**Exploratory Data Analysis (EDA) with pandas in**

**Retail Store Inventory**

The purpose of this project is to explore and analyse an retail store inventory dataset using the Pandas framework to derive insights into customer behaviour, product trends, and sales performance.

**Goals of the Project:**

● Explore the retail store inventory dataset using Pandas.

● Perform feature engineering to derive useful insights.

● Visualize data distributions and trends with various plot types.

● Summarize key findings that can aid in business decision-making.

**Materials and Methods**

The data for this project is from a simulated retail store inventory platform, containing information about date, products id, regions, and ordered details. This dataset includes date, product id, store id, category, region, inventory level, units sold, units ordered, demand forecasting, price, discount, weather condition and more. The analysis aims to understand sales performance, product trends, and units ordered.

**General Part**

● **Libraries Import**: Pandas, NumPy, Seaborn, Matplotlib

● **Dataset Exploration**: Initial exploration of the dataset, checking for missing values, duplicates, and generating summary statistics.

● **Feature Engineering**: Transformation of date columns and creation of new features like actual price and actual discount.

● **Visualization in Pandas**: Distribution analysis, relationships between variables, and time-based trends.

**Project Outcome & Insights**

The project performs Exploratory Data Analysis (EDA) on retail store inventory dataset to gain meaningful insights into sales performance, ordered units and competitor pricing. Below are the key outcomes:

**1. Sales Performance**

● **Product category Wise Top Sales**: The project groups sales based on different products categories to identify the most profitable segments.

● **Time Series Analysis**: It shows sales trends over time, helping businesses identify seasonal fluctuations and peak sales periods.

● **Top Performing Categories**: Identifies the product categories with the highest sales and revenue.

**2. Customer Behaviour Analysis**

● **Top 3 High-units sold** : Helps businesses recognize their most valuable category and plan targeted marketing strategies.

**3.Business Growth**

● **Day-over-Day Sales Growth**: Tracks annual sales growth percentages, enabling better financial planning.

**Feature Engineering:**

Created new columns such as:

● **actual\_discount**(units sold/discount).

● **actual\_pricing**(competitor pricing and price).

● **order\_weekday** (Extracted from date).

**Key Questions and Insights to be Addressed:**

Q.1) Which product category have the highest discount.

category\_sales=df.groupby('Category')['Discount'].sum().sort\_values(ascending=False)  
print("category\_sales”, category\_sales)

Category Category\_sales

Toys 15

Electronics 10

Furniture 10

Groceries 0

Q.2) Which product category have the highest units sold.

units\_sales=df.groupby('Category')['Units Sold'].sum().sort\_values(ascending=False)  
print("units\_sales",units\_sales)

Category Units sold

Electronics 982

Furniture 499

Toys 495

Groceries 493

Q.3) How much actual discount on the units sold.

df['actual\_discount']=(df['Units Sold']/df['Discount'])  
print(df['actual\_discount'])

19935 49.2

27607 inf

36571 inf

39256 33.0

53157 49.9

Q.4) How much actual pricing on the units sold.

df['actual\_pricing']=(df['Competitor Pricing']-df['Price'])  
print(df['actual\_pricing'])

19935 -3.55

27607 0.91

36571 4.73

39256 -2.66

53157 -3.95

Q.5) On which day the highest units are ordered.

df['order\_weekday']=df['Date'].dt.day\_name()  
print(df['order\_weekday'])

19935 Tuesday

27607 Tuesday

36571 Sunday

39256 Saturday

53157 Friday

Q.6) Which product category have the highest units sold.

units\_sales=df.groupby('Category')['Units Sold'].sum().sort\_values(ascending=False)  
print("units\_sales",units\_sales)

units\_sales Category

Electronics 982

Furniture 499

Toys 495

Groceries 493

Q.7) Which of the top 3 product id's are sold.

top\_products=df.groupby('Product ID')['Units Sold'].sum().sort\_values(ascending=False).head(3)  
print(top\_products)

Product ID Units sold

P0018 499

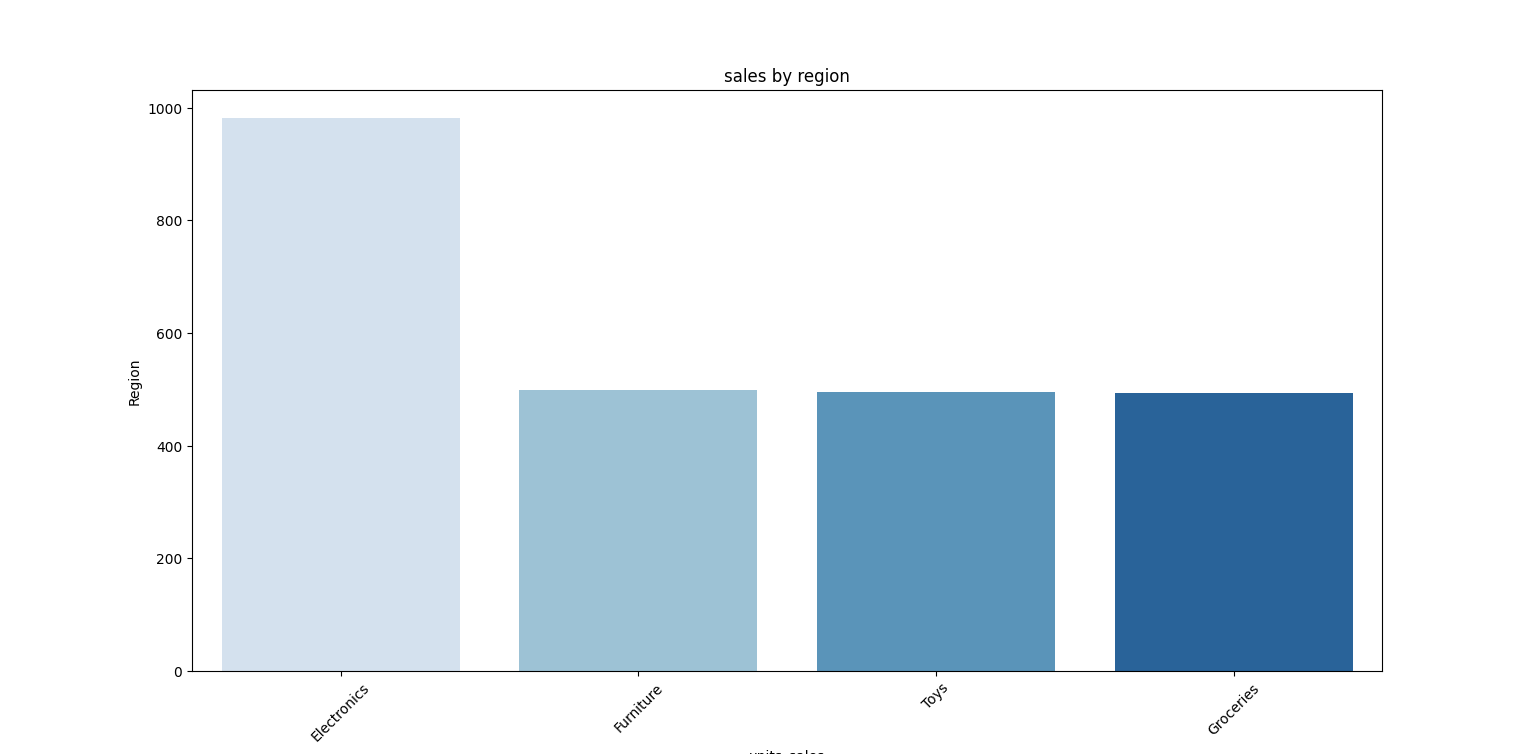
P0017 495

P0012 493

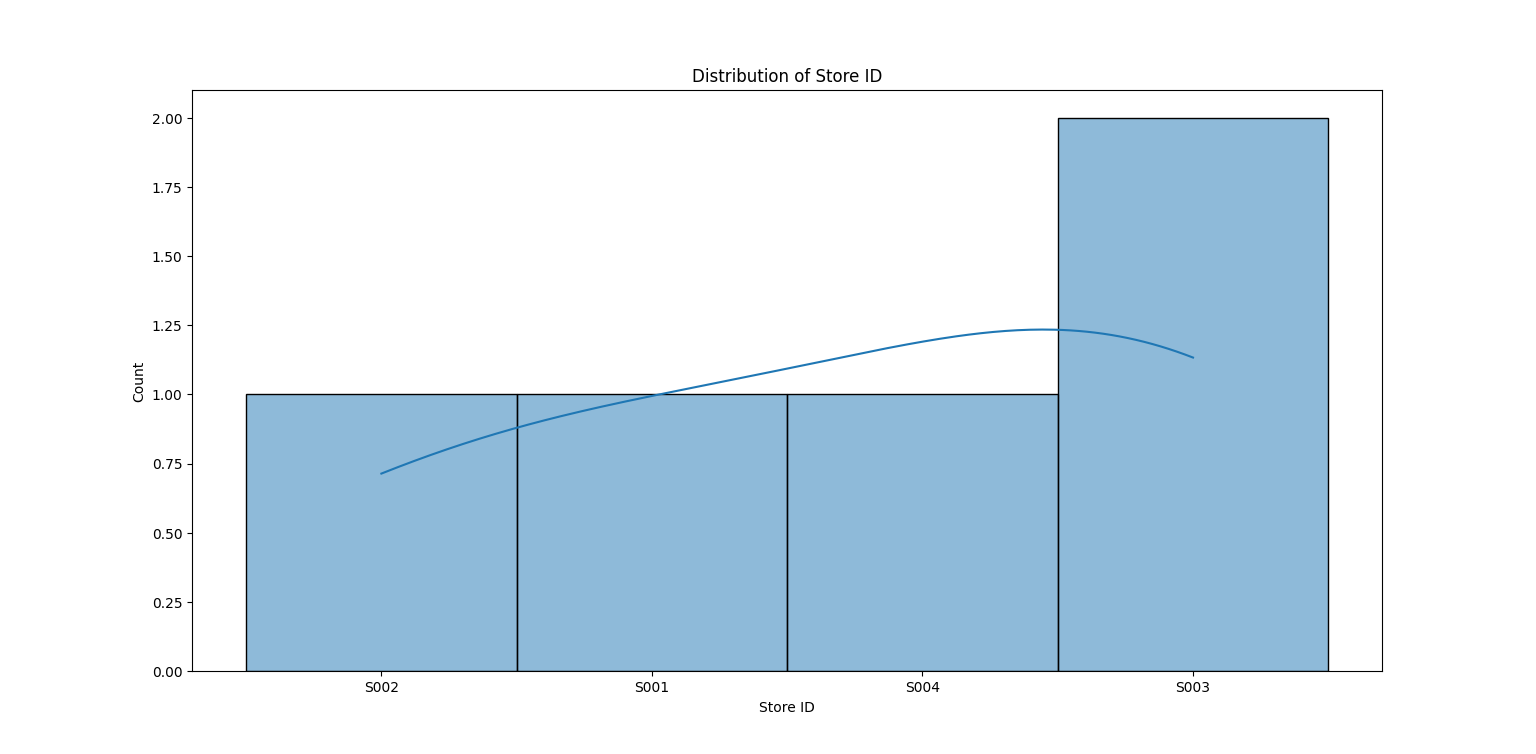
**Visualization:**

Several charts created to present inside including:

● Sales by region (Bar chart)



● Distribution of Store ID (Bar chart)



● Sales trends over time (Line chart)

