

PRODUCT SALES ANALYSIS

V.Ranjani

REG NO:420721104035

CK COLLEGE OF ENGINEERING AND TECHNOLOGY

INTRODUCTION

- Product sales analysis is a critical component of business intelligence and strategy. It involves the examination of sales data to gain insights into product performance, customer behavior, and market trends.
- Analysing product sales is crucial for any business to make informed decisions. IBM Cognos Visualization provides a powerful platform for dissecting sales data and uncovering valuable insights.
- In this report, we will explore the capabilities of IBM Cognos Visualization to visualize, analyse, and interpret product sales data.
- By leveraging this tool, businesses can gain a deeper understanding of their sales performance, identify trends, and make data-driven decisions to improve profitability and overall success.
- This analysis aims to showcase the potential of IBM Cognos Visualization in enhancing product sales strategies and driving business growth.

GIVEN DATA SET

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	13-06-2010	5422	3725	576	907	17187.74	23616.5	3121.92	6466.91
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.9	8163.85
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.8	11921.36
4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04
5	18-06-2010	2898	2539	311	1513	9186.66	16097.26	1685.62	10787.69
6	19-06-2010	6912	1470	1576	1608	21911.04	9319.8	8541.92	11465.04
7	20-06-2010	5209	2550	3415	842	16512.53	16167	18509.3	6003.46
8	21-06-2010	6322	852	3646	1377	20040.74	5401.68	19761.32	9818.01
9	22-06-2010	6865	414	3902	562	21762.05	2624.76	21148.84	4007.06

4580	15-01-2023	6425	2862	1557	600	20367.25	18145.08	8438.94	4278
4581	16-01-2023	5962	2794	5631	1553	18899.54	17713.96	30520.02	11072.89
4582	17-01-2023	4990	2233	893	1698	15818.3	14157.22	4840.06	12106.74
4583	18-01-2023	266	2482	507	1376	843.22	15735.88	2747.94	9810.88
4584	19-01-2023	2792	2621	5676	427	8850.64	16617.14	30763.92	3044.51
4585	20-01-2023	4987	1177	3145	1112	15808.79	7462.18	17045.9	7928.56
4586	21-01-2023	6896	2799	5724	1987	21860.32	17745.66	31024.08	14167.31
4587	22-01-2023	1238	480	4003	537	3924.46	3043.2	21696.26	3828.81
4588	23-01-2023	7681	3243	3529	1128	24348.77	20560.62	19127.18	8042.64
4589	24-01-2023	6290	3084	5892	1751	19939.3	19552.56	31934.64	12484.63
4590	25-01-2023	6160	3967	3285	544	19527.2	25150.78	17804.7	3878.72
4591	26-01-2023	3225	3809	1964	1851	10223.25	24149.06	10644.88	13197.63
4592	27-01-2023	962	813	3849	1987	3049.54	5154.42	20861.58	14167.31
4593	28-01-2023	4938	3404	3957	1115	15653.46	21581.36	21446.94	7949.95
4594	29-01-2023	1227	3044	5510	1896	3889.59	19298.96	29864.2	13518.48
4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.5	9689.67
4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.5	9347.43
4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62
4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21
4599	03-02-2023	1234	3854	2321	406	3911.78	24434.36	12579.82	2894.78

NECESSARY STEP TO FOLLOW

IBM Cognos Visualization primarily relies on its native features and integration capabilities for product sales analysis. However, to enhance your product sales analysis using IBM Cognos Visualization, you may want to consider integrating it with other tools and libraries for data preparation and data source connectivity.

- **IBM Cognos Analytics:**

This is the primary platform for creating reports and dashboards using IBM Cognos Visualization. It provides a wide range of data connectors and data modelling capabilities.

- **IBM Data Warehouse or DataMart:**

Ensure that your sales data is well-organized in a data warehouse or Datamart, making it easier to access and analyse in IBM Cognos.

- **ETL (Extract, Transform, Load) Tools:**

Consider using ETL tools like IBM DataStage or open-source alternatives like Talend or Apache Nifi to extract, clean, and transform data from various sources into a format compatible with IBM Cognos.

- **Exploratory data analysis:**

Exploratory data analysis (EDA) is a critical step in understanding your product sales data before using IBM Cognos for visualization. Here's a general process:

- **Data Collection:**

Gather product sales data from various sources and ensure it's in a structured format.

- **Data Cleaning:**

Remove duplicates, handle missing values, and standardize data types prepare for analysis.

- **Total unit sales Product 1, Product 2, Product 3, Product 4**

```
q = df[["Q-P1","Q-P2","Q-P3","Q-P4"]].sum()
```

```
print(q)
```

```
plt.figure(figsize=(8,8))
```

```
plt.pie(q,labels=df[["Q-P1","Q-P2","Q-P3","Q-P4"]].sum().index,shadow=True,autopct="%0.01f%%",textprops={"fontsize":20},wedgeprops={'width': 0.8},explode=[0,0,0,0.3])
```

```
plt.legend(loc='center right', bbox_to_anchor=(1.2, 0.8));
```

Extract year from the 'Day' 'Month' 'year' from the 'Date' column using a lambda function. We need to get the year from the data to analyse sales year to year

```
data['Day'] = data['Date'].apply(lambda x: x.split('-')[0])
```

```
data['Month'] = data['Date'].apply(lambda x: x.split('-')[1])
```

```
data['Year'] = data['Date'].apply(lambda x: x.split('-')[2])
```

- **SQL and Database Libraries:**

If you're dealing with large datasets, you may need to leverage SQL and database libraries to optimize your queries. IBM Db2 or open-source databases like PostgreSQL are commonly used.

- **Python or R for Data Analysis:**

For advanced analytics, you can use Python with libraries like Pandas, NumPy, or R with libraries like ggplot2 to preprocess and analyse data before visualizing it in IBM Cognos.

- **Data Visualization Libraries:**

While IBM Cognos Visualization is robust, you can complement it with other data visualization libraries like D3.js or Plotly for creating custom visualizations or interactive charts.

IMPORTANCE OF LOADING AND PREPROCESSING

Loading and preprocessing are crucial steps in product sales analysis using IBM Cognos Visualization for several reasons:

- **Data Integrity:**

Loading and preprocessing ensure that the data used for analysis is accurate and complete. Any inconsistencies or errors in the data can lead to incorrect insights and decisions.

- **Performance Optimization:**

Proper preprocessing can help optimize data for faster querying and reporting. This is particularly important for large datasets, as it reduces the time required for analysis.

- **Data Transformation:**

Preprocessing allows you to transform raw data into a format that is suitable for analysis. This includes cleaning, aggregating, and structuring the data in a way that aligns with your specific analysis goals.

- **Data Integration:**

Loading and preprocessing enable you to integrate data from multiple sources, such as sales data, customer data, and product data. This integration can provide a more holistic view of your business, helping you identify correlations and trends.

- **Data Security:**

Proper loading and preprocessing can help ensure that sensitive information is handled securely, complying with data privacy regulations and company policies.

- **User Experience:**

Well-pre-processed data results in a smoother and more user-friendly experience for analysts and decision-makers using IBM Cognos Visualization. They can create dashboards and reports with confidence in the data's accuracy.

In summary, loading and preprocessing are fundamental steps in product sales analysis using IBM Cognos Visualization. They set the stage for accurate, efficient, and insightful data analysis, which, in turn, supports data-driven decision-making within an organization.

CHALLENGES INVOLVED IN LOADING AND PREPROCESSING A PRODUCT SALES ANALYSIS

Overcoming challenges in loading and preprocessing for product sales analysis in IBM Cognos Visualization requires careful planning and execution. Here are some strategies to address common challenges:

- **Data quality assurance:**

Invest in data quality tools and processes to identify and rectify errors in the source data. Establish data governance policies to ensure data accuracy and consistency.

- **Data Integration:**

Use ETL (Extract, Transform, Load) tools to consolidate and integrate data from multiple sources. Ensure proper mapping and transformation of data to match the required format.

HOW TO OVERCOME THE CHALLENGES OF LOADING AND PREPROCESSING A PRODUCT SALES ANALYSIS

- **Monitoring and Error Handling:**

Set up monitoring systems to detect data loading and preprocessing failures. Establish error-handling processes to address issues promptly.

- **Training and Skill Development:**

Invest in training for data professionals to ensure they are proficient in IBM Cognos and data processing techniques. Stay updated on best practices and new features in Cognos.

- **Performance Tuning:**

Continuously monitor and analyse performance metrics to identify bottlenecks and make necessary adjustments to the loading and preprocessing workflows.

LOADING A DATASET IN IBM COGNOS FOR PRODUCT SALES ANALYSIS INVOLVES SEVERAL STEPS. HERE'S A HIGH-LEVEL OVERVIEW OF THE PROCESS

- **Data Source Connection:**

Open IBM Cognos. Connect to your data source, which can be a database, spreadsheet, or other data storage. You can use data modules, data servers, or direct connections, depending on your source.

- **Import Data:**

Once connected, select the option to import your dataset. Choose the dataset that contains your product sales data. This can be a table from a database or a file like Excel.

- **Data Transformation:**

Depending on your dataset, you may need to transform and clean the data. This might include data type conversions, filtering, and handling missing values.

- **Create Data Source:**

After importing and transforming the data, create a data source in Cognos. This data source will serve as the foundation for your analysis.

- **Data Modelling:**

Define relationships between the tables, create calculations, and customize your data source to suit your analysis requirements.

Program:

```
<chart>

  <options>

    <chartSize autoScaling="yes"/>

  </options>

  <type>Bar</type>

  <data>

    <crosstab>

      <data>

        <value name="Product" expr="[Product].[Product Name]"/>

        <value name="Sales" expr="total([Sales])"/>

      </data>

    </crosstab>

  </data>

</chart>

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

Input data files are available in the read-only "../input/" directory

import os

for dirname, _, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))
```


LOADING DATASET

Program:

```
df=pd.read_csv("/kaggle/input/product-sales-data/statsfinal.csv")  
  
df.head(5)
```

Output:

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	13-06-2010	5422	3725	576	907	17187.74	23616.5	3121.92	6466.91
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.9	8163.85
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.8	11921.36
4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04
5	18-06-2010	2898	2539	311	1513	9186.66	16097.26	1685.62	10787.69
6	19-06-2010	6912	1470	1576	1608	21911.04	9319.8	8541.92	11465.04
7	20-06-2010	5209	2550	3415	842	16512.53	16167	18509.3	6003.46
8	21-06-2010	6322	852	3646	1377	20040.74	5401.68	19761.32	9818.01
9	22-06-2010	6865	414	3902	562	21762.05	2624.76	21148.84	4007.06

4580	15-01-2023	6425	2862	1557	600	20367.25	18145.08	8438.94	4278
4581	16-01-2023	5962	2794	5631	1553	18899.54	17713.96	30520.02	11072.89
4582	17-01-2023	4990	2233	893	1698	15818.3	14157.22	4840.06	12106.74
4583	18-01-2023	266	2482	507	1376	843.22	15735.88	2747.94	9810.88
4584	19-01-2023	2792	2621	5676	427	8850.64	16617.14	30763.92	3044.51
4585	20-01-2023	4987	1177	3145	1112	15808.79	7462.18	17045.9	7928.56
4586	21-01-2023	6896	2799	5724	1987	21860.32	17745.66	31024.08	14167.31
4587	22-01-2023	1238	480	4003	537	3924.46	3043.2	21696.26	3828.81
4588	23-01-2023	7681	3243	3529	1128	24348.77	20560.62	19127.18	8042.64
4589	24-01-2023	6290	3084	5892	1751	19939.3	19552.56	31934.64	12484.63
4590	25-01-2023	6160	3967	3285	544	19527.2	25150.78	17804.7	3878.72
4591	26-01-2023	3225	3809	1964	1851	10223.25	24149.06	10644.88	13197.63
4592	27-01-2023	962	813	3849	1987	3049.54	5154.42	20861.58	14167.31
4593	28-01-2023	4938	3404	3957	1115	15653.46	21581.36	21446.94	7949.95
4594	29-01-2023	1227	3044	5510	1896	3889.59	19298.96	29864.2	13518.48
4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.5	9689.67
4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.5	9347.43
4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62
4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21
4599	03-02-2023	1234	3854	2321	406	3911.78	24434.36	12579.82	2894.78

PREPROCESSING THE DATASET

Preprocessing a dataset for product sales analysis in IBM Cognos Visualization involves several steps. Here's a general outline of the process:

- **Data Collection:**

Gather your sales data from various sources, such as databases, spreadsheets, or APIs.

- **Data Cleaning:**

Clean the data to remove any inconsistencies, missing values, or outliers. This ensures that your analysis is based on high-quality data.

- **Data Transformation:**

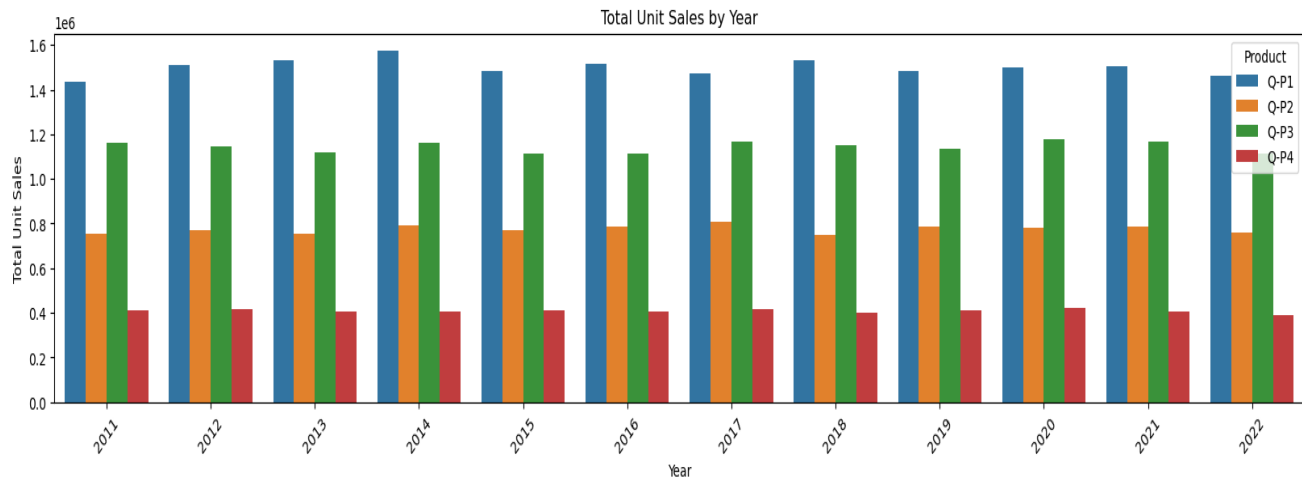
Depending on your analysis goals, you might need to transform the data. For example, you can aggregate daily sales data into monthly or yearly totals.

VISUALIZATION AND PREPROCESSING OF DATA

Program:

```
def plot_bar_chart(df, columns, stri, str1, val):  
    if val == 'sum':  
        sales_by_year = df.groupby('Year')[columns].sum().reset_index()  
    elif val == 'mean':  
        sales_by_year = df.groupby('Year')[columns].mean().reset_index()  
  
    sales_by_year_melted = pd.melt(sales_by_year, id_vars='Year', value_vars=columns, va  
r_name='Product', value_name='Sales')  
  
    plt.figure(figsize=(20,4))  
    ns.barplot(data=sales_by_year_melted, x='Year', y='Sales', hue='Product')
```

Output:

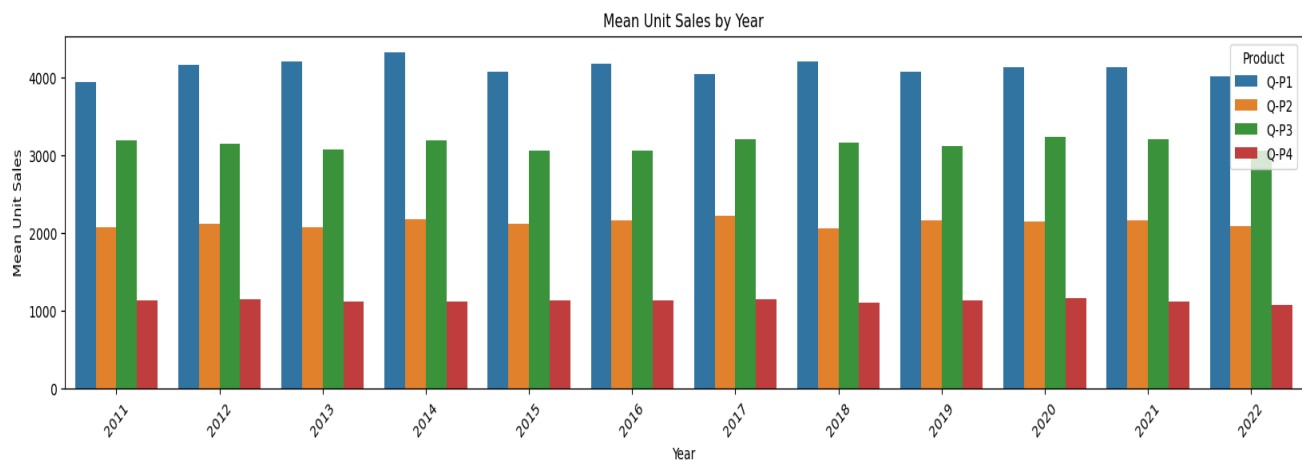


Program:

```
plt.figure(figsize=(12, 8), style='cividis')
plt.xlabel('Year')
plt.ylabel('Total Unit Sales')
plt.title('Total Unit Sales by Year')
plt.xticks(rotation=45)
plt.show()

plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Total Unit Sales', 'Year', 'sum')
plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Mean Unit Sales', 'Year', 'mean')
```

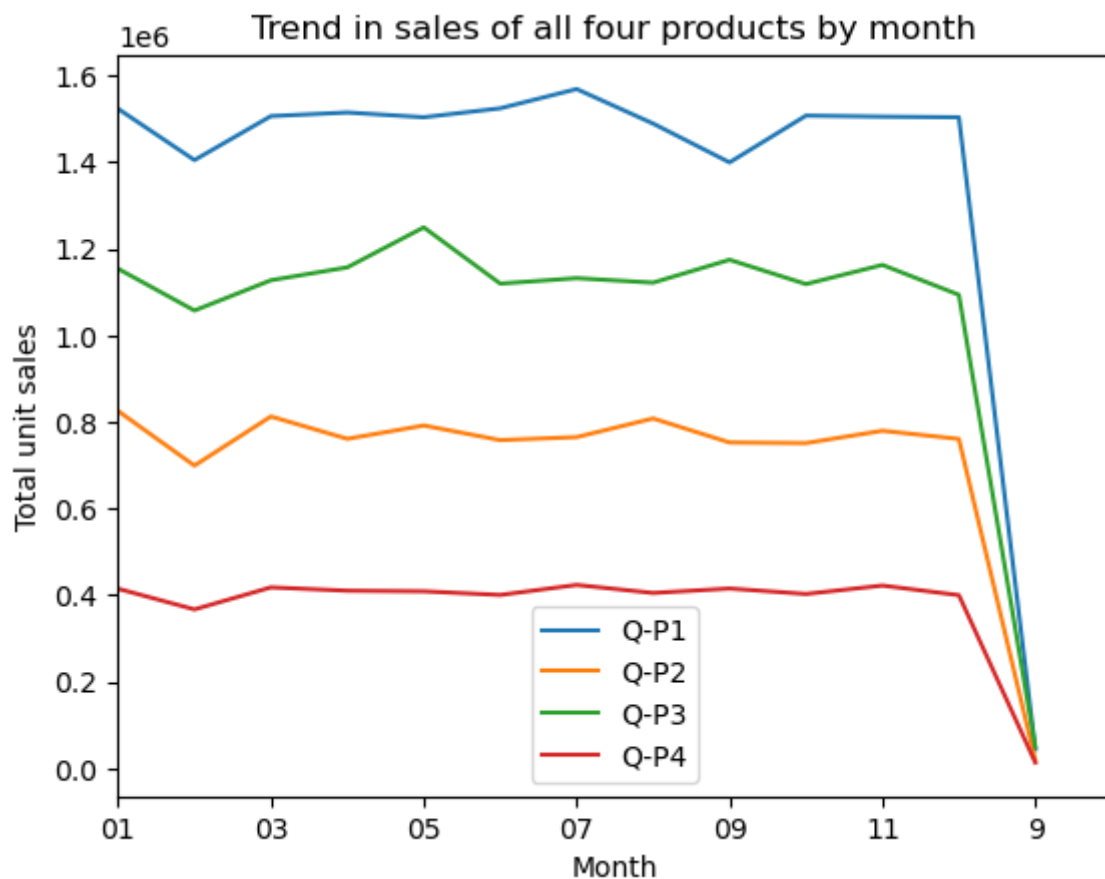
Output:



Program:

```
def month_plot():  
    fig, ax = plt.subplots()  
    data_reduced.groupby('Month')[['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']].sum().plot(ax=ax)  
    ax.set_xlim(left=0, right=13)  
    ax.set_xlabel('Month')  
    ax.set_ylabel('Total unit sales')  
    ax.set_title('Trend in sales of all four products by month')  
    plt.show()  
month_plot()
```

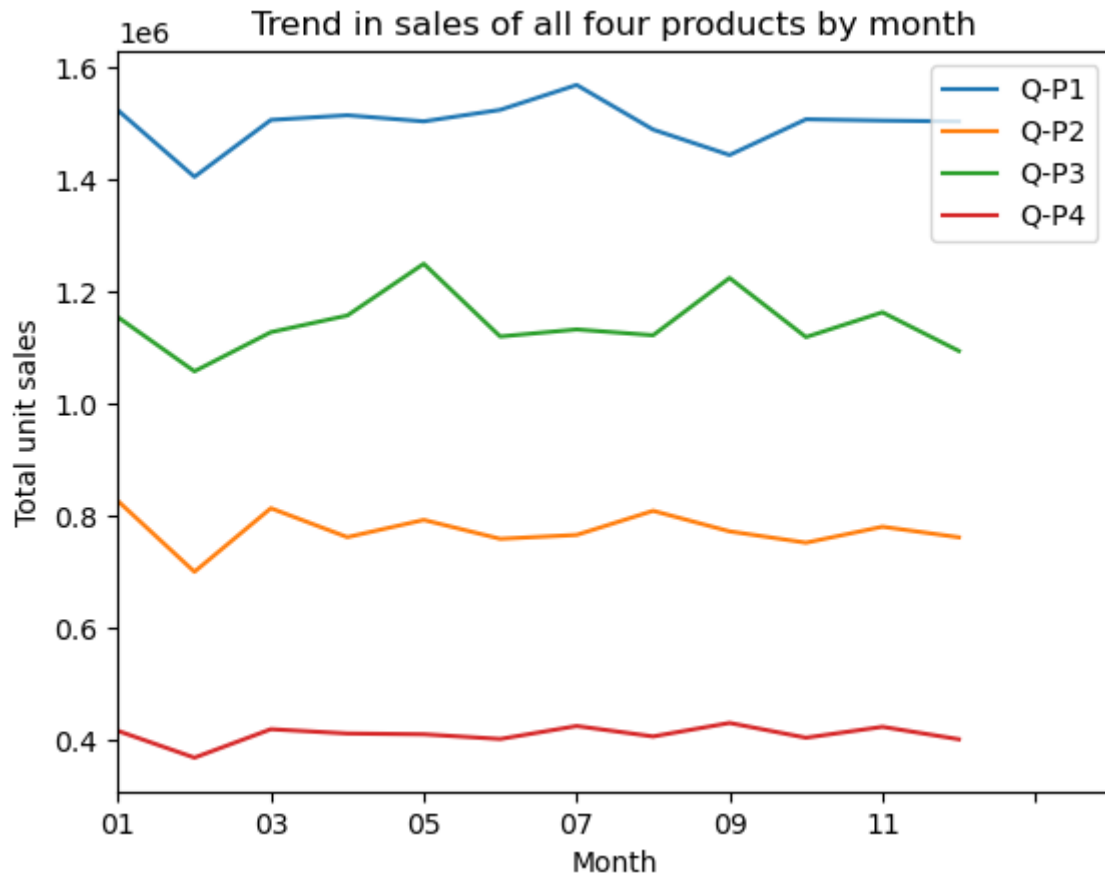
Output:



Program:

```
data_reduced['Month'] = data['Month'].replace('9', '09')  
month_plot()
```

Output:

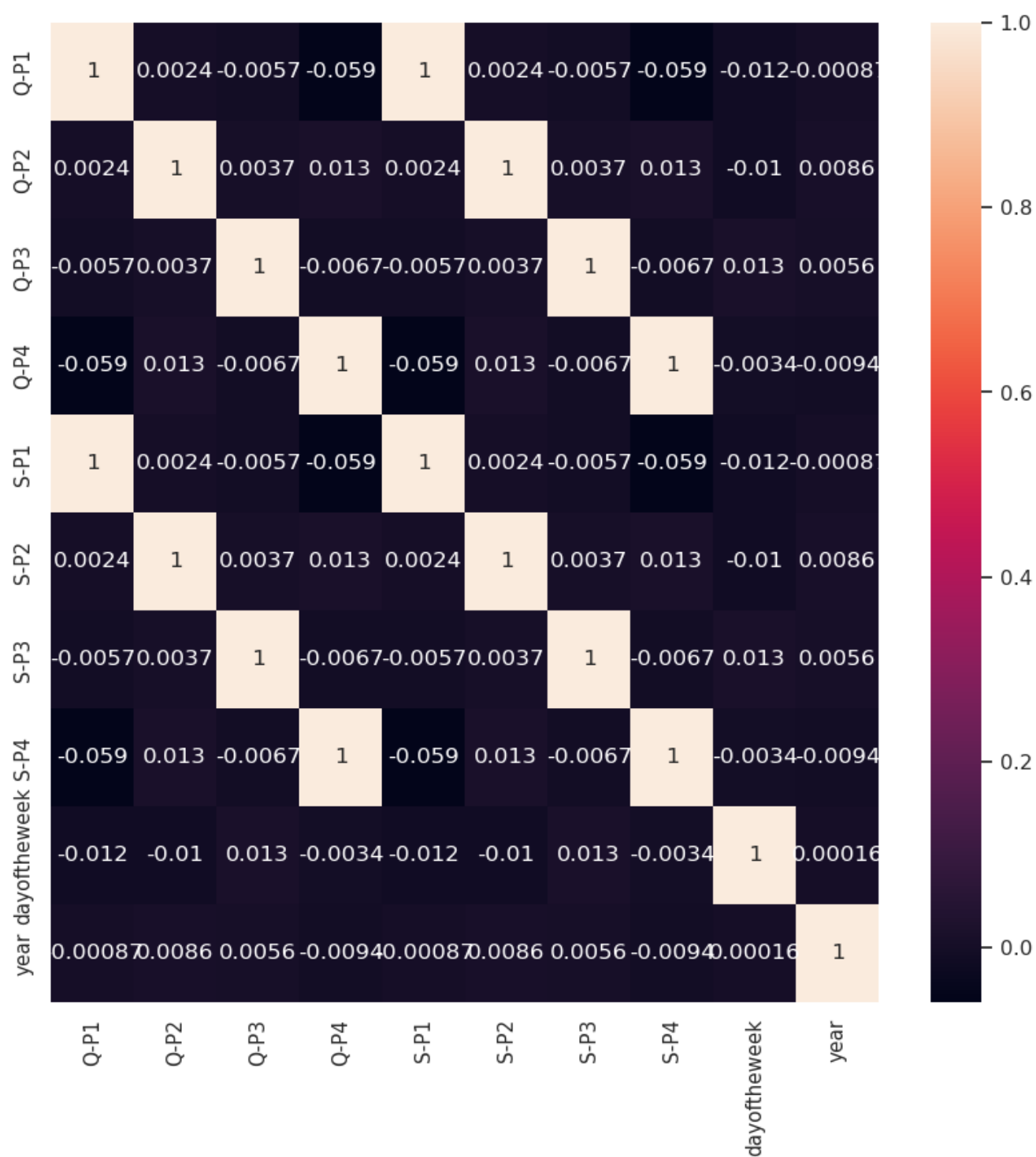


VISUALIZATION CORRELATION

Program:

```
plt.figure(figsize=(10,10))  
sns.heatmap(df.corr(),annot=True)
```

Output:



Program:

```
def avg_on_31st(df, product):  
    df_31 = df[df['Day'] == '31']  
    avg_sales = df_31[product].mean()  
    return avg_sales  
  
avg_on_31st(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']).round(2)  
  
avg_on_31st(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4']).round(2)
```

Output:

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Day	Month	Year
0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91	13	06	2010
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62	14	06	2010
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85	15	06	2010
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36	16	06	2010
4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04	17	06	2010
...
4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.50	9689.67	30	01	2023
4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.50	9347.43	31	01	2023
4597	01-02-2023	6289	3143	3588	474	19936.13	19926.62	19446.96	3379.62	01	02	2023
4598	02-02-2023	3122	1188	5899	517	9896.74	7531.92	31972.58	3686.21	02	02	2023
4599	03-02-2023	1234	3854	2321	406	3911.78	24434.36	12579.82	2894.78	03	02	2023

AVERAGE FOR UNIT SALES

Output:

```
Q-P1  3813.74
Q-P2  2058.80
Q-P3  3183.88
Q-P4  1098.61
dtype: float64
```

AVERAGE FOR REVENUE

Output:

```
S-P1  12089.55
S-P2  13052.78
S-P3  17256.63
S-P4   7833.07
dtype: float64
```

CONCLUSION

Analyzing product sales data using IBM Cognos visualization has provided valuable insights into our business performance. The visualizations have enabled us to make data-driven decisions and identify key trends and opportunities. In conclusion, this tool has been instrumental in improving our sales strategies, optimizing inventory management, and enhancing overall business efficiency. We will continue to leverage IBM Cognos visualization to drive growth and success in our organization. A product sales analysis can provide a comprehensive view of a product's performance in the market, helping you make data-driven decisions to optimize sales, marketing, and product development strategies. It's important to regularly conduct such analyses to adapt to changing market conditions and customer preferences.