Exploratory Data Analysis (EDA) Report: Retail Sales Analysis

Exploratory Data Analysis (EDA) is an essential step in data science to understand datasets, discover patterns, identify anomalies, and extract meaningful insights. In this project, we perform an end-to-end Exploratory Data Analysis on a Retail Sales Dataset sourced from Kaggle. The analysis involves data loading, cleaning, transformation, visualization, and interpretation to support business decision-making.

Procedure: Exploratory Data Analysis

1. Understand the Problem and the Data

2. Import and Inspect the Data

3. Handling Missing Values and Duplicates

4. Explore Data Characteristics

5. Perform Data Transformation

6. Visualize Data Relationships

7. Handling Outliers

8. Communicate Findings and Insights

Dataset Name: retail_sales.csv

Source: Kaggle (Public Dataset)

Project Structure

```
EDA_PROJECT/
│── data/
│   └── retail_sales5.csv
│── modules/
│   ├── data_import.py
│   ├── data_cleaning.py
│   ├── transformation.py
│   ├── stats_analysis.py
│   ├── visualization.py
│   └── modeling.py
│── outputs/
│   └── retail_sales_backup.csv
│── main.py
```

1. Understand the Problem and the Data

The objective of this project is to analyze retail sales data to identify relationships between Quantity, Sales, Profit, and Category. The analysis helps in understanding sales distribution, profitability trends, and category-wise performance.

Variables in the Dataset:

Numerical Variables: Quantity, Sales, Profit

Categorical Variables: Category

2. Import and Inspect the Data

File Name: data_import.py

The dataset is loaded into a Pandas DataFrame using the read_csv() function. This step allows efficient access and manipulation of the retail data.

The load_data() function performs the following tasks:

Displays the first five rows using head()

Displays dataset structure using info()

Helps identify data types and missing values

This step ensures familiarity with the dataset before proceeding further.

```
First Five Rows of Dataset
   Row ID        Order ID  Order Date   Ship Date  ...     Sales Quantity Discount    Profit
0       1  CA-2016-152156   11/8/2016  11/11/2016  ...  261.9600        2     0.00   41.9136
1       2  CA-2016-152156   11/8/2016  11/11/2016  ...  731.9400        3     0.00  219.5820
2       3  CA-2016-138688   6/12/2016   6/16/2016  ...   14.6200        2     0.00    6.8714
3       4  US-2015-108966  10/11/2015  10/18/2015  ...  957.5775        5     0.45 -383.0310
4       5  US-2015-108966  10/11/2015  10/18/2015  ...   22.3680        2     0.20    2.5164
```

```
[5 rows x 21 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
None
                                                                 Ln 11, Col 1   Spaces: 4
```

3. Handling Missing Values and Duplicates

File Name: data_cleaning.py

3.1 Detection of Missing Values

Missing values were identified using isnull().sum(). Any missing numerical values in Quantity, Sales, or Profit were handled appropriately.

3.2 Mean Imputation

Missing numerical values were replaced with the mean of their respective columns to preserve data distribution.

3.3 Duplicate Removal

Duplicate records were checked and removed using drop_duplicates() to maintain data integrity.

3.4 Final Dataset

After cleaning, the dataset became consistent and free from missing values and duplicates. A cleaned backup file was saved as retail_sales_backup.csv.

```
Cleaned Dataset Saved as retail_sales_backup.csv
```

4. Explore Data Characteristics

File Name: stats_analysis.py

Descriptive statistics were calculated to understand the distribution and variability of retail data.

The following statistical measures were computed:

- **Mean** – Average value of the dataset
- **Median** – Middle value when data is arranged in order
- **Mode** – Most frequently occurring value
- **Standard Deviation** – Measure of how spread out the values are
- **Minimum** – Smallest value in the dataset
- **Maximum** – Largest value in the dataset

Sales and Profit showed moderate variability, indicating different performance levels across transactions and categories.

```
Descriptive Statistics
          Quantity         Sales       Profit
count  9994.000000   9994.000000  9994.000000
mean      3.789574    229.858001    28.656896
std       2.225110    623.245101   234.260108
min       1.000000      0.444000 -6599.978000
25%       2.000000     17.280000     1.728750
50%       3.000000     54.490000     8.666500
75%       5.000000    209.940000    29.364000
max      14.000000  22638.480000  8399.976000
```

5. Perform Data Transformation

File Name: transformation.py

To prepare the data for further analysis, numerical features were standardized.

Standardization

Quantity, Sales, and Profit were scaled using StandardScaler

This ensures fair comparison between features with different units

```
Stardardized Data
[[-0.8043034   0.0515104   0.05659251]
 [-0.35486486  0.80563348  0.81505408]
 [-0.8043034  -0.34536777 -0.09300169]
 ...
 [-0.8043034   0.04608048 -0.03954647]
 [ 0.09457367 -0.32133108 -0.06547279]
 [-0.8043034   0.02134419  0.18907752]]
```

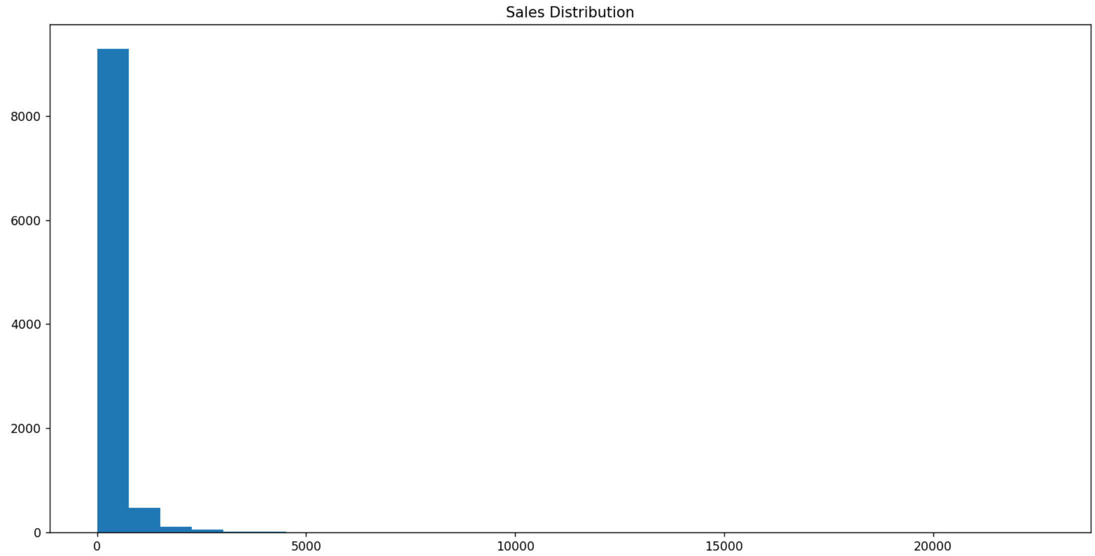
6. Visualize Data Relationships

File Name: visualization.py

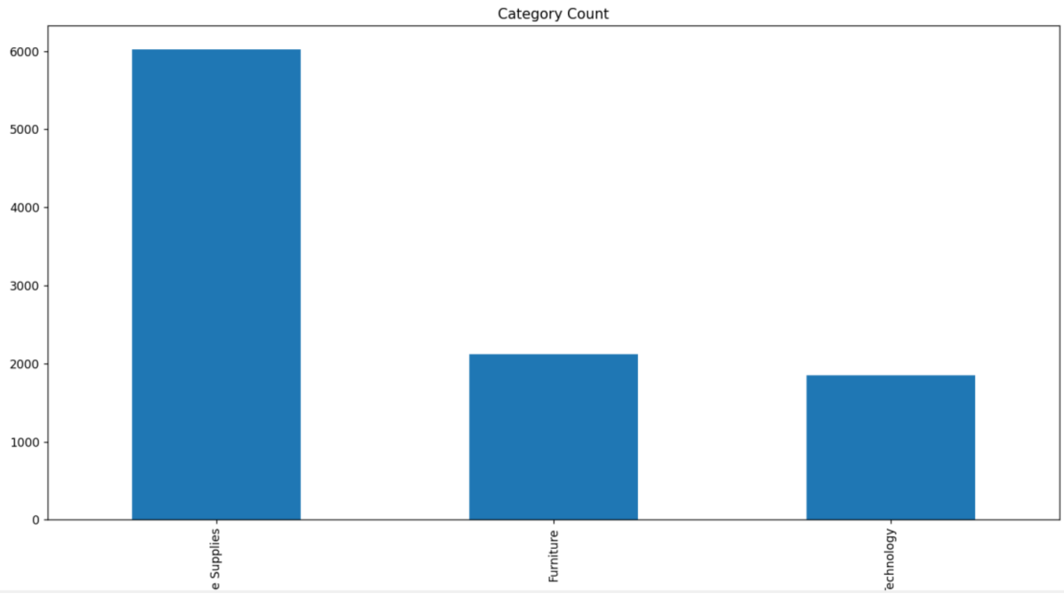
Visualization techniques were used to better understand sales trends and relationships.

Univariate Analysis

Histogram: Distribution of Sales and Profit

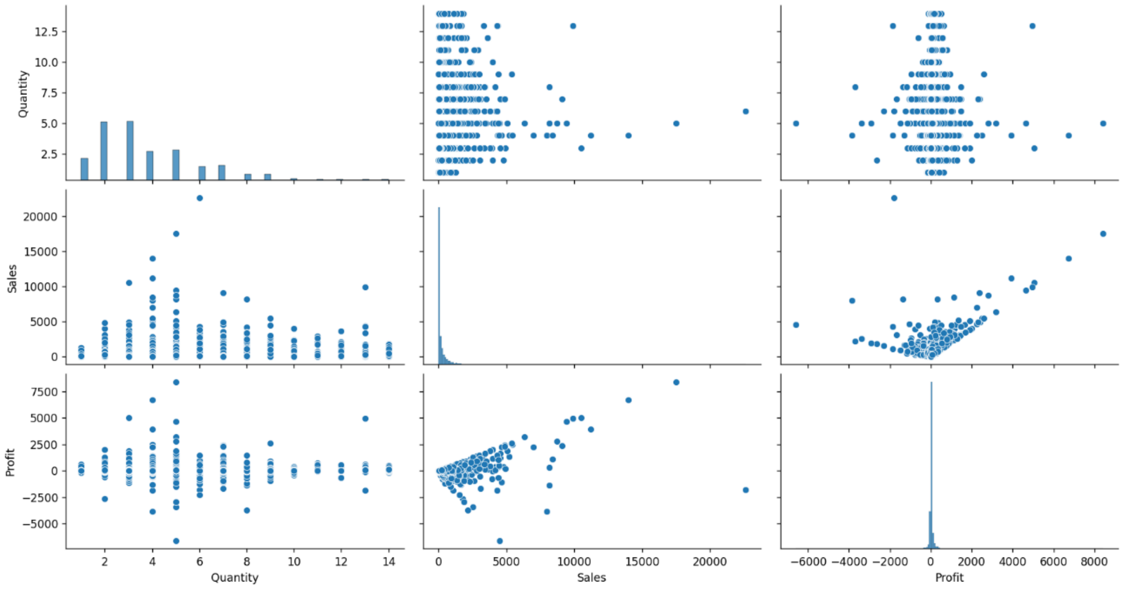


Bar Chart: Category-wise sales count

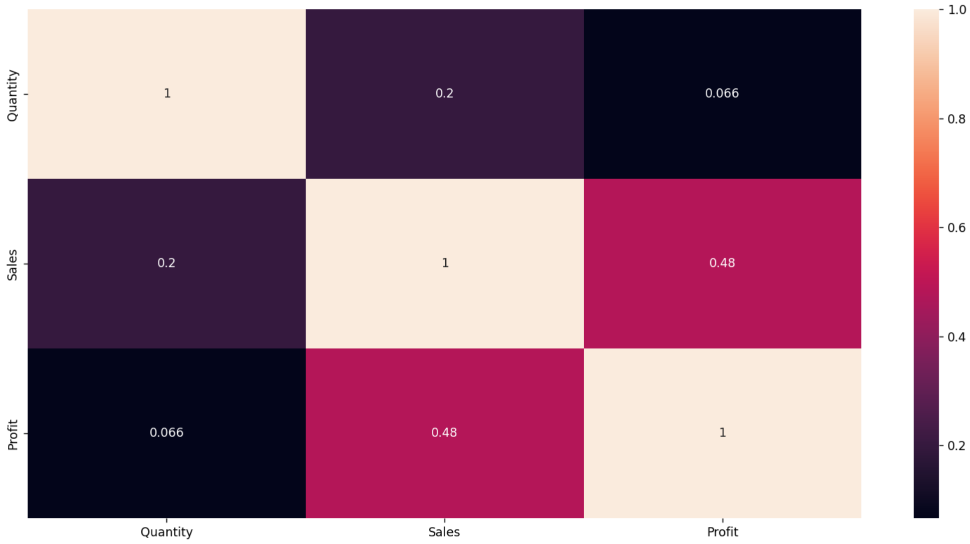


Multivariate Analysis

Pair Plot: Relationship among Quantity, Sales, and Profit



Correlation Heatmap: Strength of relationships between numerical variables



7. Handling Outliers

Outliers were identified using box plots and statistical thresholds. Extreme values in Sales and Profit were reviewed to avoid skewed analysis.

Logic:

Calculate Q1(25th percentile) and Q3(75th percentile).

Define bounds: Lower = Q1 - 1.5 *IQR; Upper = Q3 + 1.5*IQR

	Quantity	Sales	Profit
count	9994.000000	9994.000000	9994.000000
mean	3.789574	229.858001	28.656896
std	2.225110	623.245101	234.260108
min	1.000000	0.444000	-6599.978000
25%	2.000000	17.280000	1.728750
50%	3.000000	54.490000	8.666500
75%	5.000000	209.940000	29.364000
max	14.000000	22638.480000	8399.976000

8. Communicate Findings and Insights

The final step involved summarizing insights derived from the analysis.

Key Insights

Sales and Profit show a strong positive correlation

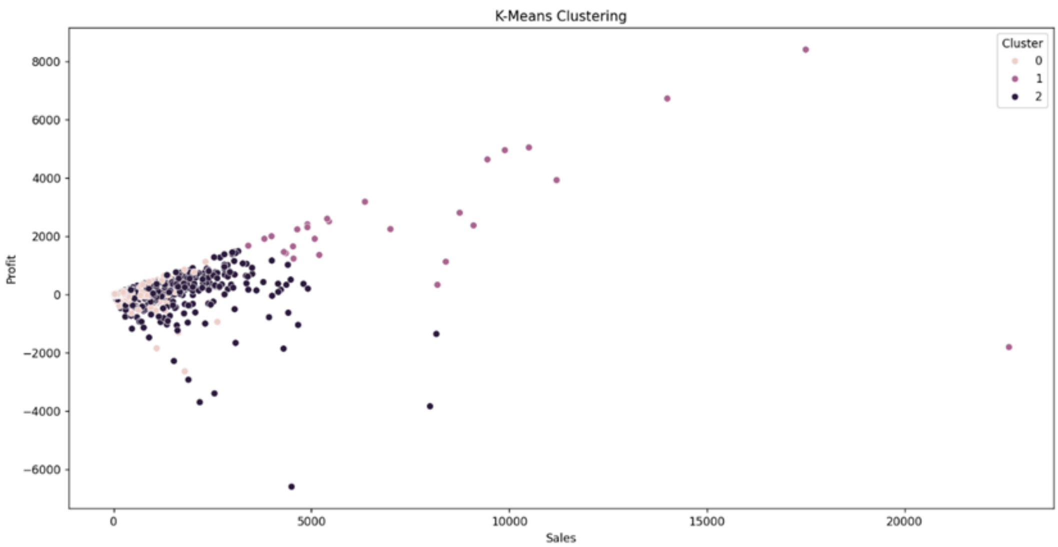
Higher quantity generally leads to increased sales

Category-wise analysis reveals uneven sales distribution

These insights can help businesses optimize pricing strategies, inventory planning, and category management.

8.1 k-means clustering

K-Means clustering is an unsupervised machine learning algorithm used to group data into clusters based on similarity. In this project, K-Means is applied to retail sales data to identify meaningful customer or product groups using numerical features.



CONCLUSION:

This project successfully demonstrates a complete Exploratory Data Analysis workflow on a Retail Sales dataset. By applying data cleaning, statistical analysis, and visualization techniques, meaningful insights were extracted to support data-driven decision-making. The project highlights the importance of EDA in understanding business data and improving retail performance.