PRODUCT SALES ANALYSIS

M.ANNAPOORANI

420721104003

CK COLLEGE OF ENGINEERING AND TECHNOLOGY

ABSTRACT

The primary aim of this project is to harness the power of IBM Cognos for the analysis of sales data to derive critical insights into various facets of our business, including top-selling products, sales trends, and customer preferences. These insights will empower us to enhance inventory management and optimize our marketing strategies. The project encompasses several key components:

OVERVIEW

Product sales analysis is a critical component of business intelligence and decision-making processes. It involves the systematic examination of historical sales data to extract valuable insights, identify trends, and make data-driven decisions. The primary goal of product sales analysis is to improve overall sales performance, optimize inventory management, and enhance profitability.

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.

OBJECTIVE

In this product sales analysis project, our core objectives are to obtain comprehensive insights into product performance, sales trends, and customer preferences. Our objectives include:

- **Product Performance Assessment:** Analysing product sales data to identify top-selling products and assess their performance in the market.
- **Sales Trend Analysis:** Studying historical sales data to understand trends, seasonality, and fluctuations in demand.
- **Customer Preferences:** Analysing customer data to determine preferences, purchase patterns, and demographics.
- **Inventory Optimization:** Leveraging insights to optimize inventory management, ensuring the availability of top-selling products and reducing carrying costs.
- **Marketing Strategy Enhancement:** Using data-driven insights to refine marketing strategies, such as targeted promotions and product recommendations.
- **Data Visualization:** Utilizing IBM Cognos for data visualization to create intuitive dashboards and reports.
- **Predictive Modelling:** Integrating Python for advanced analysis, including predictive modelling to forecast future sales trends.
- **Continuous Monitoring:** Providing actionable insights for ongoing monitoring to ensure sustained business growth.

PROJECT GOALS

Develop AI models to predict future product sales accurately. These predictions will help businesses plan inventory, set sales targets, and optimize marketing efforts.

- **Customer Sales Analysis**: Utilize AI to understand customer behaviours, preferences, and purchase patterns. This analysis can lead to personalized marketing strategies and improved customer experiences.
- **Data Driven Insights:** Extract meaningful insights from sales data using AI techniques such as machine learning, natural language processing, and image recognition.
- **Dynamic Pricing Strategies**: Implement AI algorithms for dynamic pricing, adjusting product prices in real-time based on demand, competition, and other relevant factors**.**
- **User interface visualization**: Develop a user-friendly dashboard or interface for users to interact with and visualize sales data and AI-generated insights.

METHODOLOGY

1.Data Collection and Preparation

Collect historical sales data, including product attributes, customer information, and external factors. Clean, preprocess, and feature engineer the data for AI model input.

2. Sales Prediction

Develop predictive models using AI algorithms (e.g., regression, time series analysis) to forecast future product sales. Evaluate model performance and fine-tune as necessary.

3. Customer Behaviour Analysis

Utilize AI techniques, such as clustering and recommendation systems, to analyse customer behaviour and preferences. Segment customers based on their

4. Data-Driven Insights

Apply AI methods like natural language processing to analyse customer reviews and feedback. Implement image recognition for product popularity analysis.

DATA COLLECTION PROCESS

- **Data Source:** Obtaining data from various sources, such as sales records, transaction logs, and customer databases.

- **Data Columns:** Understanding the dataset's columns, including product identifiers, sales dates, quantities sold, prices, and customer information.
- **Data Frequency:** Recognizing the frequency of data collection, which may be daily, weekly, or monthly.
- **Data Accuracy and Completeness:** Ensuring the accuracy and completeness of collected data through validation and cleansing.
- **Data Privacy and Compliance:** Complying with data privacy regulations, particularly regarding customer information.
- **Data Preprocessing:** Preprocessing steps, including handling missing values, data transformations, and aggregations, as required for analysis.
- **Data Storage:** Securely storing the dataset to maintain data integrity and confidentiality.
- **Data Documentation:** Thorough documentation of the dataset, including column descriptions, units of measurement, and any data transformations.
- **Data Retention:** Determining data retention policies in line with organizational requirements.
- **Data Validation:** Periodically validating data to ensure accuracy and consistency through quality checks and cross-referencing with other sources.
- **Data cleaning and preprocessing:** Data cleaning and preprocessing are critical steps in product sales analysis, as they ensure that the data is accurate, consistent, and ready for analysis.

EXPLORATORY DATA ANALYSIS

- **Data visualization**

  Create visualizations to gain initial insights.

- **Histogram**: Plot histograms to understand the distribution of numerical variables, especially sales amounts. Use box plots to identify outliers and understand the spread of the data.
- **Time Series Plots:** If your data includes time-based information, create time series plots to observe trends over time.
- **Bar Charts:** Create bar charts to visualize categorical data, such as product categories or customer types.
- **scatter plots:** Plot scatter plots to explore relationships between variables, e.g., sales vs. price.

CREATE VISUALISATION

Creating impactful visualizations using IBM Cognos involves:

- **Accessing the Reporting Tool:** Opening the IBM Cognos Report Authoring Tool or the relevant interface.
- **Selecting or Creating a Report:** Choosing or creating reports based on analysis objectives.
- **Adding Data:** Connecting reports to data sources containing relevant sales information.
- **Selecting Visualization Type:** Choosing suitable visualization types, such as charts, graphs, or tables.
- **Customizing Visualizations:** Tailoring the appearance of visualizations by adjusting co, labels, legends, and tooltips.
- **Applying Filters:** Using filters to focus on specific data subsets interactively.
- **Grouping and Aggregating:** Creating summary statistics or aggregates within visualizations for deeper insights.
- **Adding Interactivity:** Enhancing interactivity with features like drill-through options and tooltips for detailed information
- **Testing and Previewing:** Ensuring the accuracy and effective representation of data within the authoring tool.
- **Saving and Publishing:** Saving reports and making them.

Documentation: Documenting details of visualizations, including data sources and customization choices.

PYTHON INTEGRATION

The incorporation of Python for advanced analysis offers opportunities to gain deeper insights into sales data. Key applications include:

- **Predictive Sales Modelling:** Utilizing Python libraries like pandas and scikit-learn for predictive modelling to forecast future sales trends, enabling better inventory management and demand forecasting.
- **Customer Segmentation:** Employing machine learning algorithms to segment customers based on preferences and behaviour, facilitating targeted marketing campaigns.
- **Price Optimization:** Analysing pricing strategies using Python to maximize sales and profitability.
- **Inventory Forecasting:** Developing models for inventory forecasting to ensure optimal stock levels.
- **Product Recommendation Engines:** Implementing recommendation systems to suggest related products based on customer purchase history.

SALES PERFORMANCE EVALUATION

Sales performance evaluation in product sales analysis is a critical aspect of assessing the success and efficiency of your sales strategies and product offerings. It involves analysing various sales metrics and KPIs (Key Performance Indicators) to understand the performance of products and sales teams.

Identifying the key metrics and KPIs that are relevant to your business and product sales analysis. Common metrics include:

- Sales Revenue: The total revenue generated from product sales.
- Sales Volume: The number of units or products sold.
- Average Transaction value: The average amount customers spend per transaction.
- Gross margin: The profit made from sales after deducting the cost of goods sold.

- customer Acquisition Cost: The cost of acquiring a new customer.
- customer lifetime value: The total value a customer brings throughout their engagement with the company.
- sales growth rate: The rate at which sales revenue or volume is growing.
- Market share: Your company's share of the total market sales.
- Sales Conversion rate: The percentage of leads or prospects that convert into customers.
- Customer Churn Rate: The rate at which customers stop doing business with your company.

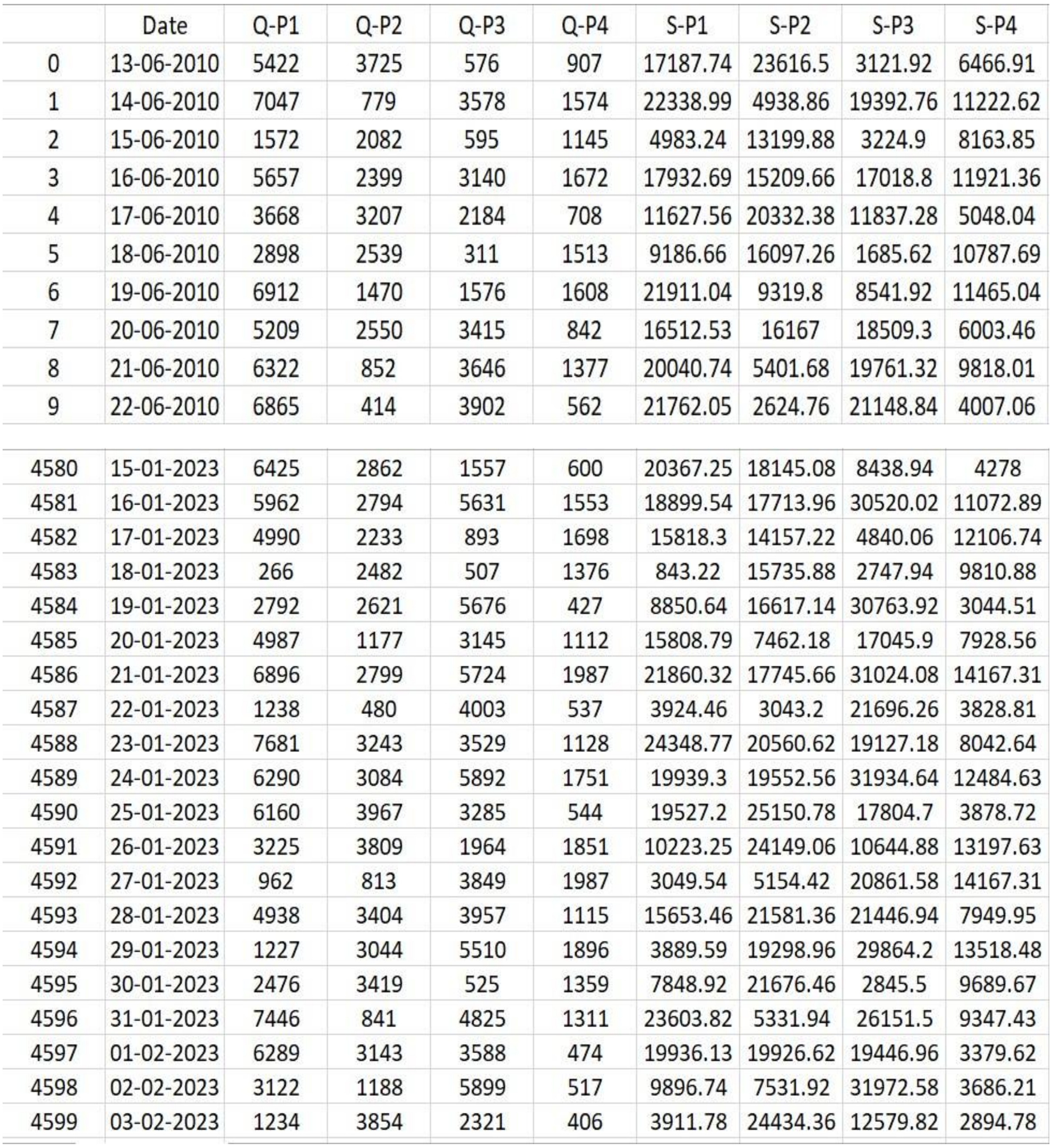
TOOLS FOR PRODUCT SALES ANALYSIS

- **Spreadsheets:** Basic tools like Microsoft Excel or Google Sheets can be used for data analysis and visualization.
- **Business intelligence software:** BI tools such as Tableau, Power BI, and QlikView offer more advanced data visualization and reporting capabilities.
- **Statistical Software:** Software like R or Python with libraries like pandas, NumPy, and Matplotlib are suitable for statistical analysis and visualization.
- **Data Analytics Platform:** Enterprise-level platforms like SAP BusinessObjects or IBM Cognos offer comprehensive sales analysis features.
- **Specialised Sales Analytics Software:** There are industry-specific tools tailored for sales analysis in various domains like retail, e-commerce, and manufacturing.

INNOVATION

- **Predictive Analysis and machine learning**
  Utilize advanced machine learning algorithms to predict future sales trends, demand patterns, and customer behaviour. Predictive models can provide more accurate forecasts and enable proactive decision-making.

GIVEN DATASET



	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

DATA CLEANING

Remove duplicates, handle missing values, and standardize data types to 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])
```

IMPORTANCE OF LOADING AND PREPROCESSING

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

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.

PROGRAM

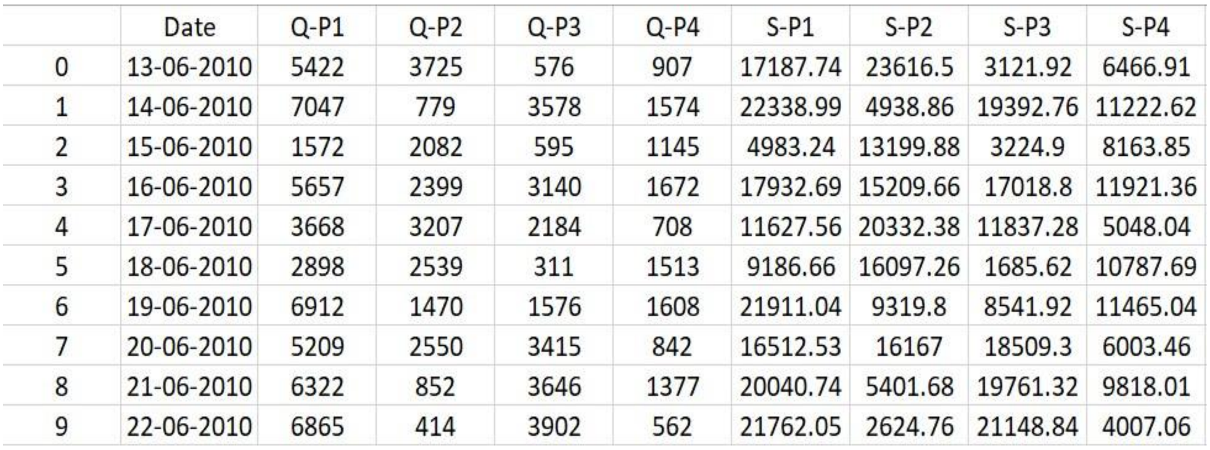
```
<chart>
  <options>
    <chart Size autoscaling="yes"/>
  </options>
  <type>Bar</type>
  <data>
    <crosstab>
      <data>
        <value name="Product" expr="[Product].[Product Name]"/>
        <value name="Sales" expr="total([Sales])"/>
      </data>
    </crosstab>
  </data>
</chart>
import 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
# For example, running this (by clicking run or pressing Shift Enter) will list all files under the input directory

import os
```

```
for drimane, _, filenames in os.walk('/Kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

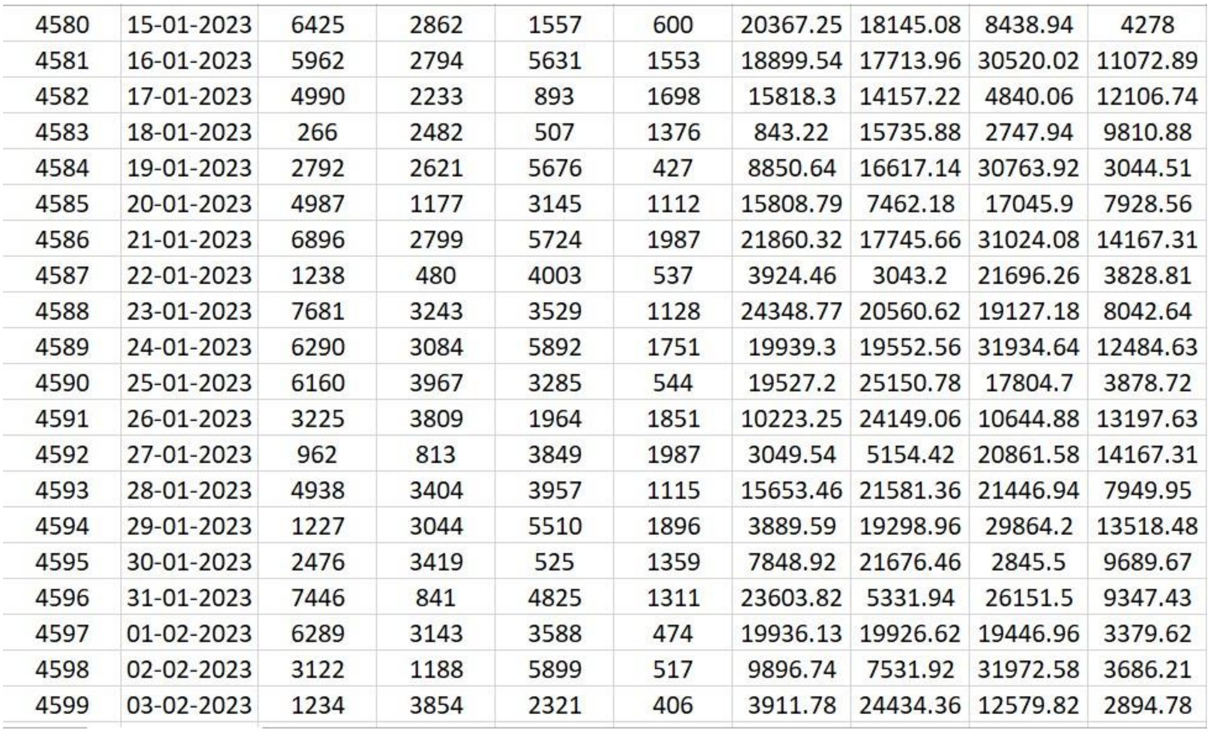
Loading dataset

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

OUTPUT



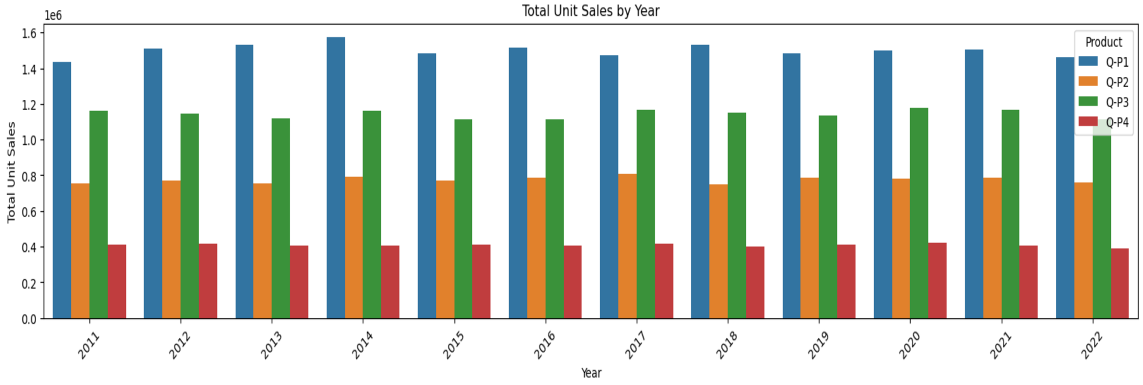
	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	13-06-2010	5422	3725	576	907	17187.74	23616.5	3121.92	6466.91
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.9	8163.85
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.8	11921.36
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



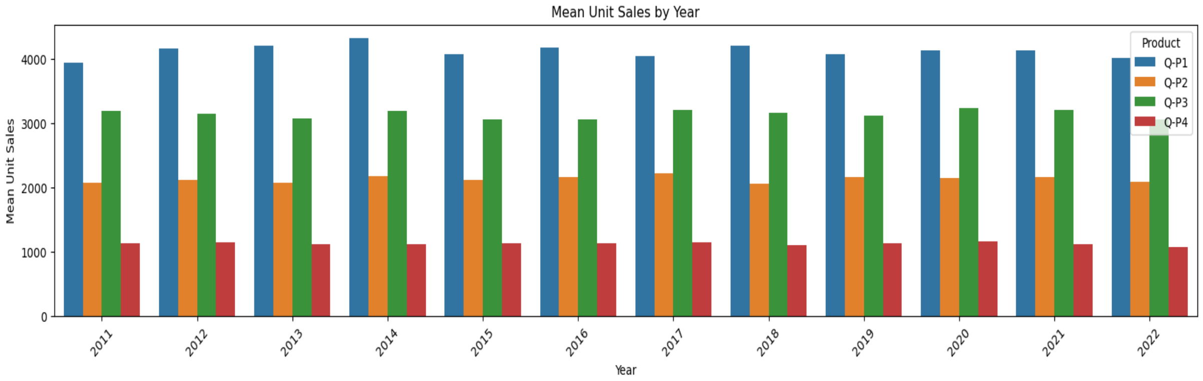
4580	15-01-2023	6425	2862	1557	600	20367.25	18145.08	8438.94	4278
4581	16-01-2023	5962	2794	5631	1553	18899.54	17713.96	30520.02	11072.89
4582	17-01-2023	4990	2233	893	1698	15818.3	14157.22	4840.06	12106.74
4583	18-01-2023	266	2482	507	1376	843.22	15735.88	2747.94	9810.88
4584	19-01-2023	2792	2621	5676	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

VISUALIZATION AND PREPROCESSING OF DATA

```
#Create a function that allows us to plot a bar chart for the 4 products
def plot_bar_chart(df, columns, stri, str1, val):
    # Aggregate sales for each product by year, by sum or mean
    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()

    # Melt the data to make it easier to plot
    sales_by_year_melted = pd.melt(sales_by_year, id_vars='Year', value_vars
=columns, var_name='Product', value_name='Sales')

    # Create a bar chart
    plt.figure(figsize=(20,4))
    sns.barplot(data=sales_by_year_melted, x='Year', y='Sales', hue='Product')
```

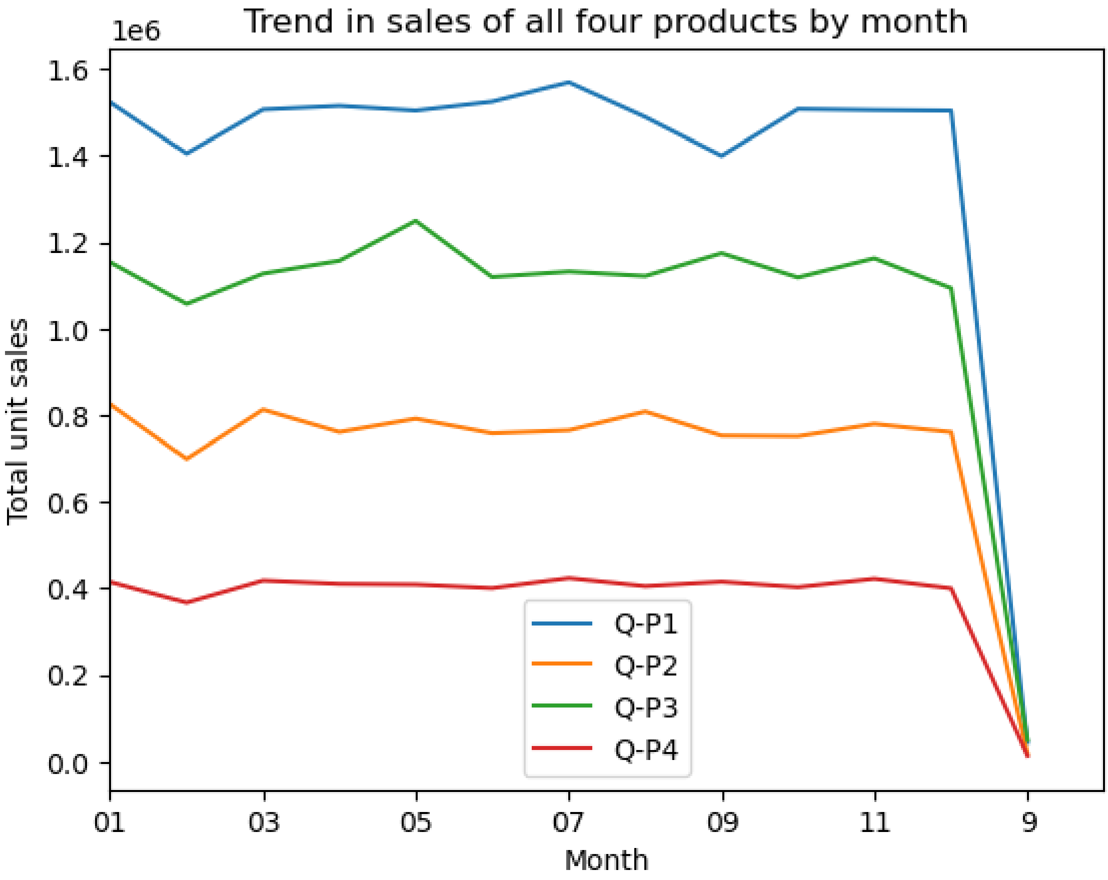


```
plt.xlabel('Year')
    plt.ylabel(stri)
    plt.title(f'{stri} by {str1}')
    plt.xticks(rotation=45)
    plt.show()
#use the plot_bar_chart function, enter the Unit Sales Columns and the Unit Sa
les string
 plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Total Unit Sales',
 'Year', 'sum')

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

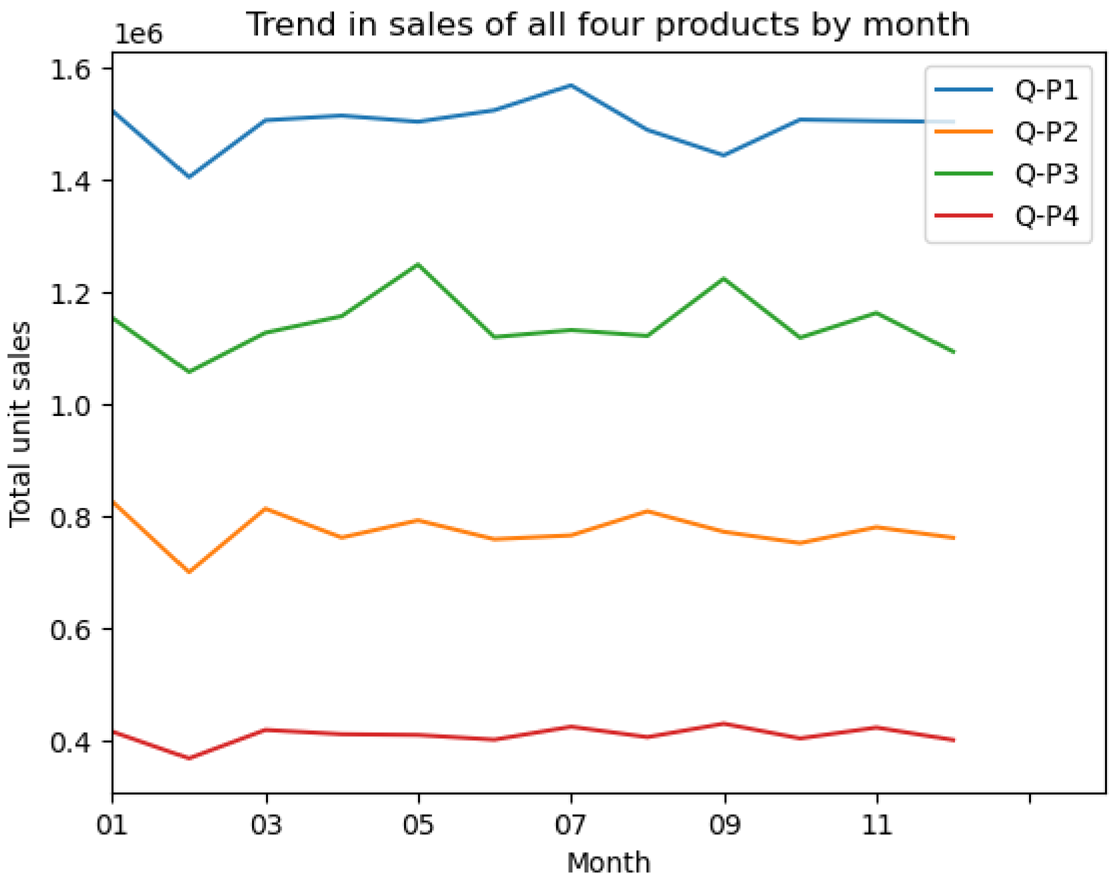


Create a figure and axis

```
def month_plot():
    fig, ax = plt.subplots()

    # Plot the sales data for each product by month
    data_reduced.groupby('Month')[['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']].sum()
.plot(ax=ax)

    # Set the x-axis limits to only show up to December
    ax.set_xlim(left=0, right=13)

    # Set the axis labels and title
    ax.set_xlabel('Month')
    ax.set_ylabel('Total unit sales')
ax.set_title('Trend in sales of all four products by month')

# Show the plot
plt.show()
```



```
# Replace all entries of '9' in the Month column with '09'
data_reduced['Month'] = data['Month'].replace('9', '09')
month_plot()
```



Visualization correlation

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

```
# gives us the average for all the 31st days across all years for each product
def avg_on_31st(df, product):
    df_31 = df[df['Day'] == '31']
    avg_sales = df_31[product].mean()
    return avg_sales
```

In [20]:

```
# Average for Unit Sales
avg_on_31st(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']).round(2)
```

```
# Average for Revenue
avg_on_31st(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4']).round(2)
```

AVERAGE FOR UNIT SALES
OUTPUT

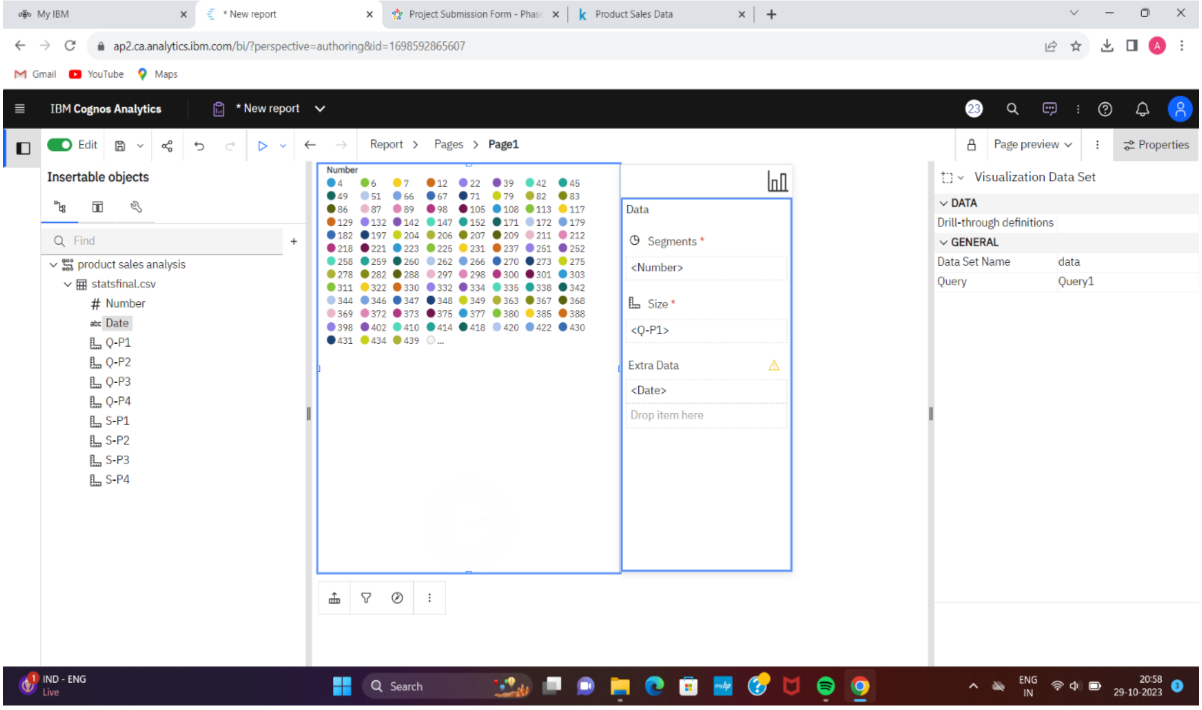
```
       Q-P1    3813.74
Q-P2    2058.80
Q-P3    3183.88
Q-P4    1098.61
dtype: float64

AVERAGE FOR REVENUE

OUTPUT

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

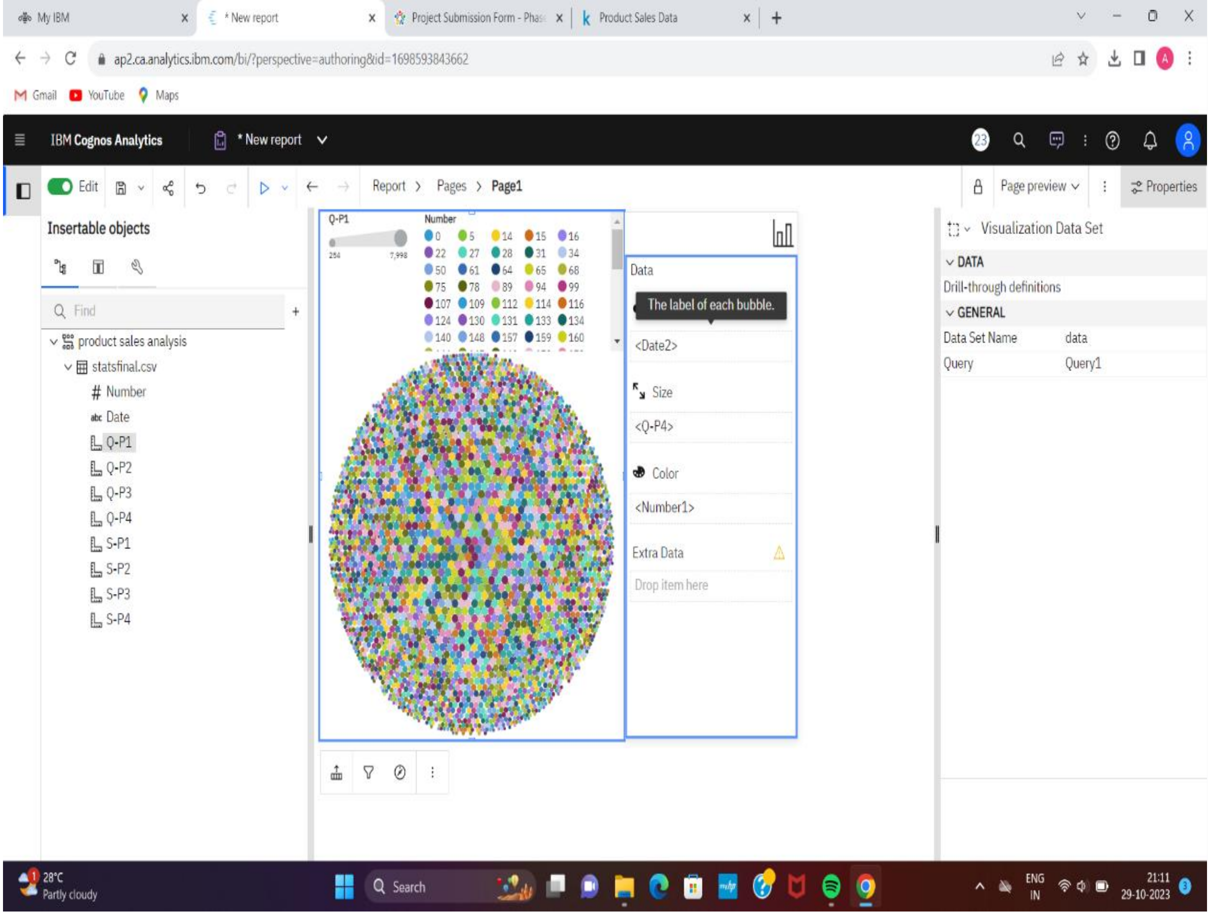
VISUALIZATION

Utilize IBM Cognos to create various types of visualizations that offer a comprehensive view of your product sales data. Common types of visualizations include bar charts, line charts, pie charts, heatmaps, scatter plots.

- **REVENUE AND SALES VOLUME TREND**
    - Create line charts to visualize trends in revenue and sales volume over time. This can help you identify seasonality, growth, or decline in sales.
    - Use trendlines to highlight the overall direction of sales and add annotations to explain significant changes or events that impacted sales.
- **ANALYSIS OF CUSTOMER SEGMENTION**
    - Develop pie charts or stacked bar charts to segment customer by demographics or buying behaviour.
    - For example, you can visualize the distribution of customers by age group, location, or purchase frequency.
    - Businesses use this information to better understand their target market, target marketing campaigns, and raise consumer satisfaction levels.

OUR DATASET REPORT





- **CUSTOMER PREFERENCES**
  Examine consumer preferences and behaviour. Utilize methods such as association rule mining to determine which products are frequently bought in tandem. This might assist you in making product recommendations or developing bundling plans.

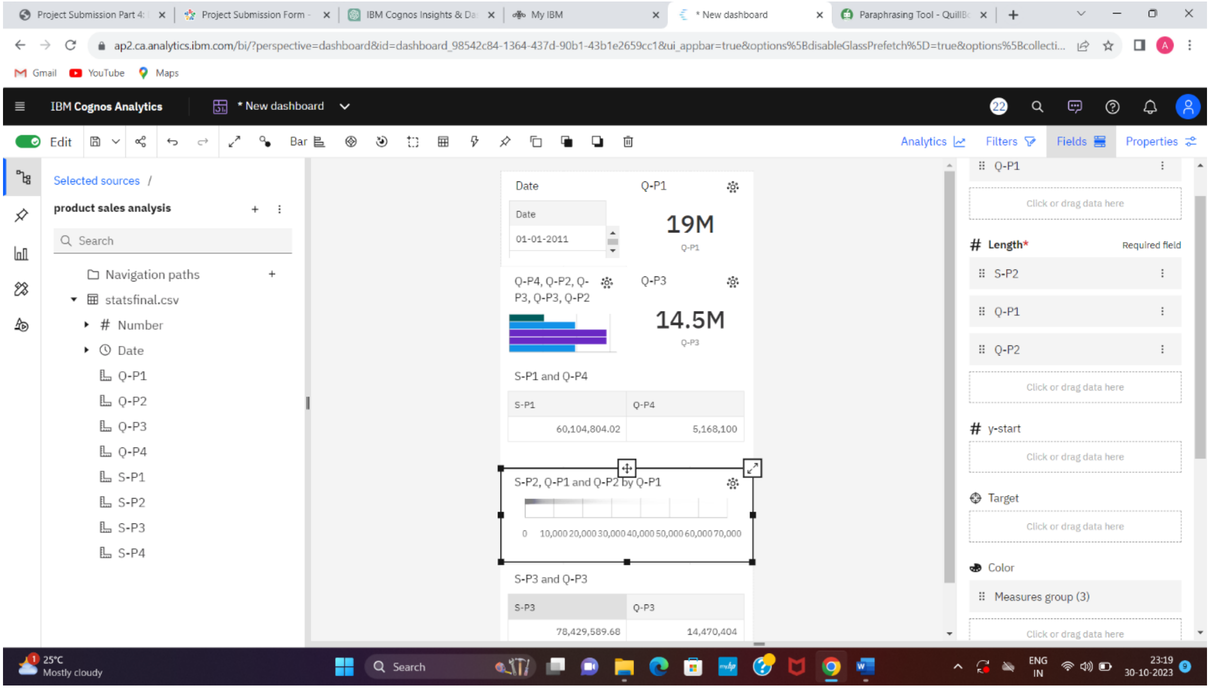
INTERATIVE DASHBOARDS

To display the insights and analysis findings, create interactive dashboards and reports in IBM Cognos. Stakeholders can examine the data and get practical insights from these dashboards.

- **SALES TRENDS**
  To see sales patterns over time, use an area or line chart. Put the total sales on the Y-axis and the date/time on the X-axis.
  Provide filters so users can change the date range.

OUR DATASET RELATED CREATED DASHBOARD

- In this first dashboard we will add the date and number and then comparing the product sales data and customer buying data.
- By comparing we will get highest sales in the product 3 and product 4 sales when compared to the product 1 and product 2.



Date
01-01-2011

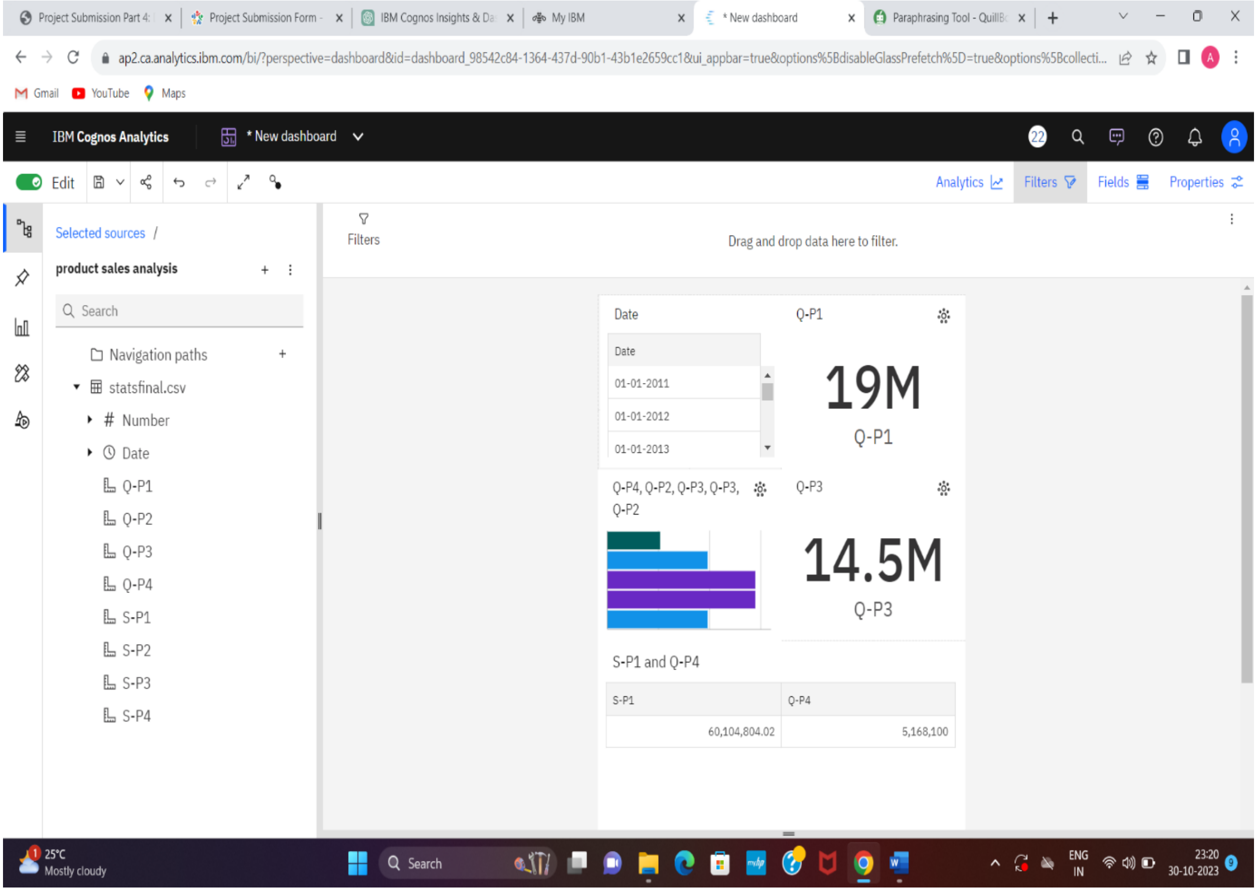
S-P1 and Q-P4

S-P1	Q-P4
60,104,804.02	5,168,100

S-P3 and Q-P3

S-P3	Q-P3
78,429,589.68	14,470,404

- In this dashboard we had compared the last year product sales with this year product sales.
- Include interactive features that let users choose which time periods to compare with by including a date range filter.
- When users click over each bar, include tooltips with detailed information (e.g., total sales values).



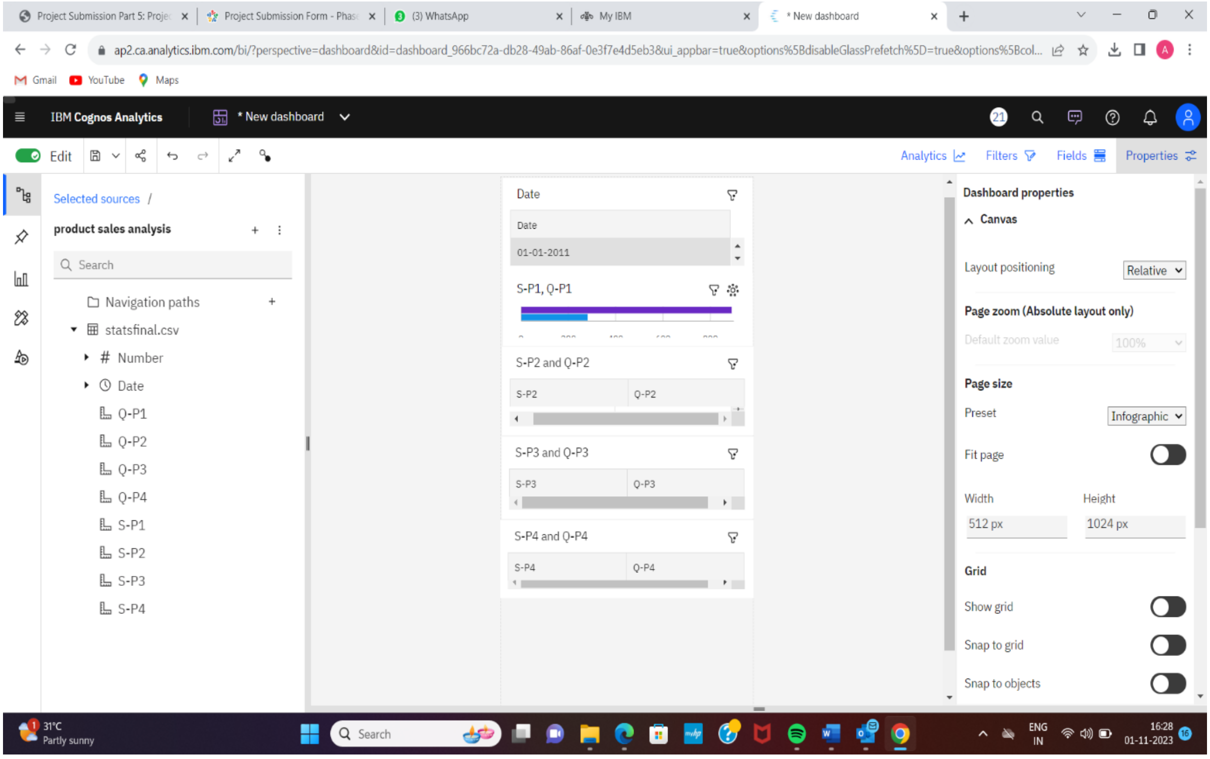
S-P1	Q-P4
60,104,804.02	5,168,100

- **IDENTIFY TRENDS**

  Examine which product categories had increases or decreases in sales over the course of the two periods.
- **SEASONAL PATTERNS**

  Seek out trends in sales that could indicate a relationship with marketing campaigns or seasonal circumstances.
- **ACTIONABLE INSIGHTS**

  Make practical inferences from our findings. For instance, plan to change inventories to match greater demand in growing categories or devote marketing resources to product categories that are dropping.



Dashboard properties

Canvas

Layout positioning: Relative

Page zoom (Absolute layout only)

Default zoom value: 100%

Page size

Preset: Infographic

Fit page

Width: 512 px

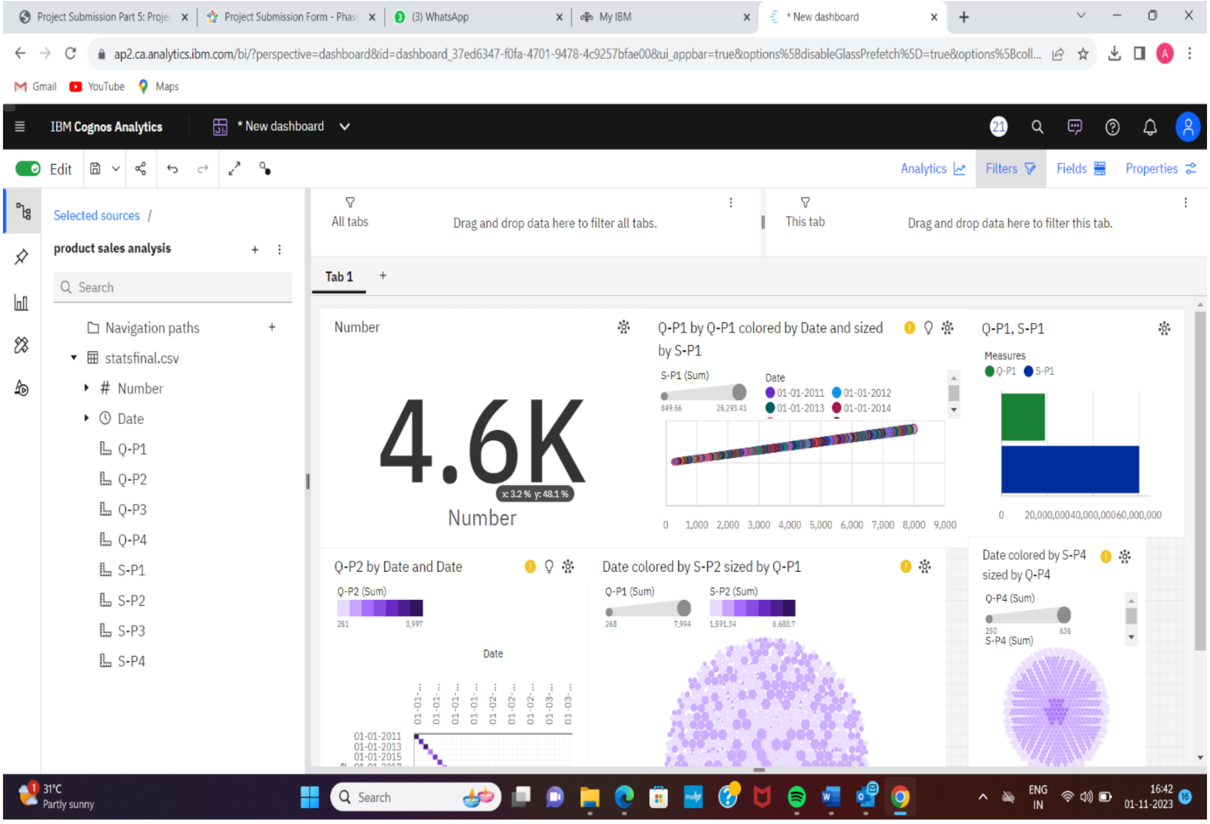
Height: 1024 px

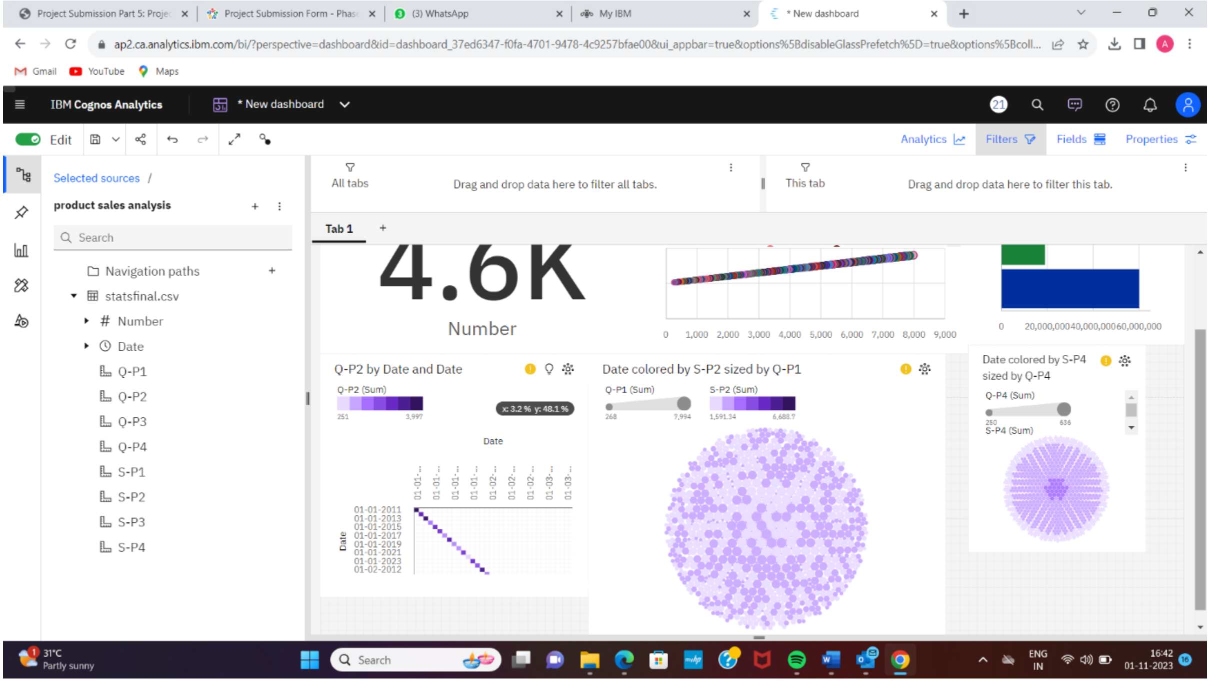
Grid

Show grid

Snap to grid

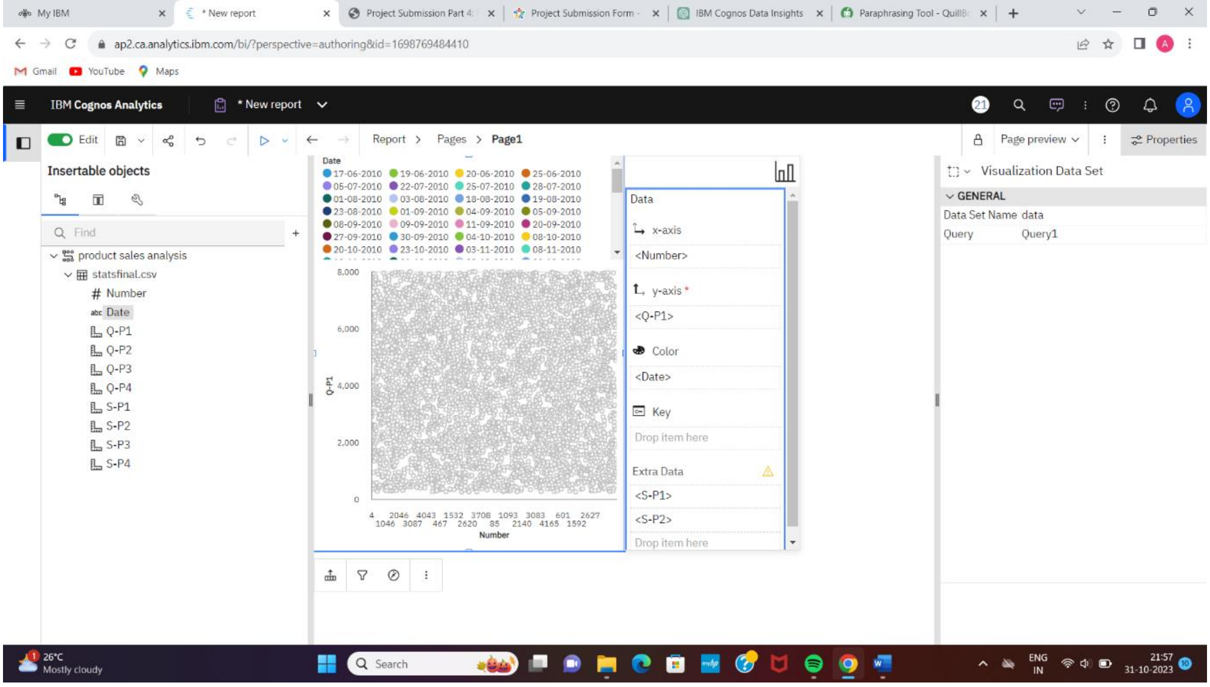
Snap to objects





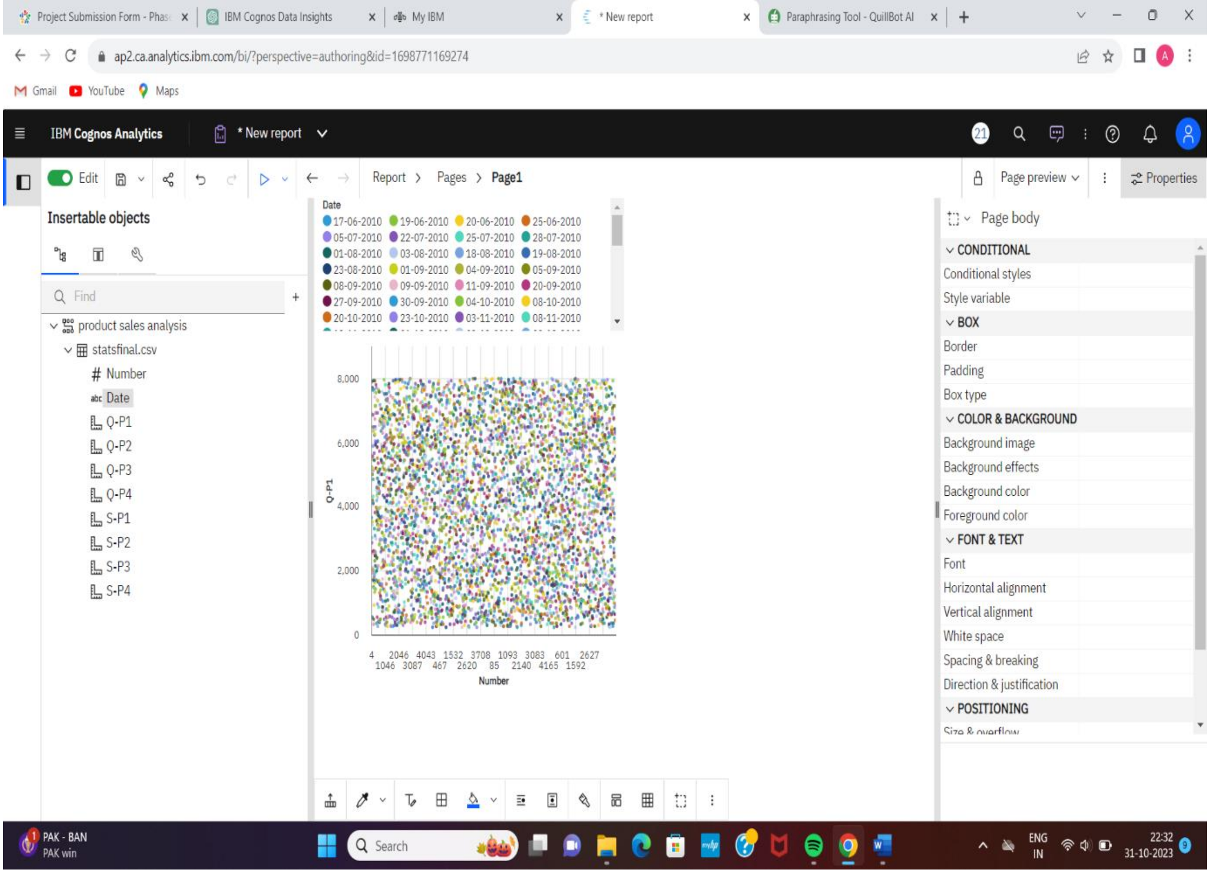
HIGHEST SALES

- Create a bar graph with the names of the products on the X-axis and the total sales on the Y-axis.
- The product with the most sales will be at the top of the list when the bars are sorted in descending order.
- To show the precise sales amount, add a data label to the top bar.



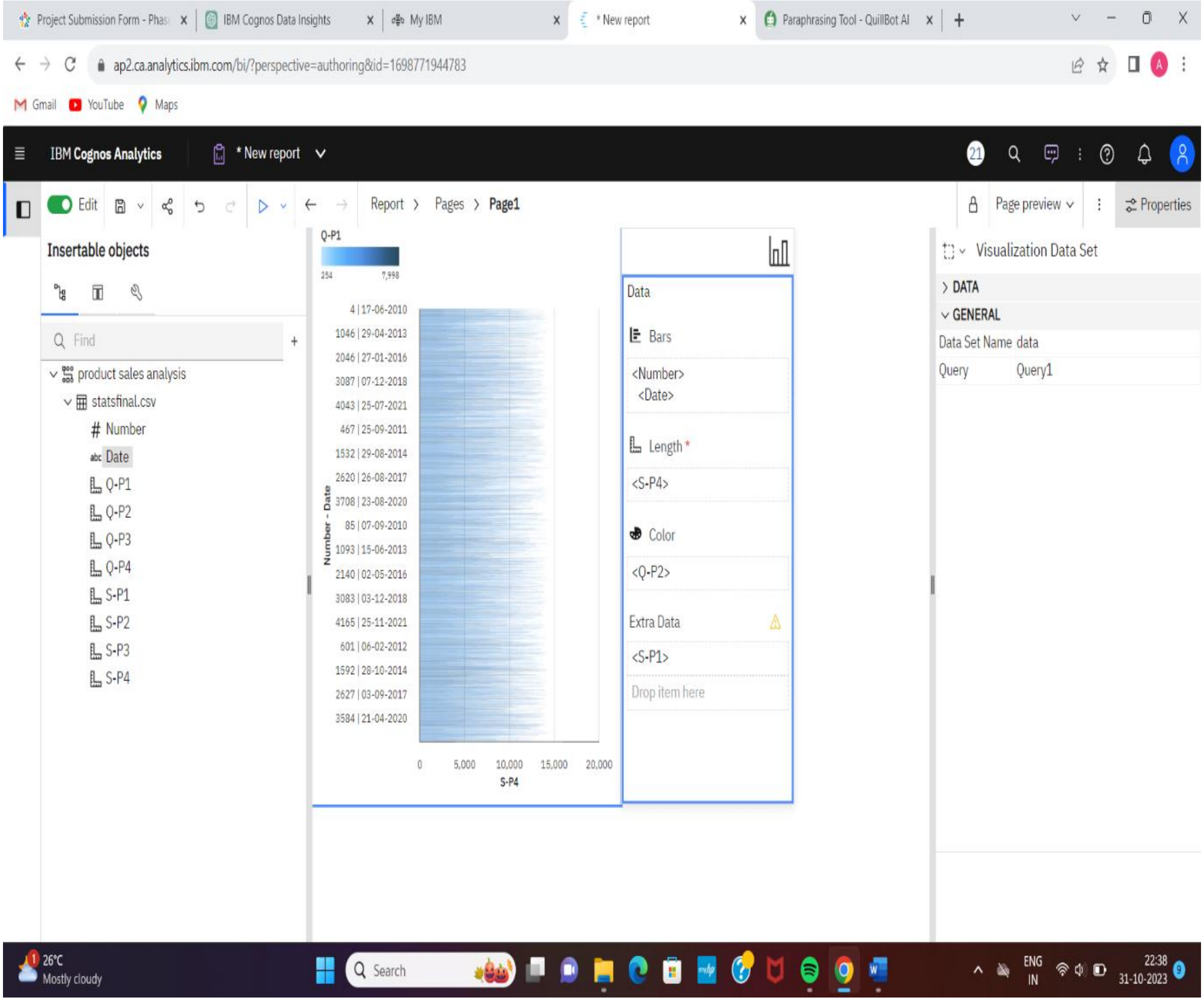
PEAK SALES PERIODS

- Make a line graph where the Y-axis represents total sales and the X-axis represents time intervals (such as daily, weekly, or monthly).
- To organize your data according to time intervals, use a temporal hierarchy.
- Incorporate trendlines or annotations to draw attention to moments of peak sales.



CUSTOMER PREFERENCES FOR SPECIFIC PRODUCTS

- Make a stacked bar chart that illustrates the differences in client preferences between several product categories.
- A bar represents each product category, and the height of each bar is segmented to represent distinct customer preferences (e.g., "like," "neutral," and "dislike").



Data

Bars
<Number>
<Date>

Length *
<S-P4>

Color
<Q-P2>

Extra Data
<S-P1>
Drop item here

Data Set Name data
Query Query1

SAMPLE LINE PLOT VISUALISATION

This is a scatter plot for defining the sales analysis by using a customer buying product details and dates of buying that product.

CONCLUSION

To conclude, this project improves inventory management and marketing tactics by utilizing data analysis, visualization, and insights. The project finds practical ideas that can result in cost savings, more customer happiness, and more revenue by adhering to an organized design thinking process and gathering and evaluating pertinent data. Data visualization with IBM Cognos offers an easy-to-use means of presenting insights and assisting in decision-making. By incorporating these insights into marketing and inventory plans, the company will be able to become more customer-focused, responsive, and agile, which will ultimately lead to long-term success.