**KINGSTON ENGINEERING COLLEGE - 5113**

**DATA ANALYTICS WITH COGNOS**

**TITLE : PRODUCT SALES ANALYSIS**

**TEAM MEMBER:**

PON LAVANYA.S **-** 511321205021

POORNIMA.S **-** 511321205022

KAVIYA.J **-** 511321205014

SRINIDHI.S **-** 511321205039

**ABOUT:**

**In today's competitive business environment, understanding the dynamics of product sales is crucial for any organization seeking to thrive. Product sales analysis is a fundamental component of market research and strategic planning, enabling businesses to comprehensively assess the performance of their products in the market. By evaluating sales data and related metrics, companies can make informed decisions about pricing, marketing strategies, inventory management, and product development, thus optimizing their overall profitability and market share.**

**CONNECT TO IBM COGNOS:**

import ibm\_db

# Replace these values with your own IBM Cognos credentials and database details

dsn\_driver = "IBM DB2 ODBC DRIVER"

dsn\_database = "YOUR\_DATABASE\_NAME"

dsn\_hostname = "YOUR\_HOSTNAME"

dsn\_port = "YOUR\_PORT"

dsn\_protocol = "TCPIP"

dsn\_uid = "YOUR\_USERNAME"

dsn\_pwd = "YOUR\_PASSWORD"

# Establish the database connection

dsn = (

"DRIVER={};"

"DATABASE={};"

"HOSTNAME={};"

"PORT={};"

"PROTOCOL={};"

"UID={};"

"PWD={};").format(dsn\_driver, dsn\_database, dsn\_hostname, dsn\_port, dsn\_protocol, dsn\_uid, dsn\_pwd)

conn = ibm\_db.connect(dsn, "", "")

# Sample query for product sales analysis

query = "SELECT Product, SUM(Sales) AS Total\_Sales FROM Sales\_Data GROUP BY Product"

# Executing the query

stmt = ibm\_db.exec\_immediate(conn, query)

# Fetching and printing the results

result = ibm\_db.fetch\_both(stmt)

while result:

print("Product:", result[0], "Total Sales:", result[1])

result = ibm\_db.fetch\_both(stmt)

# Closing the connection

ibm\_db.close(conn)

**DATASET:**

Q1- Total unit sales of product 1  
Q2- Total unit sales of product 2  
Q3- Total unit sales of product 3  
Q4- Total unit sales of product 4  
S1- Total revenue from product 1  
S2- Total revenue from product 2  
S3- Total revenue from product 3  
S4- Total revenue from product 4

import pandas as pd

# Create the dataset

data = {

'Q1': [100, 150, 200, 120],

'Q2': [80, 130, 180, 90],

'Q3': [120, 160, 210, 110],

'Q4': [90, 140, 190, 100],

'S1': [1000, 1500, 2000, 1200],

'S2': [800, 1300, 1800, 900],

'S3': [1200, 1600, 2100, 1100],

'S4': [900, 1400, 1900, 1000]

}

# Create a DataFrame

df = pd.DataFrame(data, index=['Product 1', 'Product 2', 'Product 3', 'Product 4'])

# Save the DataFrame to a CSV file

df.to\_csv('sales\_data.csv')

**PROGRAM:**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**# Sample sales data (you can load your own data here)**

**data = {**

**'Product': ['Product A', 'Product B', 'Product C', 'Product A', 'Product B', 'Product C'],**

**'Sales': [100, 150, 200, 120, 100, 180],**

**'Date': ['2023-01-01', '2023-01-01', '2023-01-01', '2023-02-01', '2023-02-01', '2023-02-01']**

**}**

**# Creating a DataFrame from the data**

**df = pd.DataFrame(data)**

**# Converting the 'Date' column to datetime**

**df['Date'] = pd.to\_datetime(df['Date'])**

**# Total sales**

**total\_sales = df['Sales'].sum()**

**print(f'Total Sales: {total\_sales}')**

**# Average sales**

**average\_sales = df['Sales'].mean()**

**print(f'Average Sales: {average\_sales}')**

**# Sales by product**

**sales\_by\_product = df.groupby('Product')['Sales'].sum()**

**# Plotting a bar chart for sales by product**

**sales\_by\_product.plot(kind='bar', rot=0)**

**plt.title('Total Sales by Product')**

**plt.xlabel('Product')**

**plt.ylabel('Total Sales')**

**plt.show()**

**# Sales trends over time**

**df.set\_index('Date', inplace=True)**

**monthly\_sales = df.resample('M')['Sales'].sum()**

**# Plotting a line chart for monthly sales trends**

**monthly\_sales.plot(marker='o')**

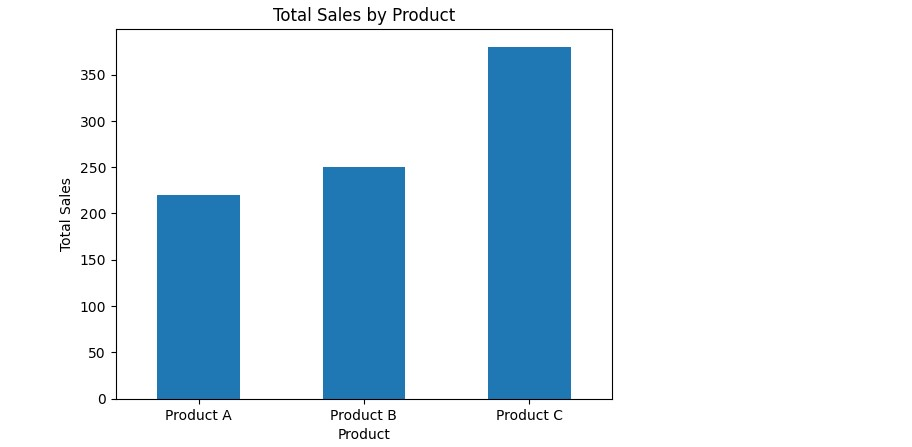
**plt.title('Monthly Sales Trends')**

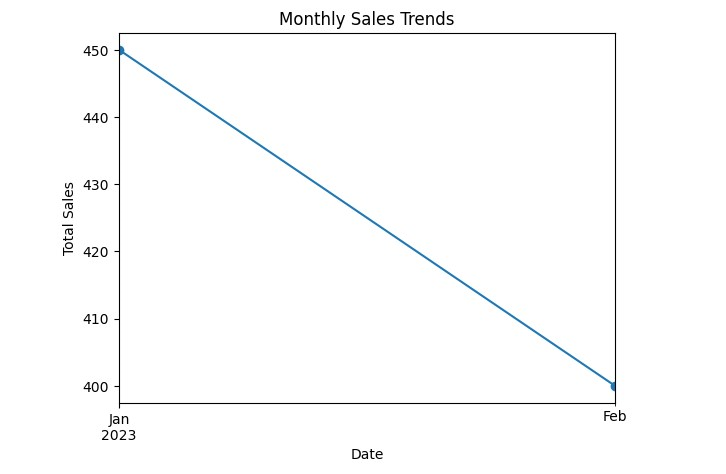
**plt.xlabel('Date')**

**plt.ylabel('Total Sales')**

**plt.show()**

**OUTPUT:**

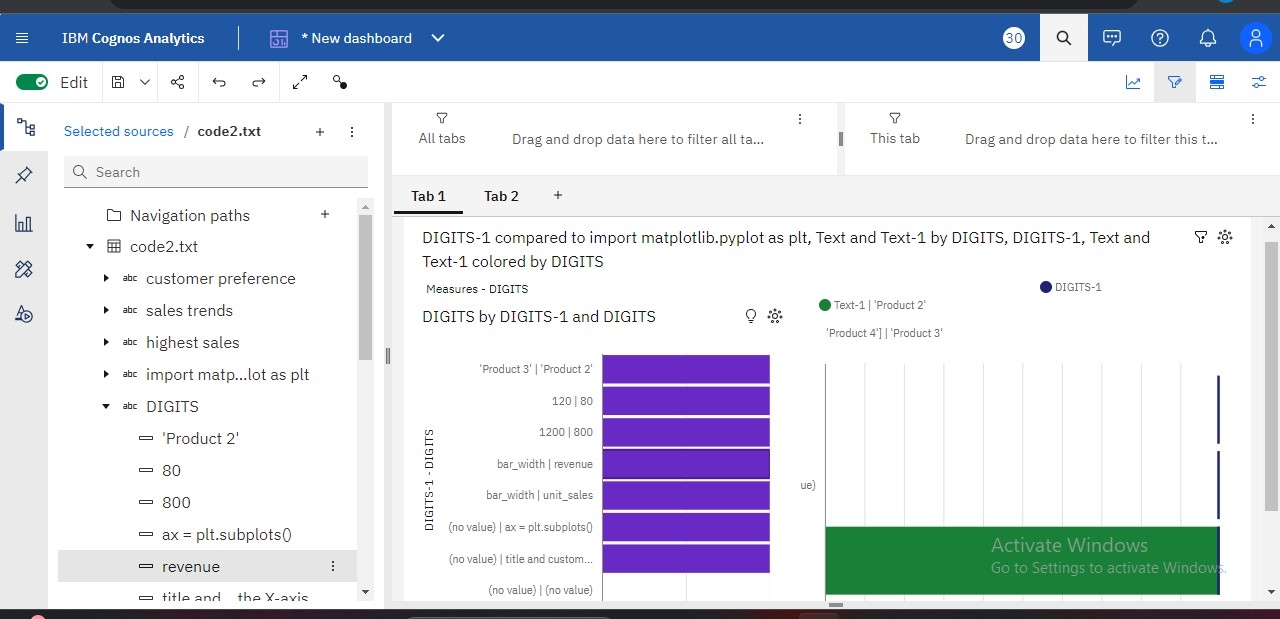




**PROGRAM:**

**import matplotlib.pyplot as plt**  
  
**# Data**  
**products = ['Product 1', 'Product 2', 'Product 3', 'Product 4']**  
**unit\_sales = [100, 80, 120, 90]**  
**revenue = [1000, 800, 1200, 900]**  
  
**# Creating the bar plot**  
**fig, ax = plt.subplots()**  
**bar\_width = 0.35**  
**index = range(len(products))**  
**rects1 = ax.bar(index, unit\_sales, bar\_width, label='Unit Sales', color='b')**  
**rects2 = ax.bar([i + bar\_width for i in index], revenue, bar\_width, label='Revenue', color='r')**  
  
**# Adding labels, title and customizing the X-axis**  
**ax.set\_xlabel('Products')**  
**ax.set\_ylabel('Quantity')**  
**ax.set\_title('Unit Sales and Revenue for Different Products')**  
**ax.set\_xticks([i + bar\_width / 2 for i in index])**  
**ax.set\_xticklabels(products)**  
**ax.legend()**  
  
**# Display the bar plot**  
**plt.show()**

**OUTPUT:**



CONCLUSION:

**- The product sales analysis code is a valuable tool for businesses to analyse their sales data and gain insights into their performance.**  
**- By analysing sales data, businesses can identify trends and patterns, enabling them to make informed decisions and improve their sales strategies.**  
**- The code allows for efficient and accurate analysis of sales data, helping businesses identify their best-selling products, understand customer preferences, and optimize their product offerings.**

**THANK YOU**