FINAL SUBMISSION

PRODUCT SALES ANALYSIS PROJECT

Date	30-10-2023
Team ID	1289
Project Name	Product Sales Analysis

Table Content

Project Definition:

The project involves using IBM Cognos to analyze sales data and extract insights about top selling products, peak sales periods, and customer preferences. The objective is to help businesses improve inventory management and marketing strategies by understanding sales trends and customer behaviour. This project includes defining analysis objectives, collecting sales data, designing relevant visualizations in IBM Cognos, and deriving actionable insights.

1.Design Thinking:

Analysis Objectives:

1. **Identifying Top-Selling Products:**
   - Query the sales data to calculate the total sales volume and revenue generated for each product.

- Create a ranked list of products based on their sales performance.
- Visualize the top-selling products using bar charts or tables to make them easily accessible.

2. **Analyzing Sales Trends:**
   - Aggregate sales data by time intervals (e.g., months or quarters) to identify seasonal trends.
   - Use line graphs or time series charts to visualize sales trends over time.
   - Apply statistical techniques to identify significant changes or anomalies in sales patterns.
3. **Understanding Customer Preferences:**
   - Merge customer demographic data with sales data using common identifiers.
   - Group customers into segments based on demographics (e.g., age groups, gender, location).
   - Analyze the purchasing behaviour of different customer segments to understand their preferences.

Data Collection:

- Retrieve transaction records from your sales database or point-of-sale systems.
- Verify data accuracy and completeness, addressing any missing or erroneous records.

Product Information:

- Access product information from your inventory management system or product database.
- Ensure product data is up-to-date, including descriptions, categories, and pricing.

Customer Demographics:

- Collect customer demographic data through surveys, customer profiles, or CRM systems.
- Maintain data quality by regularly updating and validating customer information.
- Visualization Strategy: Crafting Effective Visuals

Interactive Dashboards:

- Design interactive dashboards in IBM Cognos that provide an overview of key sales metrics.

- Include filters and drill-down options to allow users to explore data interactively.

Marketing Strategies:

- Allocate marketing budgets to campaigns that have shown the highest return on investment (ROI) based on analysis.
- Refine targeting by tailoring marketing messages to customer segments with identified preferences.
- Monitor the effectiveness of marketing campaigns in real-time using dashboards and adjust strategies accordingly.

Product Development:

- Prioritize product development efforts based on the popularity of certain product categories or features.
- Gather customer feedback through surveys or reviews to inform product enhancements.
- Implement agile development processes to respond quickly to changing customer preferences.

Pricing Strategies:

- Adjust pricing strategies for underperforming products or categories to boost sales.
- Implement dynamic pricing models that respond to real-time market demand.
- Continuously analyze price elasticity and adjust pricing accordingly to maximize profitability.

2.Project Development:

Data Pre-Processing :

REC corp LTD. is small-scaled business venture established in India. They have been selling FOUR PRODUCTS for OVER TEN YEARS .

The products are:

- P1
- P2
- P3
- P4

They have collected data from their retail centers and organized it into a small csv file , which has been given to you.

The excel file contains about 8 numerical parameters :

- 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

Step 1: Import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Step 2: Loading the datasets

```
data = pd.read_csv("statsfinal.csv")
data
```

Unnamed 0	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	23616.50	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	3224.90	8163.85
3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36
4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04
...	...	...	...	...	...	...	...	...	...
4595	30-01-2023	2476	3419	525	1359	7848.92	21676.46	2845.50	9689.67
4596	31-01-2023	7446	841	4825	1311	23603.82	5331.94	26151.50	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

[4599 rows x 10 columns]

Step 3: Checking the info of the training data

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	Date	4600 non-null	object
1	Q-P1	4600 non-null	int64
2	Q-P2	4600 non-null	int64
3	Q-P3	4600 non-null	int64
4	Q-P4	4600 non-null	int64
5	S-P1	4600 non-null	float64

6	S-P2	4600 non-null	float64
7	S-P3	4600 non-null	float64
8	S-P4	4600 non-null	float64

```
dtypes: float64(4), int64(4), object(1)
    memory usage: 323.6+ KB
```

Step 4: Check for missing values

```
data.isnull().sum()
```

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
```

Step 5: EDA

```
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])
data
```

	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Date	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

Data Visualization:

Chart 1:

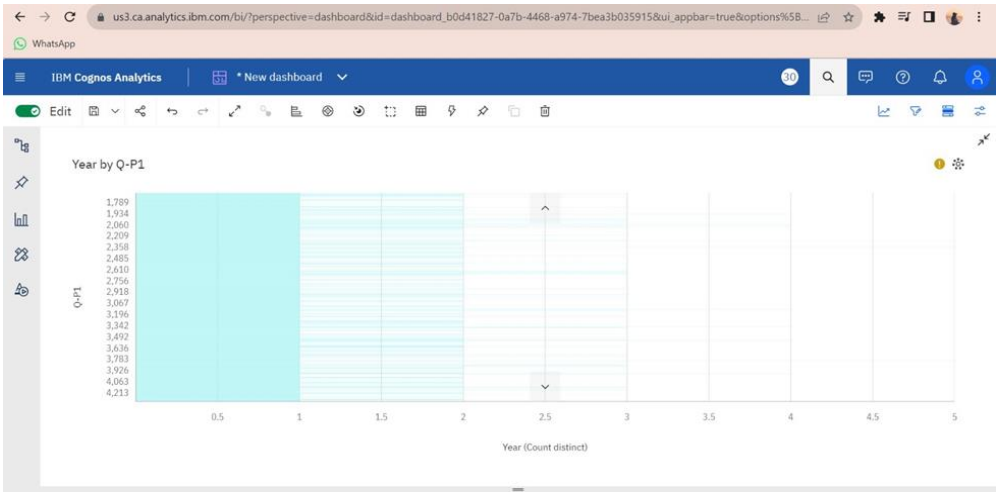


Chart Insights were not computed because this visualization is based on clipped data. Consider applying a filter to reduce the number of records, and to prevent the data from being clipped, before creating the visualization.

Chart 2:

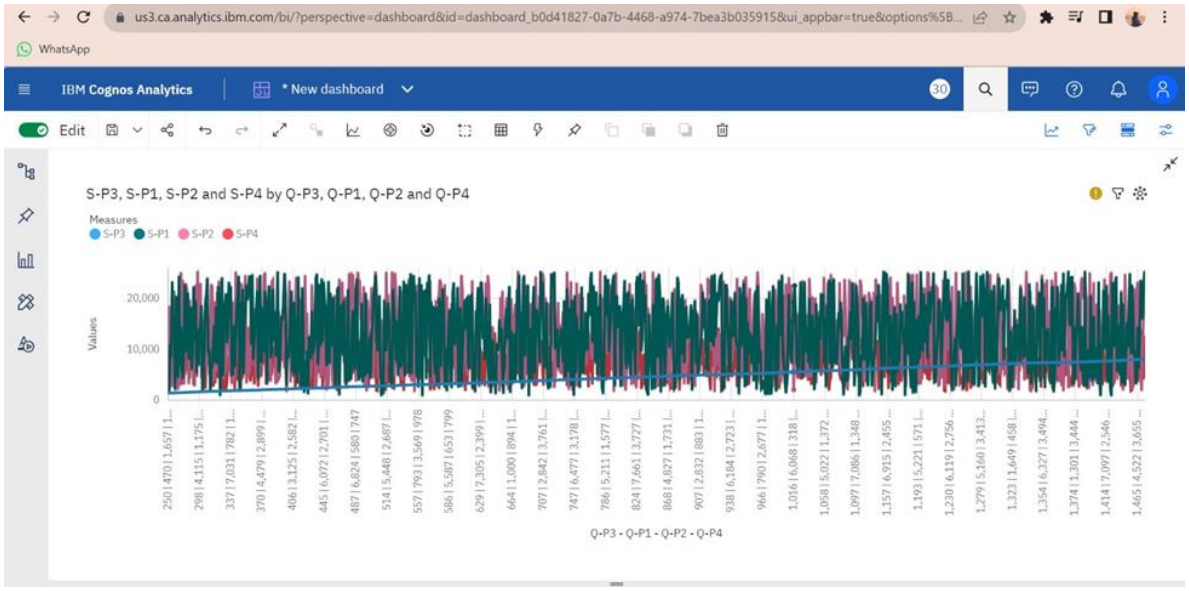
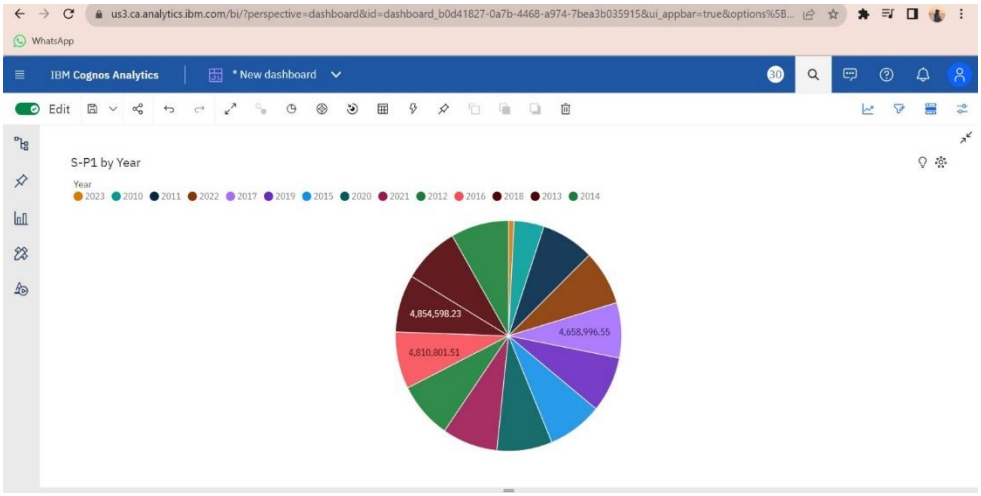


Chart Insights were not computed because this visualization is based on clipped data. Consider applying a filter to reduce the number of records, and to prevent the data from being clipped, before creating the visualization.

Chart 3:



- S-P1 is unusually low in 2023 and 2010.
- Based on the current forecasting, S-P1 may reach over 6.2 million by Year 2026.
- 2011 (7.9 %), 2022 (7.9 %), 2021 (7.9 %), 2019 (7.9 %), and 2018 (7.9 %) are the most frequently occurring categories of Year with a combined count of 1820 items with S-P1 values (39.6 % of the total) .
- 2011 (7.9 %), 2022 (7.9 %), 2021 (7.9 %), 2019 (7.9 %), and 2018 (7.9 %) are the most frequently occurring categories of Year with a combined count of 1820 items with S-P2 values (39.6 % of the total) .
- Across all years, the average of S-P1 is over thirteen thousand.
- Over all years, the average of S-P2 is nearly fourteen thousand.
- The total number of results for S-P1, across all years, is over 4500.
- The total number of results for S-P2, across all years, is over 4500.
- S-P1 ranges from over 476 thousand, in 2023, to almost 5.0 million, in 2014.
- S-P2 ranges from over 496 thousand, in 2023, to over 5.1 million, in 2017.

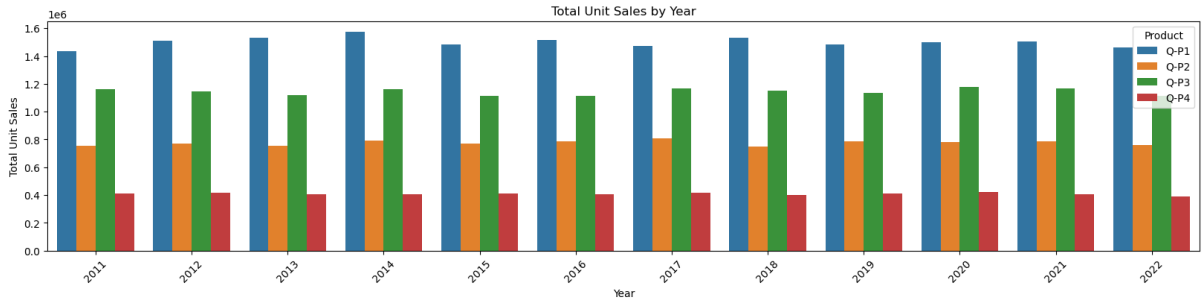
HISTOGRAM AND DISTRIBUTION:

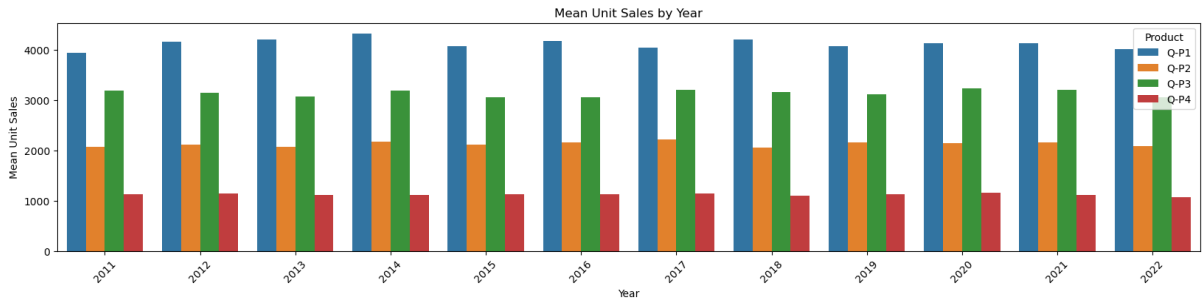
Graph our TOTAL & MEAN unit sold for each product using a histogram.

```
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, var_name='Product', value_name='Sales')

    plt.figure(figsize=(20,4))
```

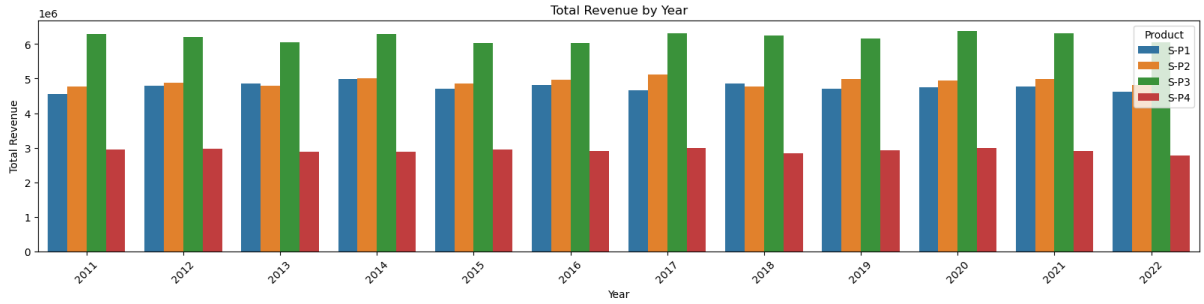
```
    sns.barplot(data=sales_by_year_melted, x='Year', y='Sales',
hue='Product') #,palette="cividis")
    plt.xlabel('Year')
    plt.ylabel(stri)
    plt.title(f'{stri} by {str1}')
    plt.xticks(rotation=45)
    plt.show()
```

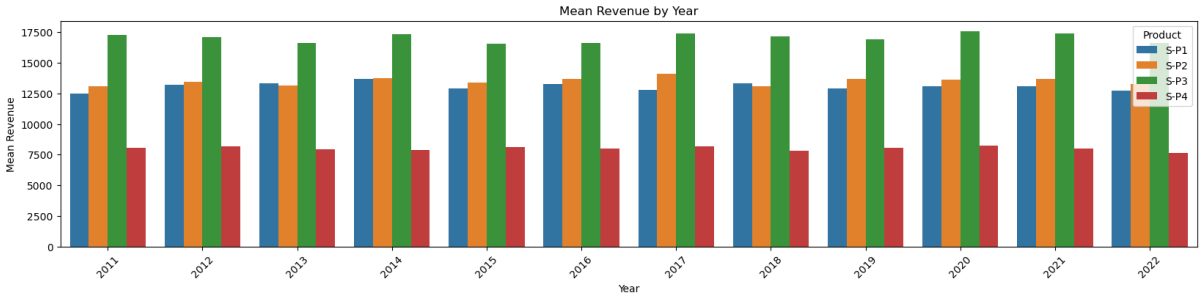
plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'QP4'],'Total Unit Sales', 'Year', 'sum')
plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'M ean Unit Sales', 'Year', 'mean')





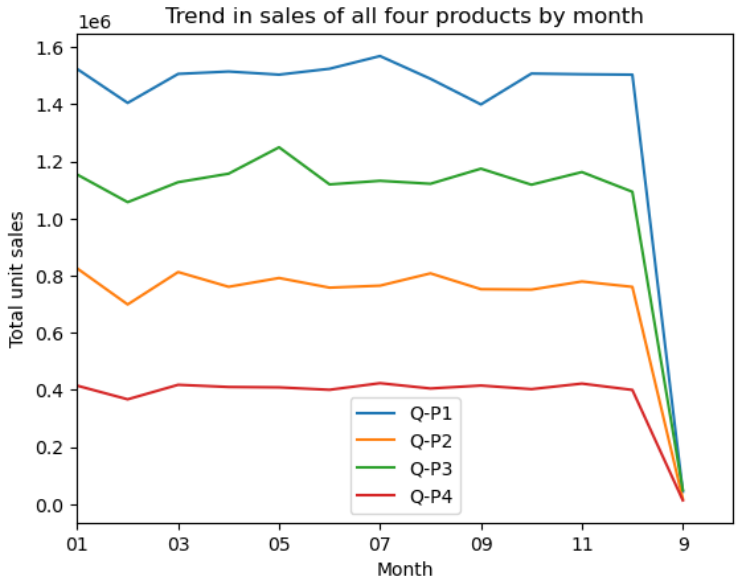

```
plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Total Revenue'
, 'Year', 'sum')

plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Mean Revenue',
'Year', 'mean')
```





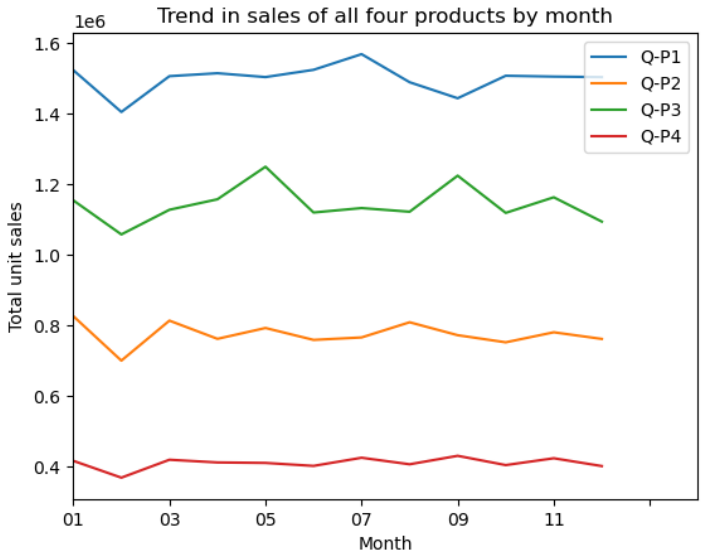
Trend in sales of all four products during certain months

```
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()
```

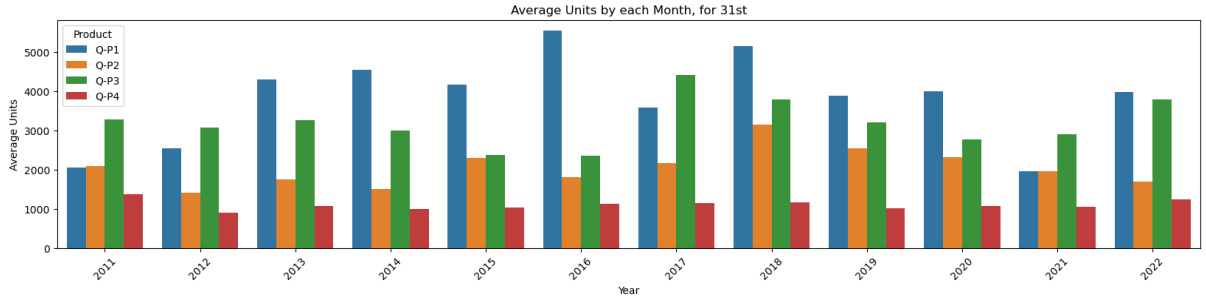



```
data_reduced['Month'] = data['Month'].replace('9', '09')
```

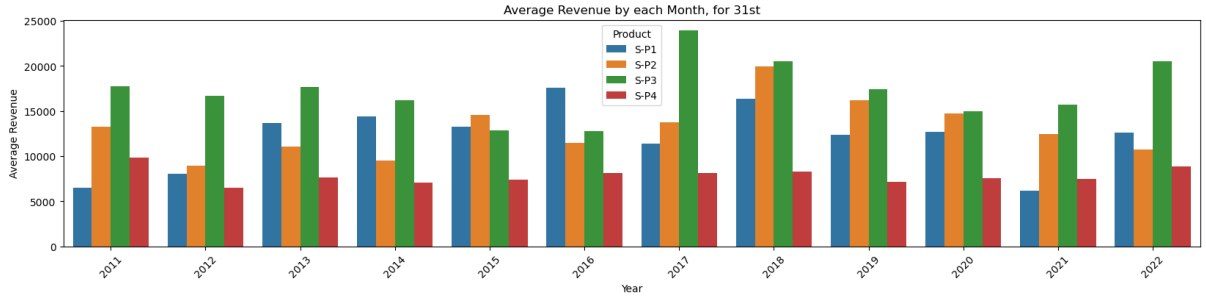
```
month_plot()
```




```
plot_bar_chart(_31_months, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Average  Units', 'each Month, for 31st', 'mean')
```




```
plot_bar_chart(_31_months, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Average Revenue', 'each Month, for 31st', 'mean')
```




```
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)
```

Q-P1	3813.74
Q-P2	2058.80
Q-P3	3183.88
Q-P4	1098.61

dtype: float64

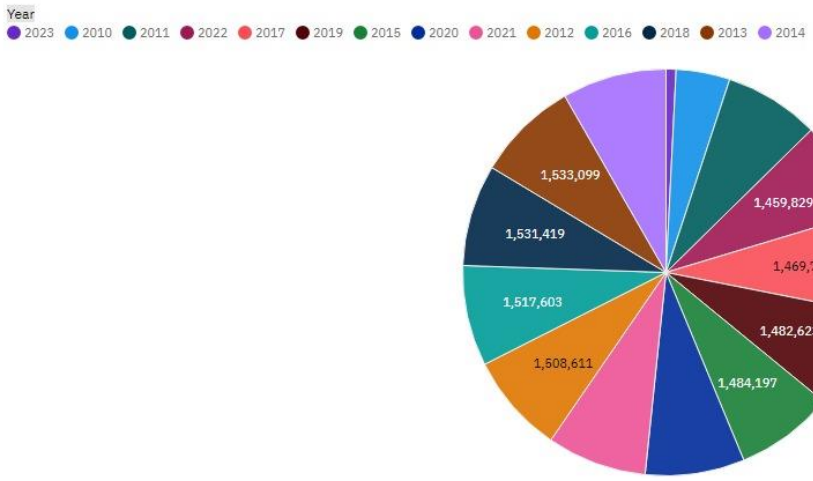
```
avg_on_31st(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4']).round(2)
```

S-P1	12089.55
S-P2	13052.78
S-P3	17256.63
S-P4	7833.07

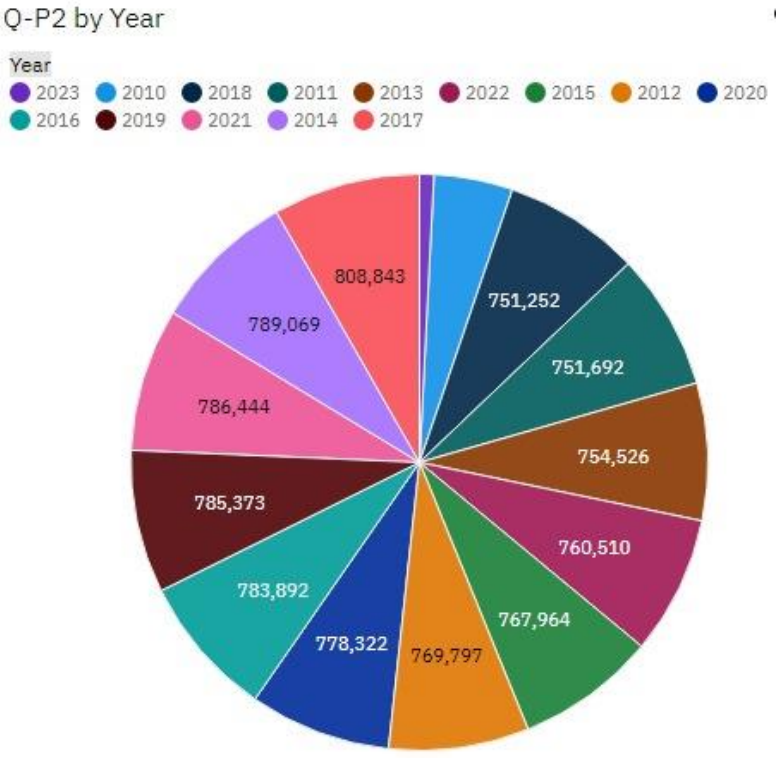
dtype: float64

Visualization:

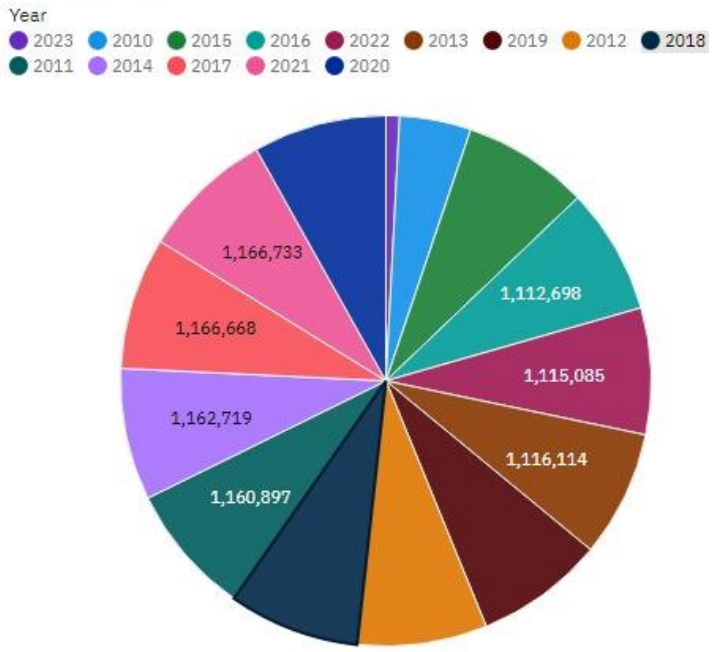
Q-P1 by Year



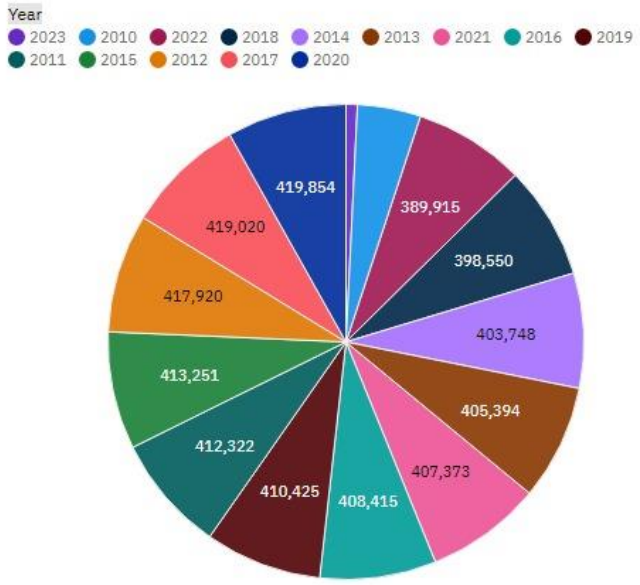
Q-P2 by Year



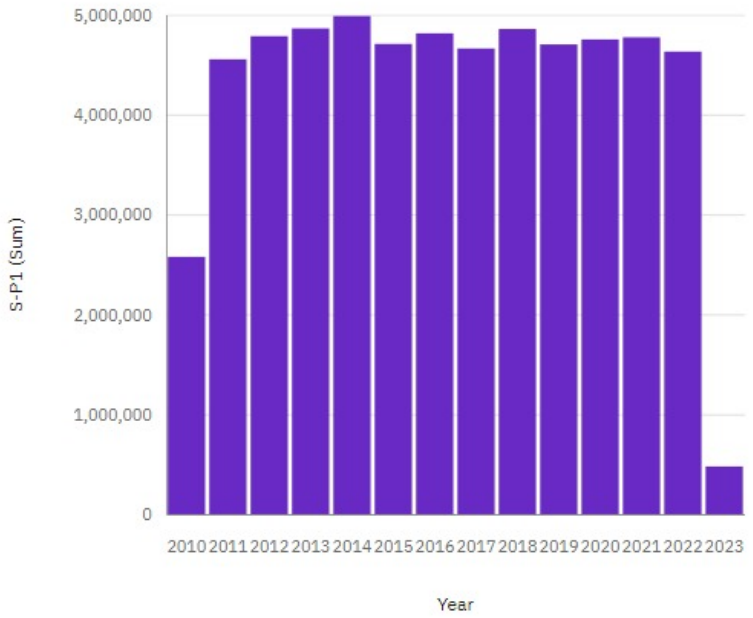
Q-P3 by Year



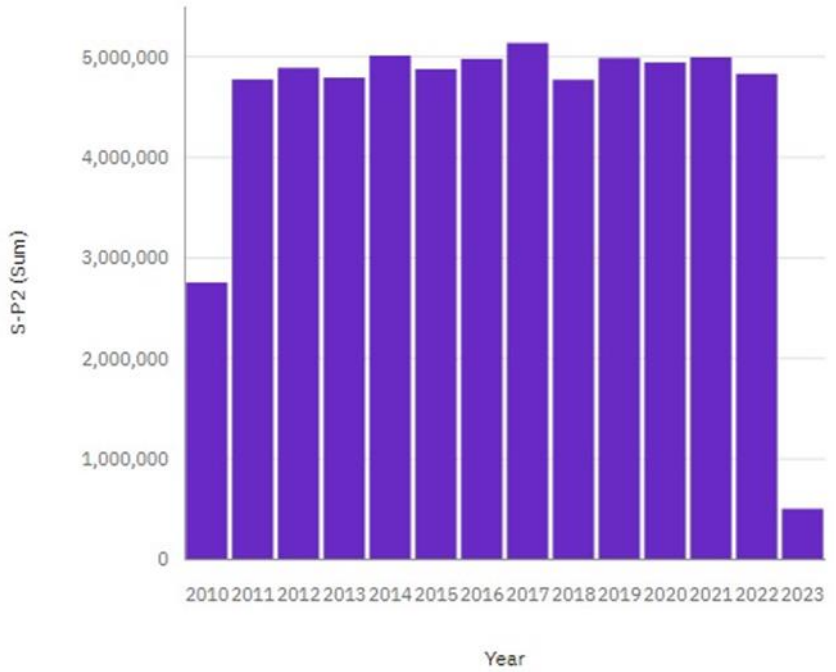
Q-P4 by Year

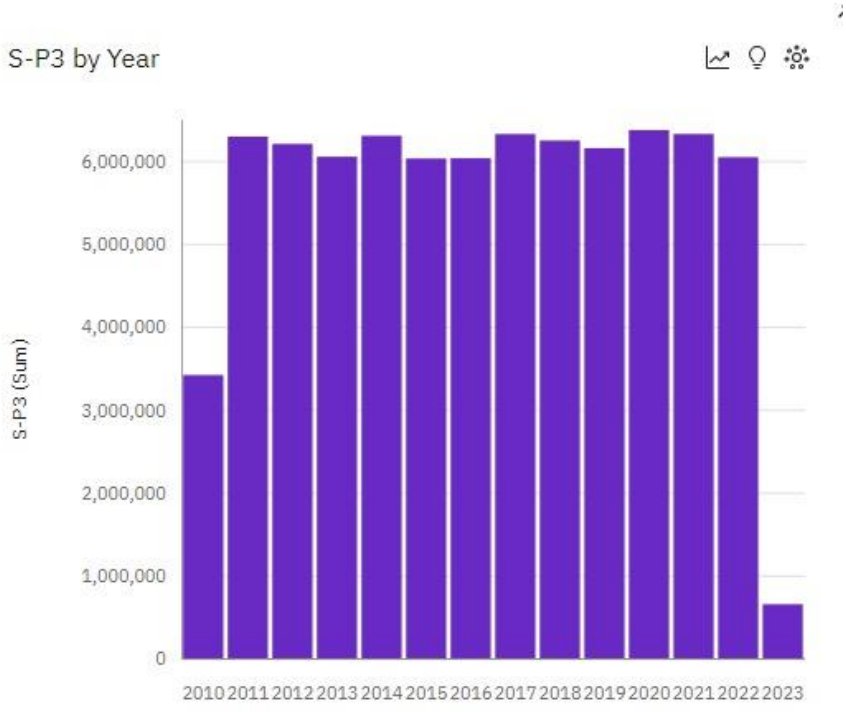


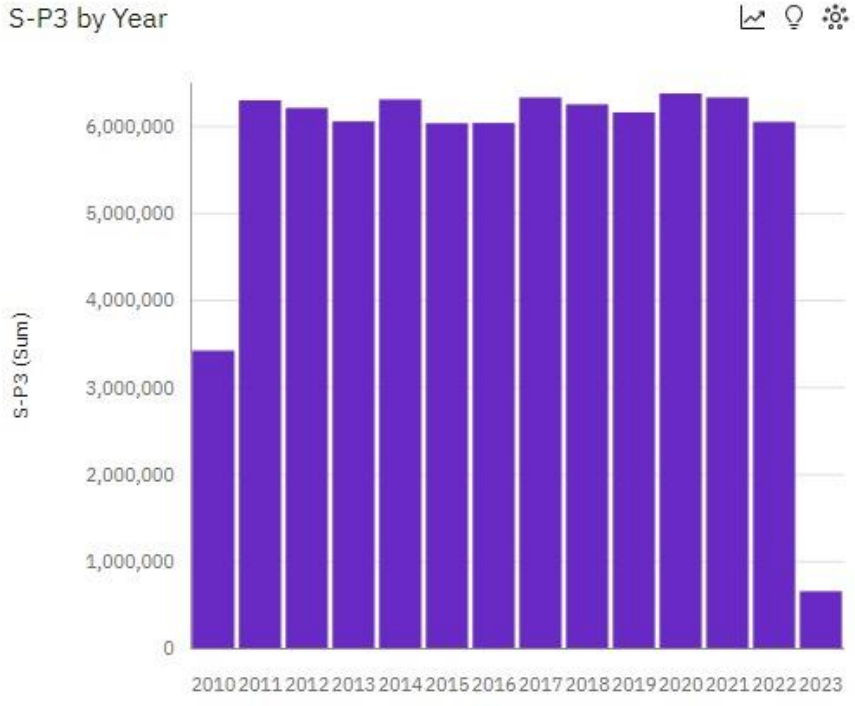
S-P1 by Year



S-P2 by Year




S-P3 by Year
S-P3 (Sum)
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023


S-P3 by Year
S-P3 (Sum)
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023