PRODUCT SALES DATA ANALYSIS :

Help the company in **finding trends** and **insights**

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

Now, **REC corp LTD** needs you to solve the following questions:

1) Is there **any trend in the sales** of all **four products** during **certain months?**

2) Out of all **four products** , which product has seen the **highest sales** in all the **given years?**

3) The **company** has all it's **retail centers** closed on the **31st of December every year**. Mr: Hariharan , the CEO , would love to get an estimate on **no: of units of each product** that could be sold on 31st of Dec , every year , if all their **retail centers** were kept open.

4) **The CEO** is considering an idea to **drop the production of any one of the products**. He wants you to **analyze this data and suggest** whether his idea would result in a **massive setback** for the **company**.

5) The CEO would also like to **predict the sales and revenues** for the **year 2024**. He wants you to give a **yearly estimate** with the best **possible accuracy**.

Can you help **REC corp ltd** with your **analytical and data science skills ?**

Step 1: Import libraries

```
[87] # import the important packages
     import pandas as pd  # library used for data manipulation and analysis
     import numpy as np  # library used for working with arrays
     import matplotlib.pyplot as plt  # library for plots and visualizations
     import seaborn as sns  # library for visualizations

     %matplotlib inline

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

Step 2: Loading the datasets

```
[88] #if you open in juypter notebook
     data = pd.read_csv('statsfinal.csv')
```

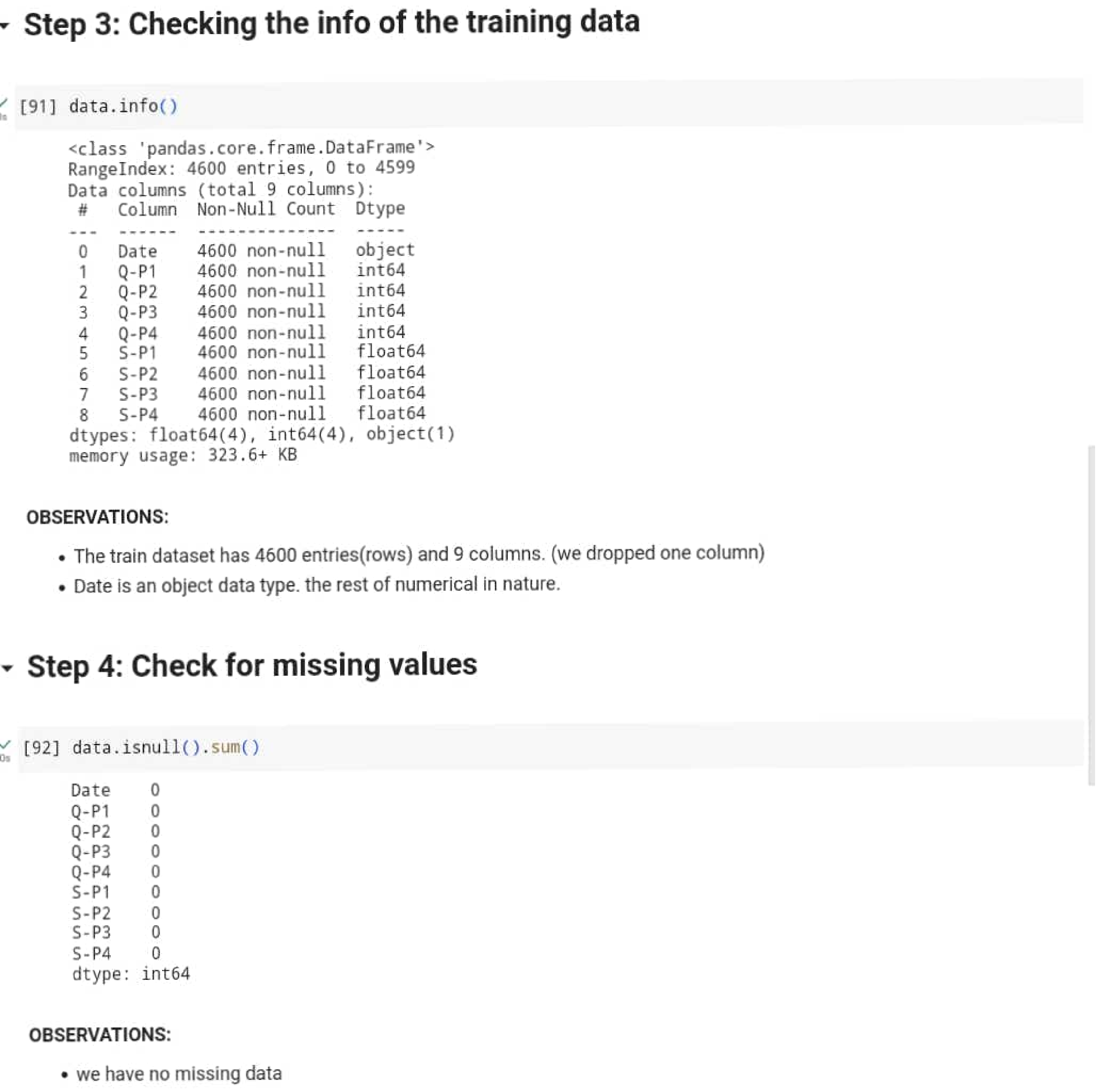
```
[89] # Checking the first 5 and last 5 rows of the dataset
     data.head(-1)
```

```
      Unnamed: 0        Date  Q-P1  Q-P2  Q-P3  Q-P4      S-P1      S-P2      S-P3      S-P4
   0           0  13-06-2010  5422  3725   576   907  17187.74  23616.50   3121.92   6466.91
   1           1  14-06-2010  7047   779  3578  1574  22338.99   4938.86  19392.76  11222.62
   2           2  15-06-2010  1572  2082   595  1145   4983.24  13199.88   3224.90   8163.85
   3           3  16-06-2010  5657  2399  3140  1672  17932.69  15209.66  17018.80  11921.36
   4           4  17-06-2010  3668  3207  2184   708  11627.56  20332.38  11837.28   5048.04
 ...         ...         ...   ...   ...   ...   ...       ...       ...       ...       ...
4594        4594  29-01-2023  1227  3044  5510  1896   3889.59  19298.96  29864.20  13518.48
4595        4595  30-01-2023  2476  3419   525  1359   7848.92  21676.46   2845.50   9689.67
4596        4596  31-01-2023  7446   841  4825  1311  23603.82   5331.94  26151.50   9347.43
4597        4597  01-02-2023  6289  3143  3588   474  19936.13  19926.62  19446.96   3379.62
4598        4598  02-02-2023  3122  1188  5899   517   9896.74   7531.92  31972.58   3686.21

4599 rows × 10 columns
```

OBSERVATIONS:

- We can observe the first entry in the data, starts at 13-06-2010. This means the data for year 2010 is not complete.
- We can observe the last entry in the data, ends at 02-02-2023. This means the data for year 2023 is also not complete. it will be best to drop year 2010 and year 2023.

```
[90] # drop the first column
     data = data.drop(columns=['Unnamed: 0'])
```



Step 3: Checking the info of the training data

```
[91] 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
```

OBSERVATIONS:

- The train dataset has 4600 entries(rows) and 9 columns. (we dropped one column)
- Date is an object data type. the rest of numerical in nature.

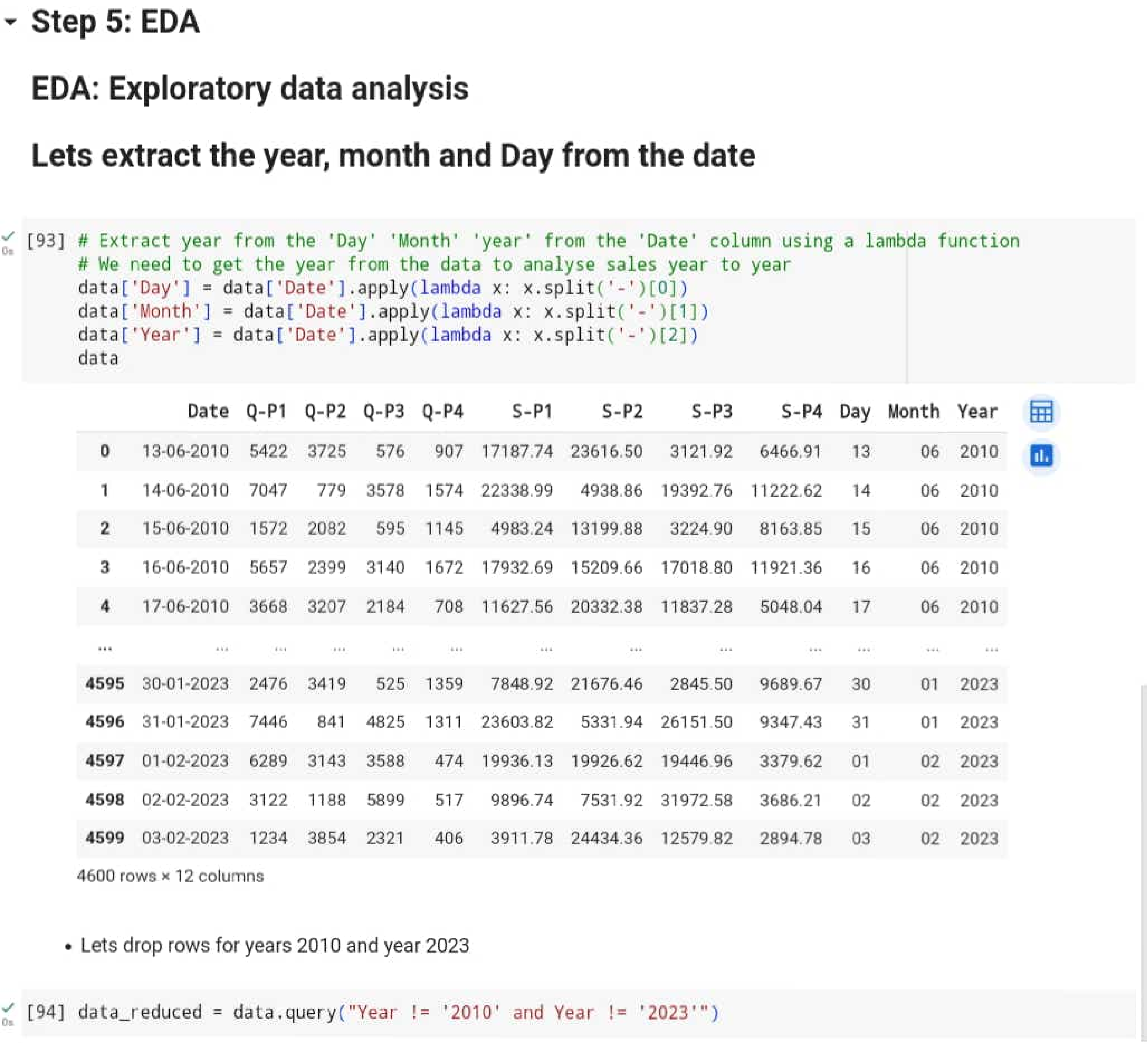
Step 4: Check for missing values

```
[92] 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
```

OBSERVATIONS:

- we have no missing data



Step 5: EDA

EDA: Exploratory data analysis

Lets extract the year, month and Day from the date

```
[93] # 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])
     data
```

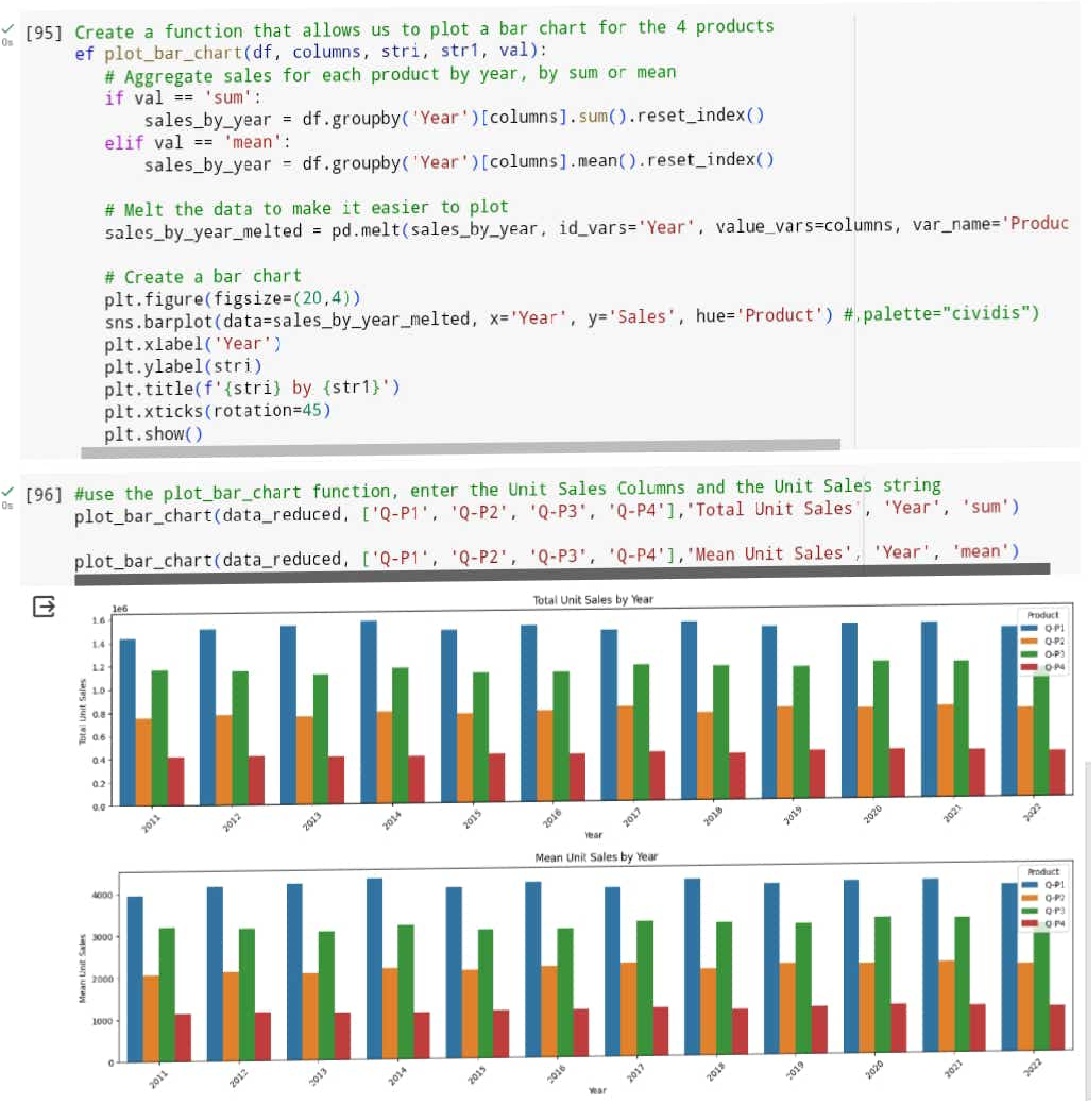
	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Day	Month	Year
0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91	13	06	2010
1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62	14	06	2010
2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85	15	06	2010
3	16-06-2010	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

4600 rows × 12 columns

- Lets drop rows for years 2010 and year 2023

```
[94] data_reduced = data.query("Year != '2010' and Year != '2023'")
```

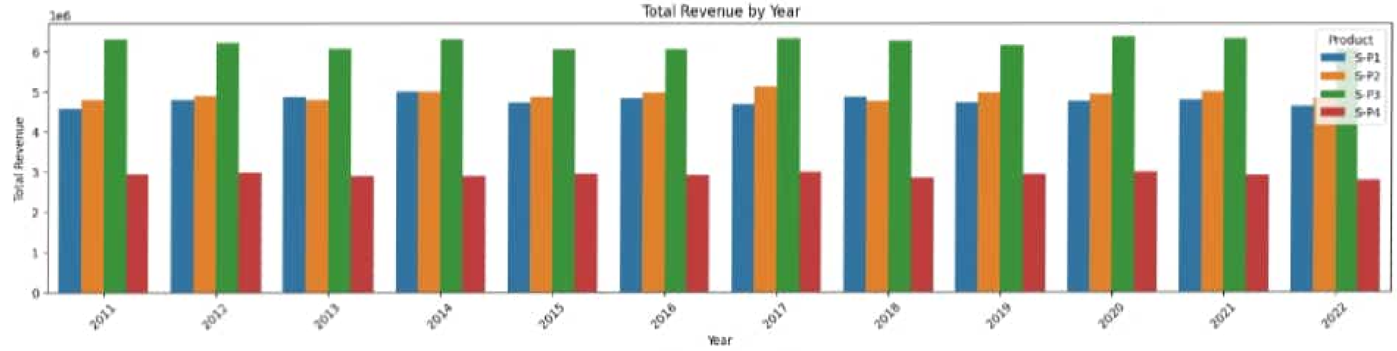
Graph our TOTAL & MEAN unit sold for each product using a histogram

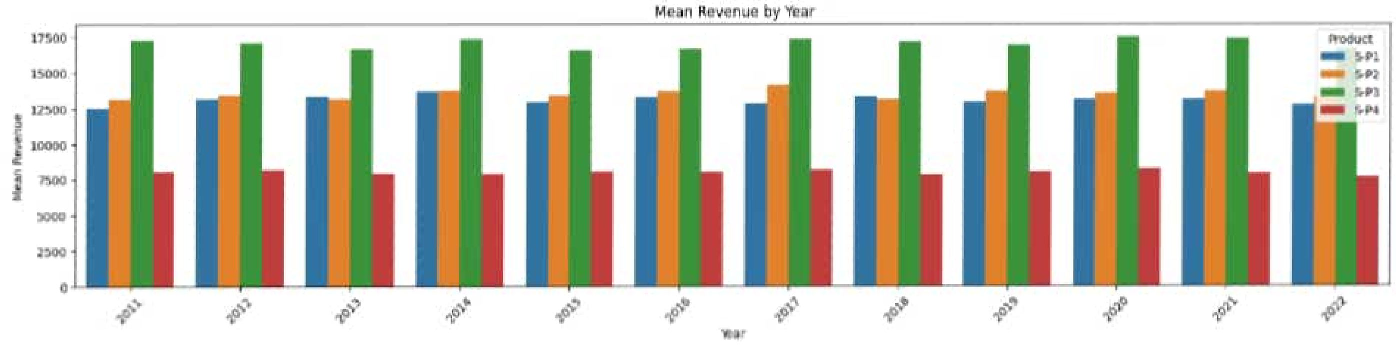


```
[95] Create a function that allows us to plot a bar chart for the 4 products
     ef plot_bar_chart(df, columns, stri, str1, val):
        # Aggregate sales for each product by year, by sum or mean
        if val == 'sum':
            sales_by_year = df.groupby('Year')[columns].sum().reset_index()
        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='Produc

        # Create a bar chart
        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()
```

```
[96] #use the plot_bar_chart function, enter the Unit Sales Columns and the Unit Sales 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')
```

```
[97] #use the plot_bar_chart function, enter the Revenue Columns and the Revenue string
     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')
```





OBSERVATIONS:

- We can observe that P1 has the highest unit sales for each year. And it's highest is in year 2014.
- We can observe taht P4 has the lowest unit sales of all the products. Note

REC corp LTD needs you to solve the following questions:

1) Is there **any trend** in the **sales of all four products** during **certain months?**

"Trend in sales of all four products during certain months"

OBSERVATIONS:

- We can observe that all products drop in Feb. There also appears a very drastic drop after 12th month.
- The value show 9, which must be part of month 09. We need to rename this column to match with the 09. Before doing further analysis.

3) The company has all it's retail centers closed on the 31st of December every year. Mr: Hariharan , the CEO , would love to get an estimate on no: of units of each product that could be sold on 31st of Dec , every year , if all their retail centers were kept open.

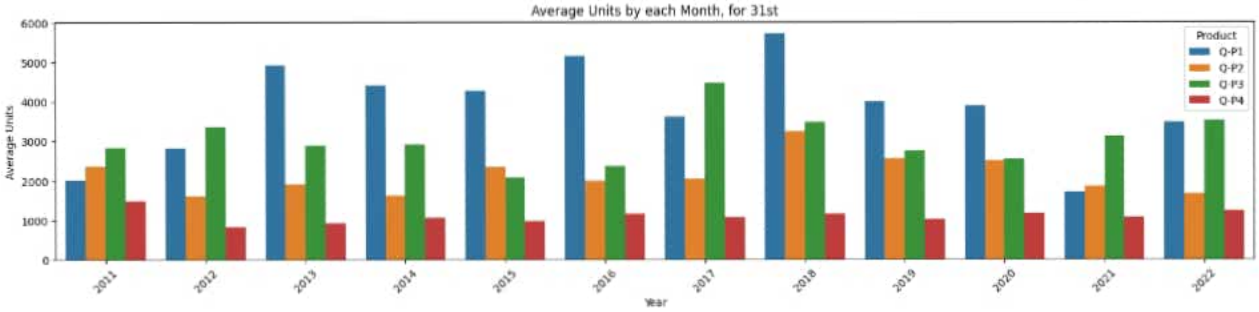
[Estimate for each product the unit of sales that could be sold on 31st of Dec, if all their retail centers were kept open.]

```
[98] #get the 31st day for each month in each year. Note: not every month has 31 days
     def month_31_data(df, months):
         m31_data = df[df['Month'].isin(months) & (df['Day'] == '31')]
         return m31_data

     _31_months = month_31_data(data_reduced, ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10',
     _31_months
```

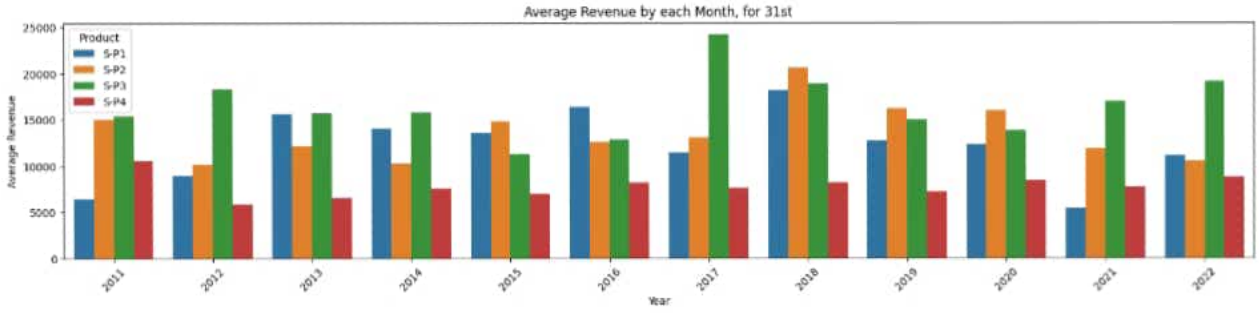
	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4	Day	Month	Year
231	31-01-2011	939	3325	1863	1612	2976.63	21080.50	10097.46	11493.56	31	01	2011
290	31-03-2011	464	2220	421	1663	1470.88	14074.80	2281.82	11857.19	31	03	2011
351	31-05-2011	1507	2980	3816	1202	4777.19	18893.20	20682.72	8570.26	31	05	2011
412	31-07-2011	4336	744	4717	667	13745.12	4716.96	25566.14	4755.71	31	07	2011
442	31-08-2011	4548	1484	1596	1974	14417.16	9408.56	8650.32	14074.62	31	08	2011
...	...	...	...	...	...	...	...	...	...	...	...	...
4291	31-03-2022	3092	1645	4823	1864	9801.64	10429.30	26140.66	13290.32	31	03	2022
4352	31-05-2022	3669	2710	3067	1593	11630.73	17181.40	16623.14	11358.09	31	05	2022
4413	31-07-2022	1437	833	1867	1270	4555.29	5281.22	10119.14	9055.10	31	07	2022
4443	31-08-2022	1035	1639	3658	841	3280.95	10391.26	19826.36	5996.33	31	08	2022
4535	31-11-2022	4600	2006	3796	1426	14582.00	12718.04	20574.32	10167.38	31	11	2022

72 rows × 12 columns

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



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



OBSERVATIONS:

- Overall we can see that P1 has the highest unit sales on the 31st for each year, except for 2021 and 2022. (These could be as a result to Covid and other economy issues.)
- P3 has the second highest unit sales for all the 31st in each year.

5)The CEO would also like to predict the sales and revenues for the year 2024. He wants you to give a yearly estimate with the best possible accuracy.

```
[101] # 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

      # Average for Unit 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
```

```
[102] # Average for Revenue
      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
```

CONCLUSION:

Unit Sales 2011 - 2022

- P1 has the highest unit sales for each year. And it's highest is in year 2014.
- We can observe that P4 has the lowest unit sales of all the products. Revenues 2011 - 2022
- We can observe that P3 brought in the most revenue. This could be as a result of multiple things:
- P3 was sold for higher than the rest, as it had the second highest unit sales for each year.
- We can observe than P1 and P2 brought in similar revenues for each year. With P2 bringing in slightly more.
- P1 despite having the most unit sold, brought in the second lowest revenue each year.

Average Month Sales 2011 - 2022

- We can observe that all Products unit sales drop in Feb.
- We can observe that Feb and Dec have the lowest sales for each product For P1
- We can observe Mar - Jul having the highest unit sales For P2
- We can observe Jan, Mar - Aug having the highest unit sales For P3
- We can observe May & Sep having the highest unit sales For P4
- We can observe uniform sales from Jan - Dec