

# DATA ANALYTICS USING COGNOS – GROUP 3

## Project 5: Product Sales Analysis

### Project Description:

The "Product Sales Analysis" project focuses on harnessing the power of IBM Cognos to analyse sales data, extract valuable insights, and drive informed business decisions. By exploring top-selling products, identifying peak sales periods, and understanding customer preferences, this initiative aims to enhance inventory management and refine marketing strategies. Key components of the project include defining analysis objectives, collecting relevant sales data, designing insightful visualizations within IBM Cognos, and translating findings into actionable insights.

## Phase 3: DEVELOPMENT PART 1

### 1. Data Pre-Processing:

#### Step 1: Library Imports

The code imports the required Python libraries.

- pandas (imported as pd) is a popular library for data manipulation and analysis.
- matplotlib (imported as plt) is a library used for creating data visualizations, such as plots and charts.

#### Step 2: Data Loading

The code reads a dataset from a CSV file named "statsfinal.csv" and loads it into a Pandas DataFrame named 'data'.

#### Step 3: Data Inspection:

The code prints the first few rows of the 'data' DataFrame using the .head() method. By default, .head() displays the first 5 rows of the DataFrame. This allows to quickly inspect and get an initial overview of the dataset's structure and contents.

## CODE:

```
import pandas as pd
import matplotlib.pyplot as plt

# Read the dataset
data = pd.read_csv("statsfinal.csv")

# Print the first few rows of the dataset
print(data.head())
```

```
   Unnamed: 0      Date  Q-P1  Q-P2  Q-P3  Q-P4  S-P1  S-P2 \
0           0  13-06-2010  5422  3725   576   907 17187.74 23616.50
1           1  14-06-2010   7047    779  3578  1574  22338.99  4938.86
2           2  15-06-2010   1572   2082   595  1145   4983.24 13199.88
3           3  16-06-2010   5657   2399  3140  1672  17932.69 15209.66
4           4  17-06-2010   3668   3207  2184   708  11627.56 20332.38

      S-P3      S-P4
0  3121.92  6466.91
1 19392.76 11222.62
2  3224.90   8163.85
3 17018.80 11921.36
4 11837.28  5048.04
```

## Step 4: Check Missing Values

The code checks for missing values within a dataset named 'data' and prints a summary that reveals the number of missing values in each column. This is a crucial step in data pre-processing, helping identify and address missing data to ensure accurate and reliable analysis.

## Step 5: Derive Information

The `data.info()` method in Pandas offers a swift yet comprehensive overview of a dataset. It presents details such as non-null counts, data types, and memory usage for each column. This summary is a handy tool for gaining rapid insights into the dataset's characteristics, facilitating initial data exploration and comprehension.

## CODE :

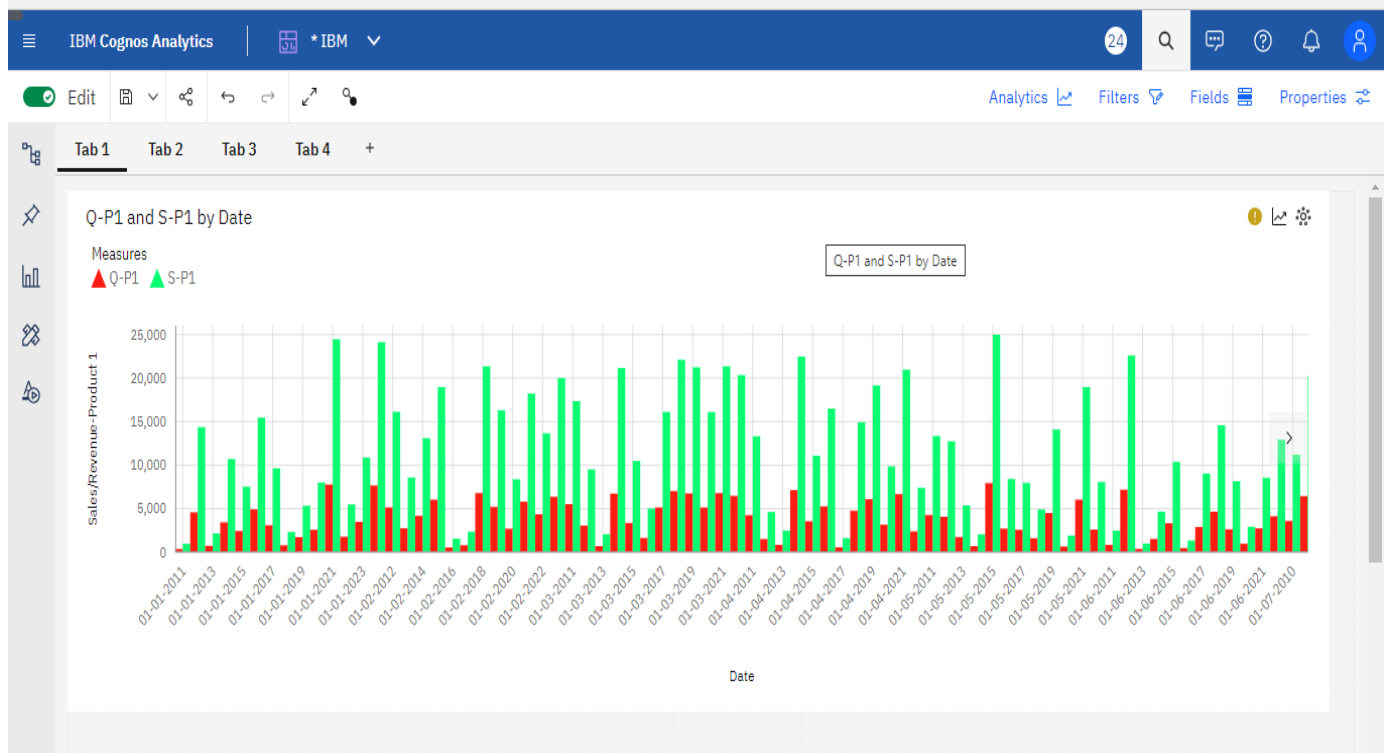
```
# Check for missing values
print(data.isnull().sum())
# Gather Information
data.info()
```

```
Unnamed: 0      0
Date           0
Q-P1           0
Q-P2           0
Q-P3           0
Q-P4           0
S-P1           0
S-P2           0
S-P3           0
S-P4           0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0  4600 non-null     int64
1   Date        4600 non-null     object
2   Q-P1        4600 non-null     int64
3   Q-P2        4600 non-null     int64
4   Q-P3        4600 non-null     int64
5   Q-P4        4600 non-null     int64
6   S-P1        4600 non-null     float64
7   S-P2        4600 non-null     float64
8   S-P3        4600 non-null     float64
9   S-P4        4600 non-null     float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB
```

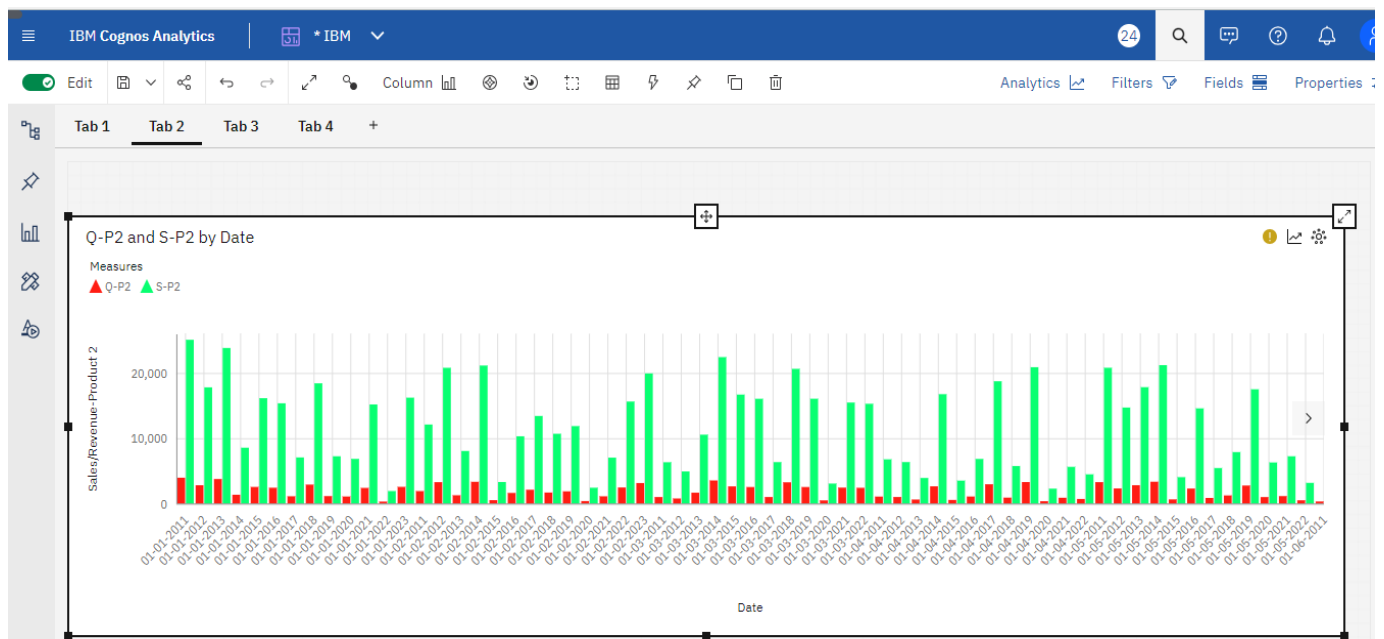
## 2. Data Visualization:

Using IBM Cognos Analytics, we visualize the dataset to get clear insights about the data. Here Bar plot have been used to analyse the Sales over a period of time along with the revenue generated on each date.

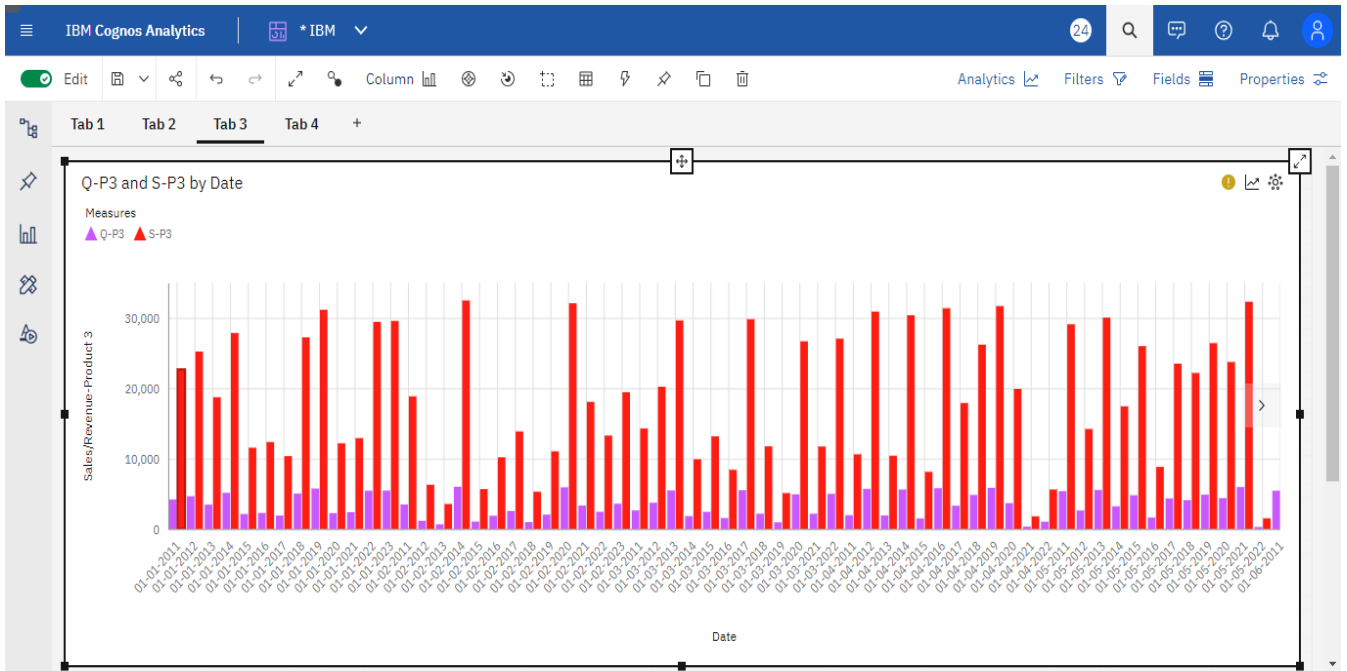
### PRODUCT – 1:



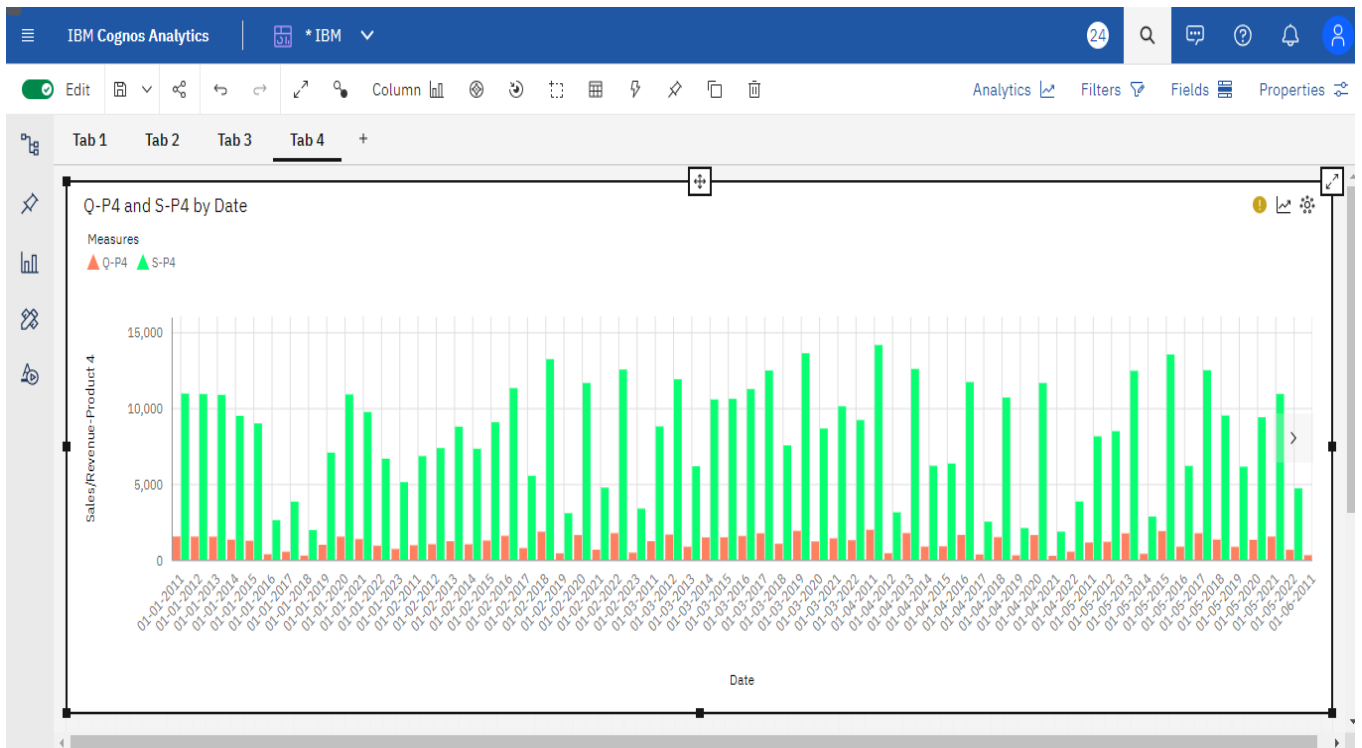
### PRODUCT – 2:



## PRODUCT – 3:



## PRODUCT – 4:



### 3. Sales Correlation Analysis :

This code calculates the correlation between the "Sales" column and all other columns in a dataset, sorts the correlations in descending order, for each product in the dataset and prints the results. It helps analyze the relationship between "Sales" and other variables in the dataset for all the four products.

**CODE :**

**PRODUCT-1:**

```
[6] correlation = data.corr()
    print(correlation["Q-P1"].sort_values(ascending=False))
```

Q-P1	1.000000
S-P1	1.000000
S-P2	0.002422
Q-P2	0.002422
Unnamed: 0	-0.001120
Q-P3	-0.005650
S-P3	-0.005650
S-P4	-0.059365
Q-P4	-0.059365

Name: Q-P1, dtype: float64  
<ipython-input-6-315fbae7fc4>:1: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False.  
correlation = data.corr()

**PRODUCT-2:**

```
correlation = data.corr()
print(correlation["Q-P2"].sort_values(ascending=False))
```

S-P2	1.000000
Q-P2	1.000000
S-P4	0.013082
Q-P4	0.013082
Unnamed: 0	0.008716
S-P3	0.003729
Q-P3	0.003729
Q-P1	0.002422
S-P1	0.002422

Name: Q-P2, dtype: float64  
<ipython-input-7-6aa1a9be9741>:1: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False.  
correlation = data.corr()

### PRODUCT-3:

```
correlation = data.corr()  
print(correlation["Q-P3"].sort_values(ascending=False))
```

```
S-P3      1.000000  
Q-P3      1.000000  
Unnamed: 0    0.005255  
S-P2      0.003729  
Q-P2      0.003729  
Q-P1     -0.005650  
S-P1     -0.005650  
Q-P4     -0.006693  
S-P4     -0.006693  
Name: Q-P3, dtype: float64  
<ipython-input-8-267e2c327f73>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False.  
correlation = data.corr()
```

### PRODUCT-4:

```
[9] correlation = data.corr()  
print(correlation["Q-P4"].sort_values(ascending=False))
```

```
S-P4      1.000000  
Q-P4      1.000000  
S-P2      0.013082  
Q-P2      0.013082  
Q-P3     -0.006693  
S-P3     -0.006693  
Unnamed: 0 -0.009739  
Q-P1     -0.059365  
S-P1     -0.059365  
Name: Q-P4, dtype: float64  
<ipython-input-9-99240dcf8482>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False.  
correlation = data.corr()
```

### CONCLUSION:

In this document building the project by loading and pre-processing the dataset have been implemented. Different analysis and visualization have been performed and implemented using IBM Cognos.