KINGSTON ENGINEERING COLLEGE - 5113

DATA ANALYTICS WITH COGNOS

TITLE : PRODUCT SALES ANALYSIS

TEAM MEMBERS:

PON LAVANYA.S - **au**511321205021

POORNIMA.S - **au**511321205022

KAVIYA.J - **au**511321205014

SRINIDHI.S - **au**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

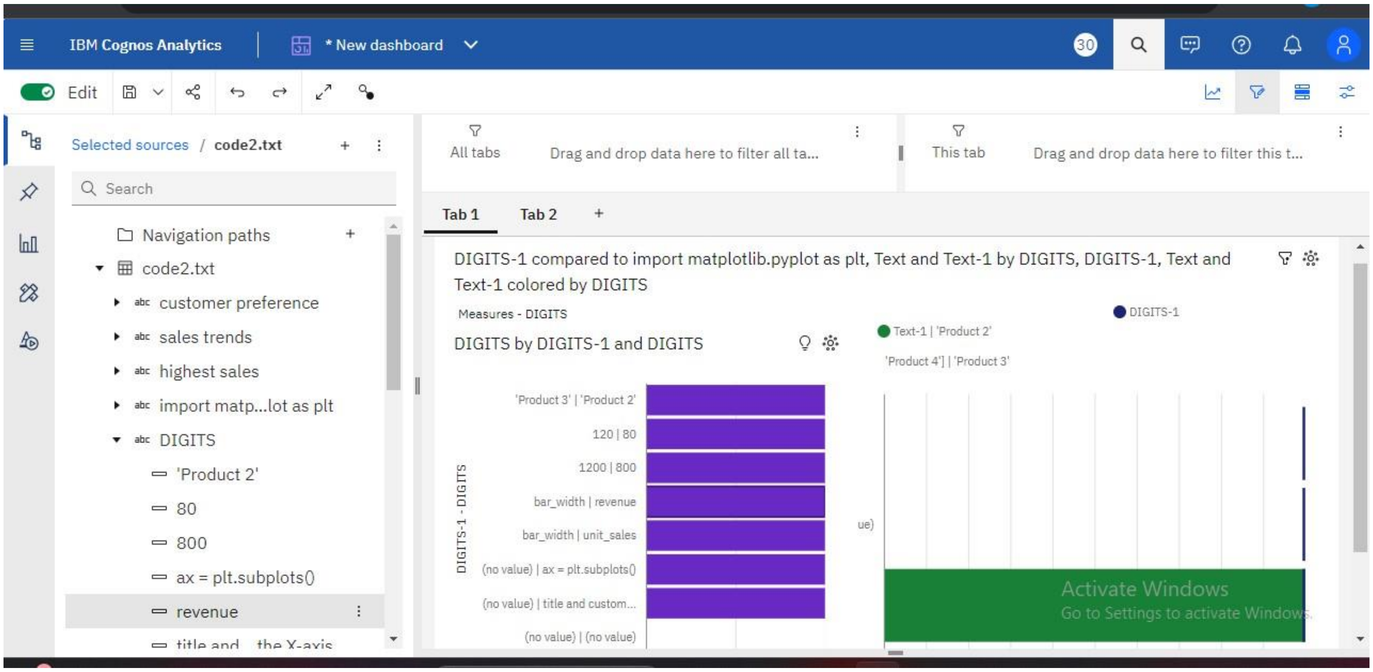
```
import pandas as pd

# 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)
```

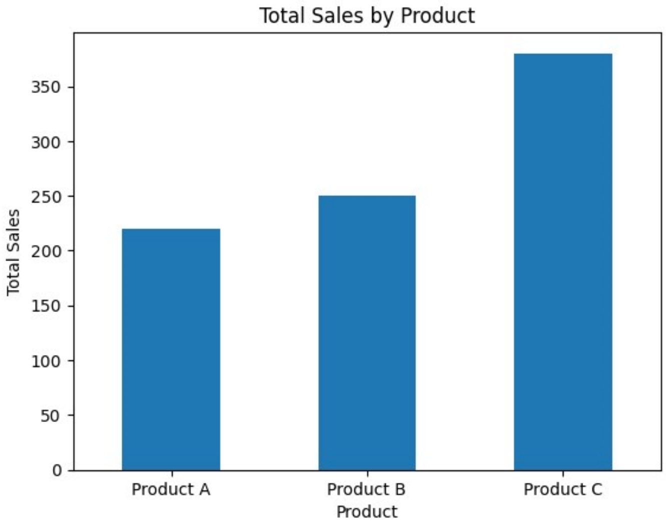
```
plt.title('Total Sales by Product')
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.show()

# 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()
```

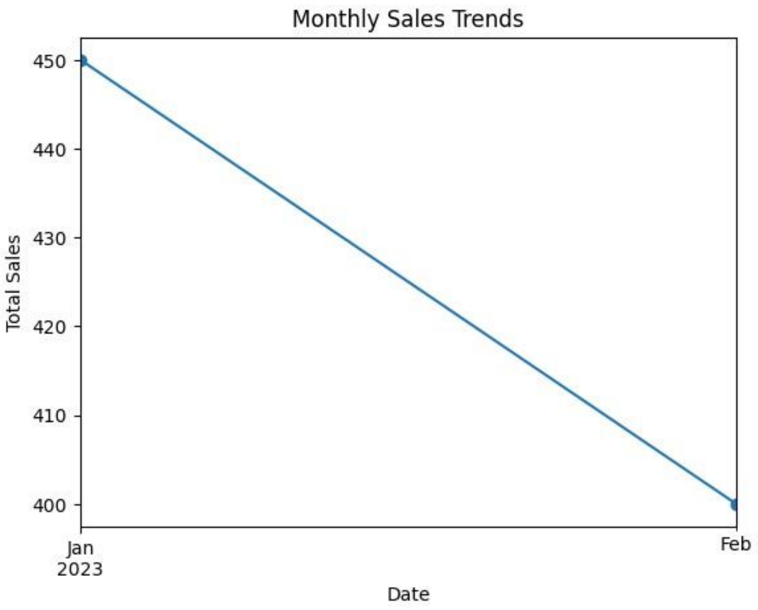
OUTPUT:



DIGITS-1 compared to import matplotlib.pyplot as plt, Text and Text-1 by DIGITS, DIGITS-1, Text and Text-1 colored by DIGITS

DIGITS by DIGITS-1 and DIGITS





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