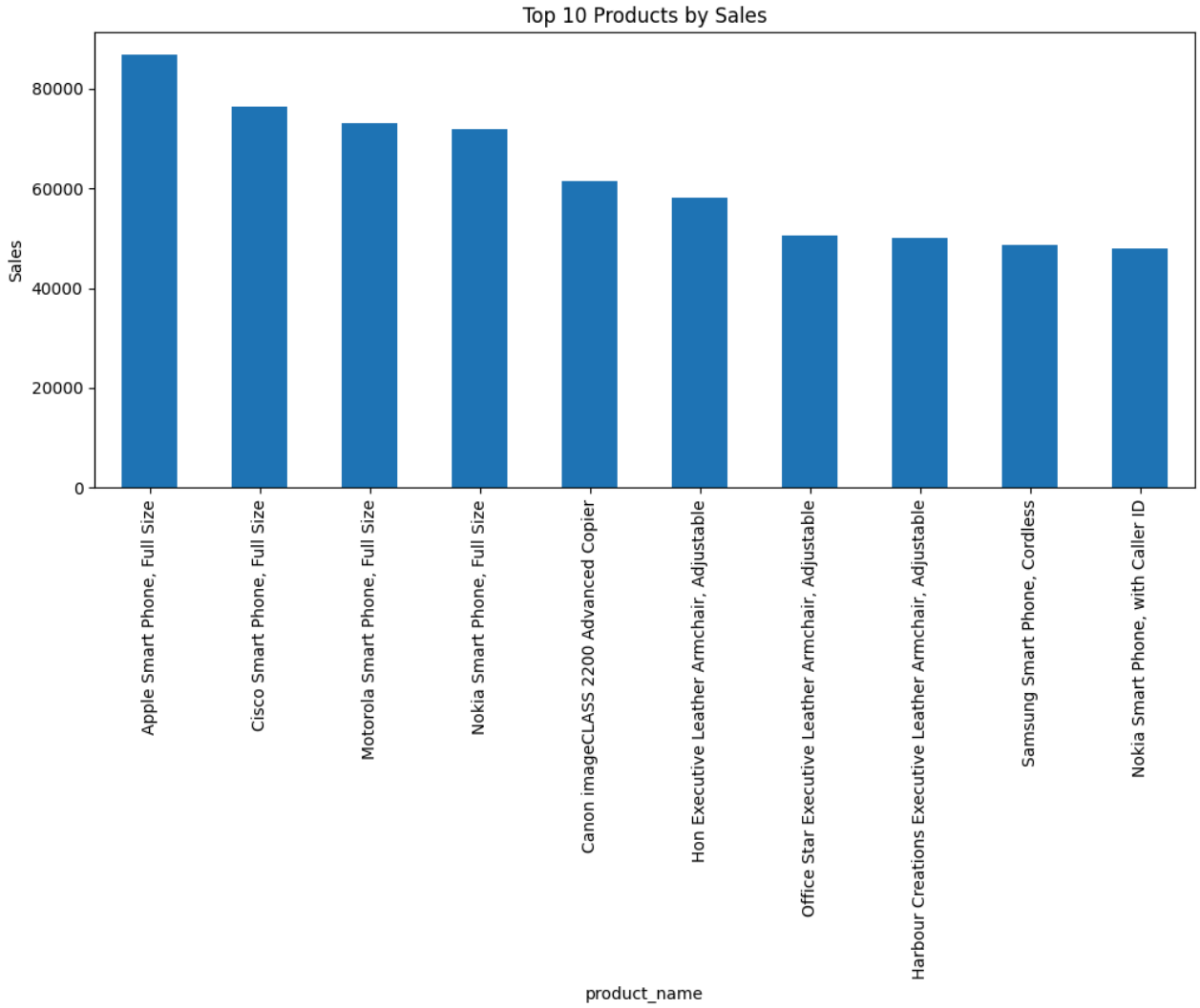
📊 Business Question 2

Top 10 Products by Sales

```
top_products = (
    df.groupby("product_name")["sales"]
```

```
    .sum()
    .sort_values(ascending=False)
    .head(10)
)

top_products.plot(kind="bar", figsize=(12,5))
plt.title("Top 10 Products by Sales")
plt.ylabel("Sales")
plt.show()
```

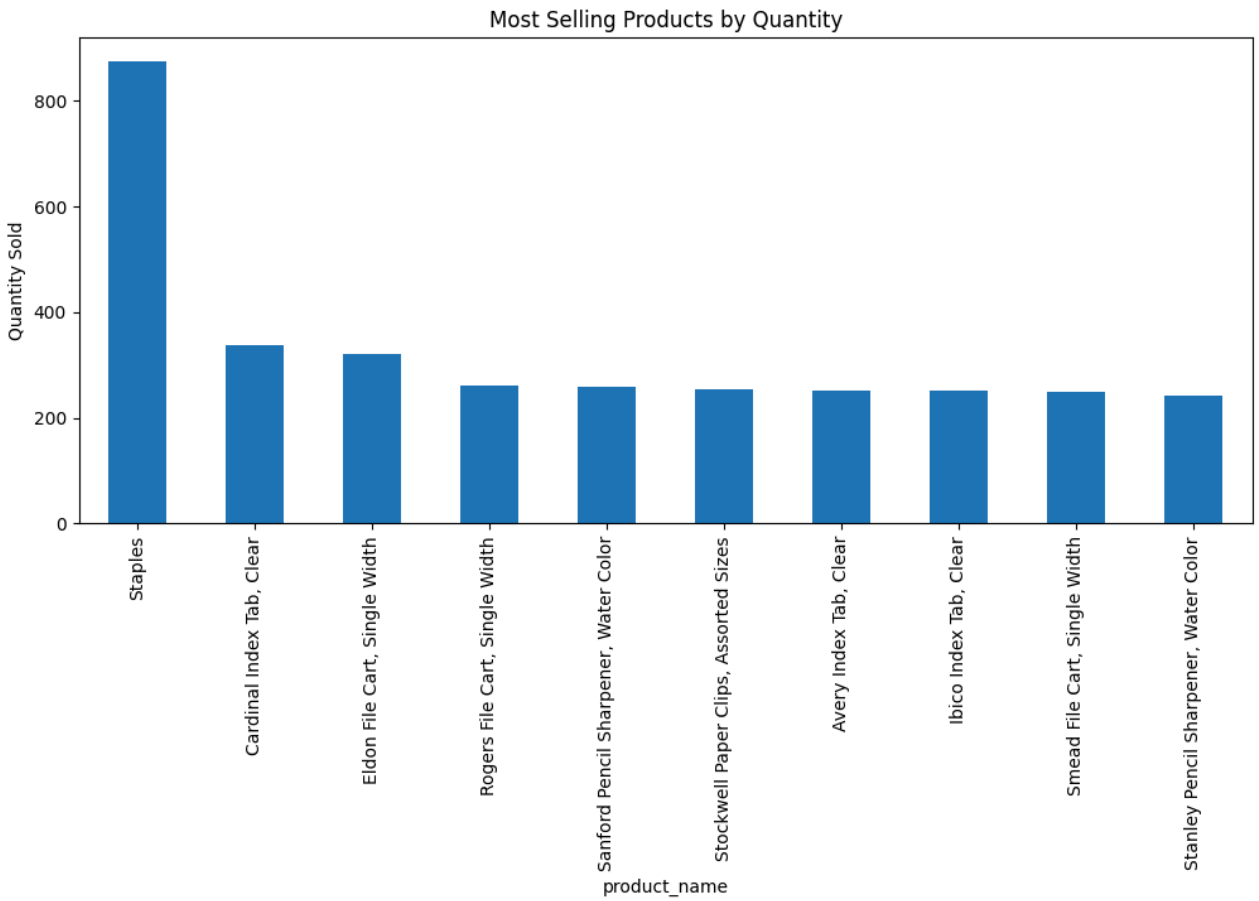


🔎 Insight:

These products generate the highest revenue and are key drivers of overall sales performance.

```
most_selling = (
    df.groupby("product_name")["quantity"]
    .sum()
    .sort_values(ascending=False)
    .head(10)
)

most_selling.plot(kind="bar", figsize=(12,5))
plt.title("Most Selling Products by Quantity")
plt.ylabel("Quantity Sold")
plt.show()
```



🔎 Insight:

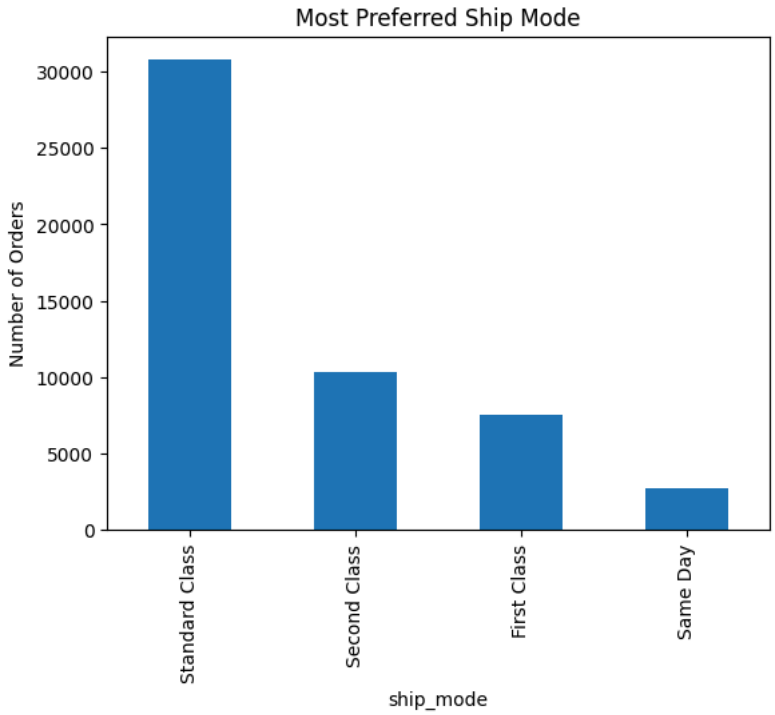
High quantity products are not always the highest revenue products, indicating pricing differences.

🚚 Business Question 4

Most Preferred Ship Mode

```
ship_mode_counts = df["ship_mode"].value_counts()

ship_mode_counts.plot(kind="bar")
plt.title("Most Preferred Ship Mode")
plt.ylabel("Number of Orders")
plt.show()
```



🔎 Insight:

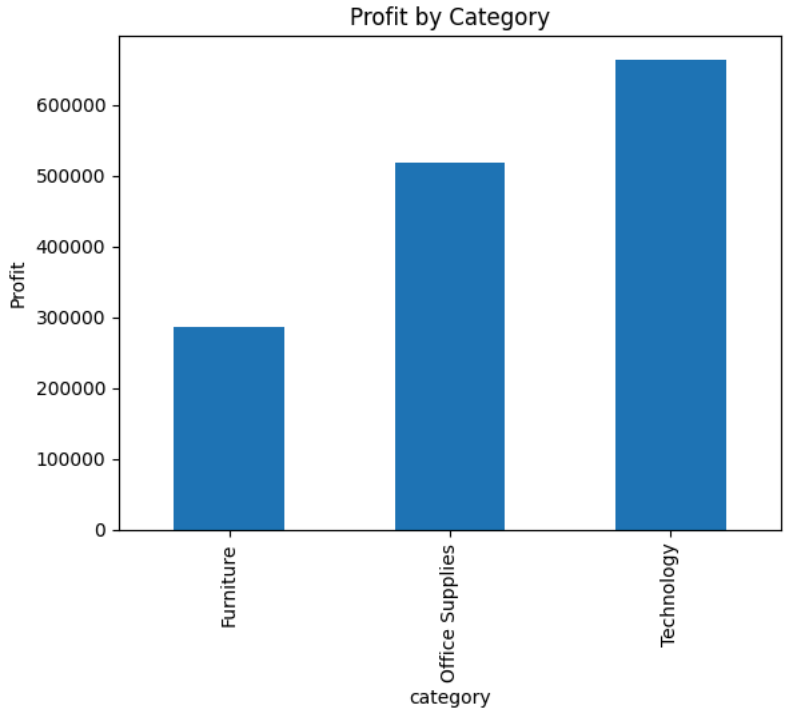
Understanding shipping preference helps optimize logistics strategy and cost management.

💰 Business Question 5

Most Profitable Category & Sub-Category

```
category_profit = df.groupby("category")["profit"].sum()

category_profit.plot(kind="bar")
plt.title("Profit by Category")
plt.ylabel("Profit")
plt.show()
```



```
sub_category_profit = (
    df.groupby("sub_category")["profit"]
    .sum()
    .sort_values(ascending=False)
)

sub_category_profit.plot(kind="bar", figsize=(14,5))
plt.title("Profit by Sub-Category")
plt.ylabel("Profit")
plt.show()
```