|  |  |
| --- | --- |
| DATE | 26.10.2023 |
| TEAM ID | 3932 |
| PROJECT NAME | Product Sales Analysys |

[1]:

*# 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")

[5]:

df = pd.read\_csv("C:\Dataset\statsfinal.csv")

[6]:

df.columns

1. : Index(['Unnamed: 0', 'Date', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4', 'S-P1', 'S-P2', 'S-P3', 'S-P4'],

dtype='object')

1. :

print(df)

*# Restore the default display options if needed* pd.reset\_option('display.max\_rows') pd.reset\_option('display.max\_columns')

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

… … … … … … … … …

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4595 | 4595 | 30-01-2023 | 2476 | 3419 | 525 | 1359 | 7848.92 | 21676.46 |
| 4596 | 4596 | 31-01-2023 | 7446 | 841 | 4825 | 1311 | 23603.82 | 5331.94 |
| 4597 | 4597 | 01-02-2023 | 6289 | 3143 | 3588 | 474 | 19936.13 | 19926.62 |
| 4598 | 4598 | 02-02-2023 | 3122 | 1188 | 5899 | 517 | 9896.74 | 7531.92 |
| 4599 | 4599 | 03-02-2023 | 1234 | 3854 | 2321 | 406 | 3911.78 | 24434.36 |

|  |  |  |
| --- | --- | --- |
|  | 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 |
| … | … | … |
| 4595 | 2845.50 | 9689.67 |
| 4596 | 26151.50 | 9347.43 |
| 4597 | 19446.96 | 3379.62 |
| 4598 | 31972.58 | 3686.21 |
| 4599 | 12579.82 | 2894.78 |
| [4600 | rows x 10 | columns] |

1. :
2. :

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 4600 entries, 0 to 4599 Data columns (total 10 columns):

# Column Non-Null Count Dtype

1. Unnamed: 0 4600 non-null int64
2. Date 4600 non-null object
3. Q-P1 4600 non-null int64
4. Q-P2 4600 non-null int64
5. Q-P3 4600 non-null int64
6. Q-P4 4600 non-null int64
7. S-P1 4600 non-null float64
8. S-P2 4600 non-null float64
9. S-P3 4600 non-null float64
10. S-P4 4600 non-null float64 dtypes: float64(4), int64(5), object(1) memory usage: 359.5+ KB

df.dtypes

1. : Unnamed: 0 int64 Date object

Q-P1 int64

Q-P2 int64

Q-P3 int64

Q-P4 int64

S-P1 float64

S-P2 float64

S-P3 float64

S-P4 float64 dtype: object

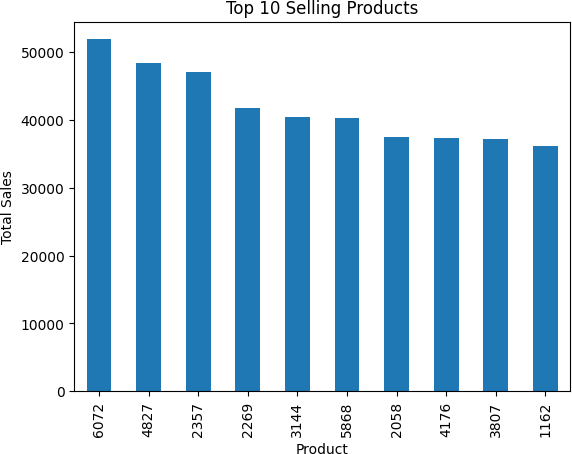
1. :

top\_selling\_products = df.groupby('Q-P1')['S-P4'].sum().

𝗌sort\_values(ascending=**False**)

*# Visualize the top-selling products* top\_selling\_products.head(10).plot(kind='bar') plt.title('Top 10 Selling Products') plt.xlabel('Product')

plt.ylabel('Total Sales') plt.show()



1. :

df.sample(20)

1. : Unnamed: 0 Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 \

|  |  |  |
| --- | --- | --- |
| 4354 | 4354 02-06-2022 3443 3033 1834 1475 | 10914.31 19229.22 |
| 2576 | 2576 13-07-2017 4725 437 4640 534 | 14978.25 2770.58 |
| 3833 | 3833 26-12-2020 1355 963 2433 1615 | 4295.35 6105.42 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3942 | 3942 | 15-04-2021 | 3746 | 3755 | 2969 | 1532 | 11874.82 | 23806.70 |
| 407 | 407 | 26-07-2011 | 2394 | 2624 | 3755 | 301 | 7588.98 | 16636.16 |
| 4569 | 4569 | 04-01-2023 | 5575 | 1684 | 2384 | 1774 | 17672.75 | 10676.56 |
| 1877 | 1877 | 10-08-2015 | 5640 | 1564 | 3036 | 1957 | 17878.80 | 9915.76 |
| 4463 | 4463 | 20-09-2022 | 7951 | 2658 | 5246 | 1989 | 25204.67 | 16851.72 |
| 1961 | 1961 | 03-11-2015 | 2124 | 3304 | 2498 | 1419 | 6733.08 | 20947.36 |
| 1319 | 1319 | 28-01-2014 | 2536 | 2076 | 781 | 1375 | 8039.12 | 13161.84 |
| 3545 | 3545 | 13-03-2020 | 5877 | 3652 | 474 | 1921 | 18630.09 | 23153.68 |
| 1044 | 1044 | 27-04-2013 | 4807 | 1513 | 1447 | 1710 | 15238.19 | 9592.42 |
| 1499 | 1499 | 27-07-2014 | 1949 | 2194 | 5914 | 344 | 6178.33 | 13909.96 |
| 2392 | 2392 | 10-01-2017 | 5264 | 3565 | 4623 | 873 | 16686.88 | 22602.10 |
| 3459 | 3459 | 15-12-2019 | 3220 | 3652 | 4613 | 1206 | 10207.40 | 23153.68 |
| 2134 | 2134 | 26-04-2016 | 7962 | 1879 | 3572 | 634 | 25239.54 | 11912.86 |
| 3971 | 3971 | 14-05-2021 | 4378 | 3820 | 3085 | 713 | 13878.26 | 24218.80 |
| 1435 | 1435 | 24-05-2014 | 2791 | 1572 | 843 | 1478 | 8847.47 | 9966.48 |
| 2540 | 2540 | 07-06-2017 | 2208 | 2736 | 4716 | 1026 | 6999.36 | 17346.24 |
| 900 | 900 | 03-12-2012 | 6466 | 564 | 2436 | 1883 | 20497.22 | 3575.76 |
|  | S-P3 | S-P4 | | | | | | |
| 4354 | 9940.28 | 10516.75 | | | | | | |
| 2576 | 25148.80 | 3807.42 | | | | | | |
| 3833 | 13186.86 | 11514.95 | | | | | | |
| 3942 | 16091.98 | 10923.16 | | | | | | |
| 407 | 20352.10 | 2146.13 | | | | | | |
| 4569 | 12921.28 | 12648.62 | | | | | | |
| 1877 | 16455.12 | 13953.41 | | | | | | |
| 4463 | 28433.32 | 14181.57 | | | | | | |
| 1961 | 13539.16 | 10117.47 | | | | | | |
| 1319 | 4233.02 | 9803.75 | | | | | | |
| 3545 | 2569.08 | 13696.73 | | | | | | |
| 1044 | 7842.74 | 12192.30 | | | | | | |
| 1499 | 32053.88 | 2452.72 | | | | | | |
| 2392 | 25056.66 | 6224.49 | | | | | | |
| 3459 | 25002.46 | 8598.78 | | | | | | |
| 2134 | 19360.24 | 4520.42 | | | | | | |
| 3971 | 16720.70 | 5083.69 | | | | | | |
| 1435 | 4569.06 | 10538.14 | | | | | | |
| 2540 | 25560.72 | 7315.38 | | | | | | |
| 900 | 13203.12 | 13425.79 | | | | | | |

1. :

df.drop(columns=["Unnamed: 0"],inplace=**True**) df.sample()

1. : Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 1724 10-03-2015 3794 2420 631 1313 12026.98 15342.8 3420.02 9361.69
2. :

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

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

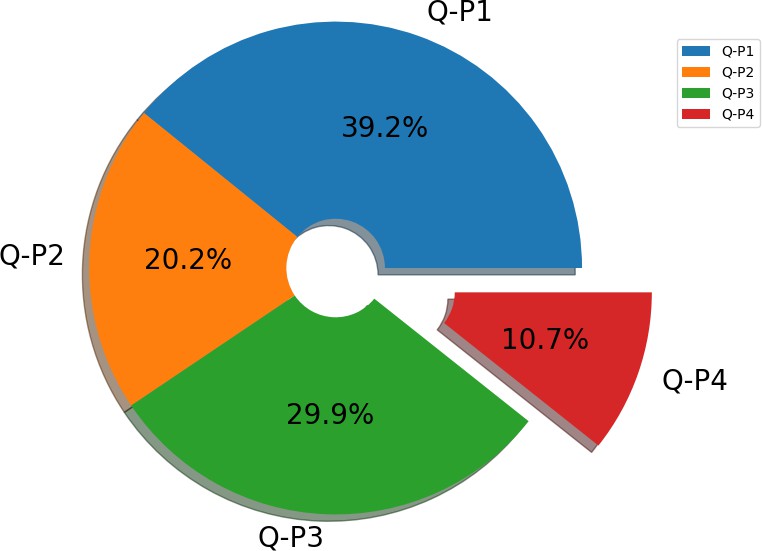
print(q) plt.figure(figsize=(8,8))

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

𝗌index,shadow=**True**,autopct="**%0.01f%%**",textprops={"fontsize":

𝗌20},wedgeprops={'width': 0.8},explode=[0,0,0,0.3]) plt.legend(loc='center right', bbox\_to\_anchor=(1.2, 0.8));

|  |  |
| --- | --- |
| Q-P1 | 18960506 |
| Q-P2 | 9799295 |
| Q-P3 | 14470404 |
| Q-P4 | 5168100 |
| dtype: | int64 |



[15]:

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [15]: |  | 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 |  |
|  | … | … | … … | | … … | … | … |  | … |  |  |  |
| 4595 | | 4595 | 30-01-2023 | | 2476 | 3419 | 525 | 1359 | 7848.92 | | 21676.46 | |
| 4596 | | 4596 | 31-01-2023 | | 7446 | 841 | 4825 | 1311 | 23603.82 | | 5331.94 | |
| 4597 | | 4597 | 01-02-2023 | | 6289 | 3143 | 3588 | 474 | 19936.13 | | 19926.62 | |
| 4598 | | 4598 | 02-02-2023 | | 3122 | 1188 | 5899 | 517 | 9896.74 | | 7531.92 | |
| 4599 | | 4599 | 03-02-2023 | | 1234 | 3854 | 2321 | 406 | 3911.78 | | 24434.36 | |
|  |  | S-P3 | S-P4 | Day | Month | Year | | | | | | |
|  | 0 | 3121.92 | 6466.91 | 13 | 06 | 2010 | | | | | | |
|  | 1 | 19392.76 | 11222.62 | 14 | 06 | 2010 | | | | | | |
|  | 2 | 3224.90 | 8163.85 | 15 | 06 | 2010 | | | | | | |
|  | 3 | 17018.80 | 11921.36 | 16 | 06 | 2010 | | | | | | |
|  | 4  …  4595 | 11837.28  …  2845.50 | 5048.04  … ..  9689.67 | 17  …  30 | 06  …  01 | 2010  2023 | | | | | | |
|  | 4596 | 26151.50 | 9347.43 | 31 | 01 | 2023 | | | | | | |
|  | 4597 | 19446.96 | 3379.62 | 01 | 02 | 2023 | | | | | | |
|  | 4598 | 31972.58 | 3686.21 | 02 | 02 | 2023 | | | | | | |
|  | 4599 | 12579.82 | 2894.78 | 03 | 02 | 2023 | | | | | | |
|  | [4600 | rows x 13 | columns] |  |  |  | | | | | | |
| [19]: |  |  |  |  |  |  | | | | | | |

data\_reduced = data.query("Year != '2010' and Year != '2023'")

**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',␣

𝗌value\_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')␣

𝗌*#,palette="cividis")*

plt.xlabel('Year') plt.ylabel(stri) plt.title(f'**{**stri**}** by **{**str1**}**')

plt.xticks(rotation=45) plt.show()

[20]:

*#use the plot\_bar\_chart function, enter the Unit Sales Columns and the Unit*␣

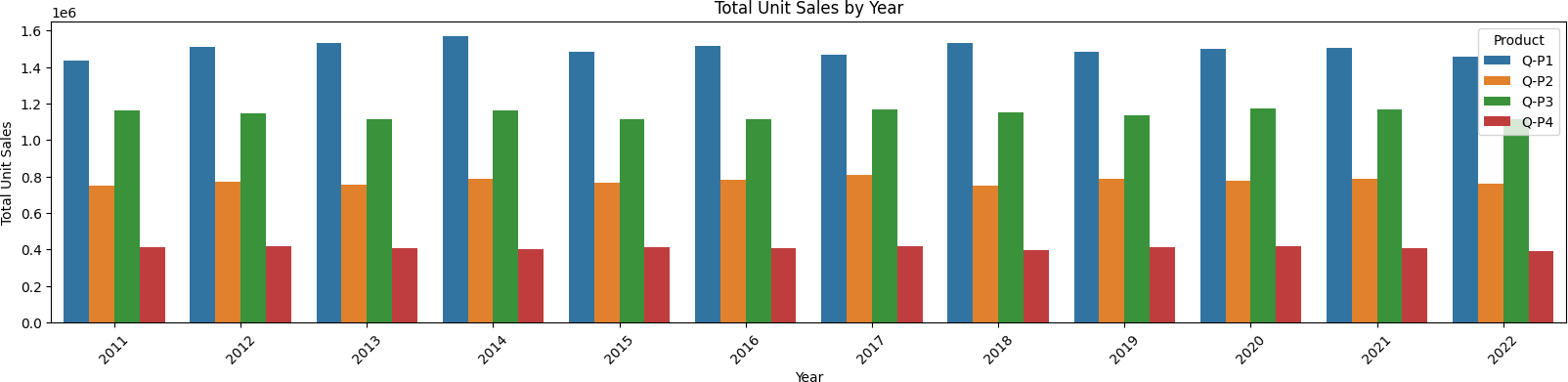
𝗌*Sales string*

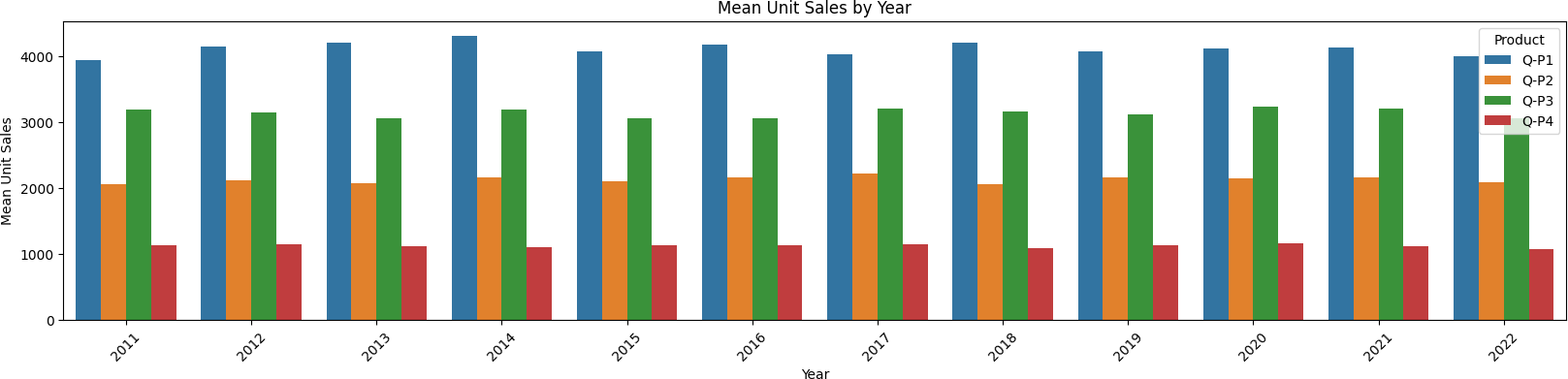
plot\_bar\_chart(data\_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Total Unit␣

𝗌Sales', 'Year', 'sum')

plot\_bar\_chart(data\_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Mean Unit␣

𝗌Sales', 'Year', 'mean')





[21]:

*#use the plot\_bar\_chart function, enter the Revenue Columns and the Revenue*␣

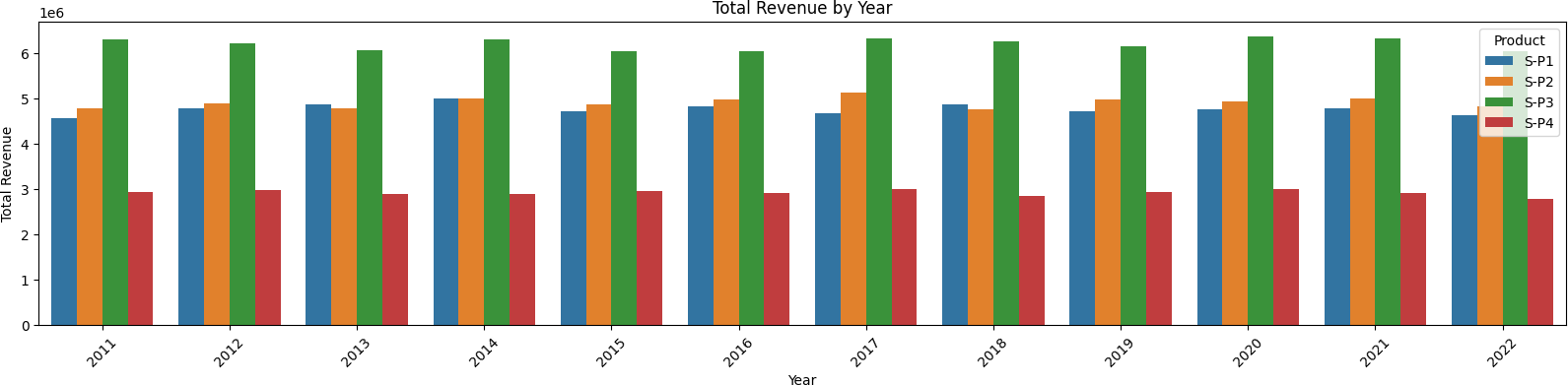
𝗌*string*

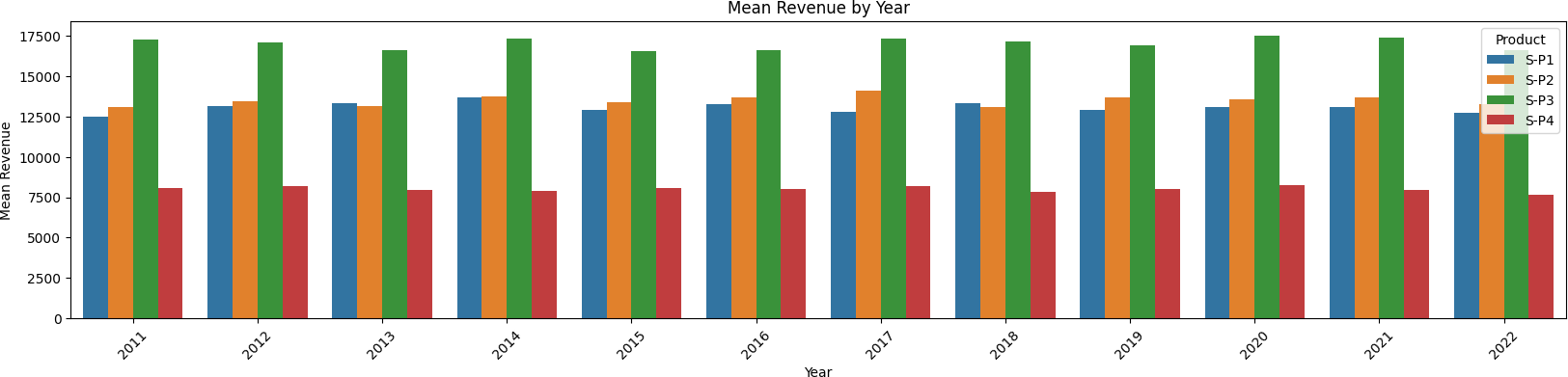
plot\_bar\_chart(data\_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Total Revenue',␣

𝗌'Year', 'sum')

plot\_bar\_chart(data\_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Mean Revenue',␣

𝗌'Year', 'mean')





[22]:

**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().

𝗌plot(ax=ax)

*# Set the x-axis limits to only show up to December*

ax.set\_xlim(left=0, right=13)

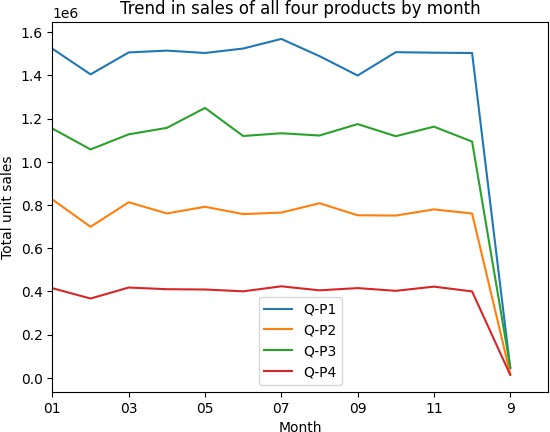
*# Set the axis labels and title* ax.set\_xlabel('Month') ax.set\_ylabel('Total unit sales')

ax.set\_title('Trend in sales of all four products by month')

*# Show the plot*

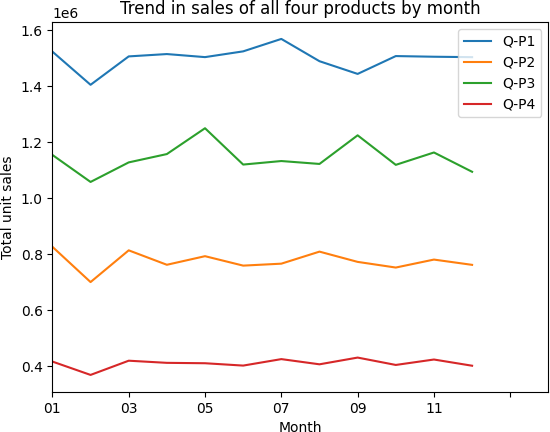
plt.show()

month\_plot()



[23]:

data\_reduced['Month'] = data['Month'].replace('9', '09') month\_plot()



[25]:

**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',␣

𝗌'07', '08', '09', '10', '11', '12'])

[26]:

\_31\_months

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [26]: |  | Unnamed: 0 | Date | Q-P1 | Q-P2 | Q-P3 | Q-P4 |  | S-P1 | S-P2 | \ |
|  | 231 | 231 | 31-01-2011 | 939 | 3325 | 1863 | 1612 |  | 2976.63 | 21080.50 |  |
|  | 290 | 290 | 31-03-2011 | 464 | 2220 | 421 | 1663 |  | 1470.88 | 14074.80 |  |
|  | 351 | 351 | 31-05-2011 | 1507 | 2980 | 3816 | 1202 |  | 4777.19 | 18893.20 |  |
|  | 412 | 412 | 31-07-2011 | 4336 | 744 | 4717 | 667 |  | 13745.12 | 4716.96 |  |
|  | 442 | 442 | 31-08-2011 | 4548 | 1484 | 1596 | 1974 |  | 14417.16 | 9408.56 |  |
|  | … | … | … … | … … | … | … |  | … |  |  |  |
| 4352 | | 4352 | 31-05-2022 | 3669 | 2710 | 3067 | 1593 | 11630.73 | | 17181.40 | |
| 4413 | | 4413 | 31-07-2022 | 1437 | 833 | 1867 | 1270 | 4555.29 | | 5281.22 | |
| 4443 | | 4443 | 31-08-2022 | 1035 | 1639 | 3658 | 841 | 3280.95 | | 10391.26 | |
| 4474 | | 4474 | 31-9-2022 | 6964 | 1873 | 5481 | 1336 | 22075.88 | | 11874.82 | |

4535 4535 31-11-2022 4600 2006 3796 1426 14582.00 12718.04

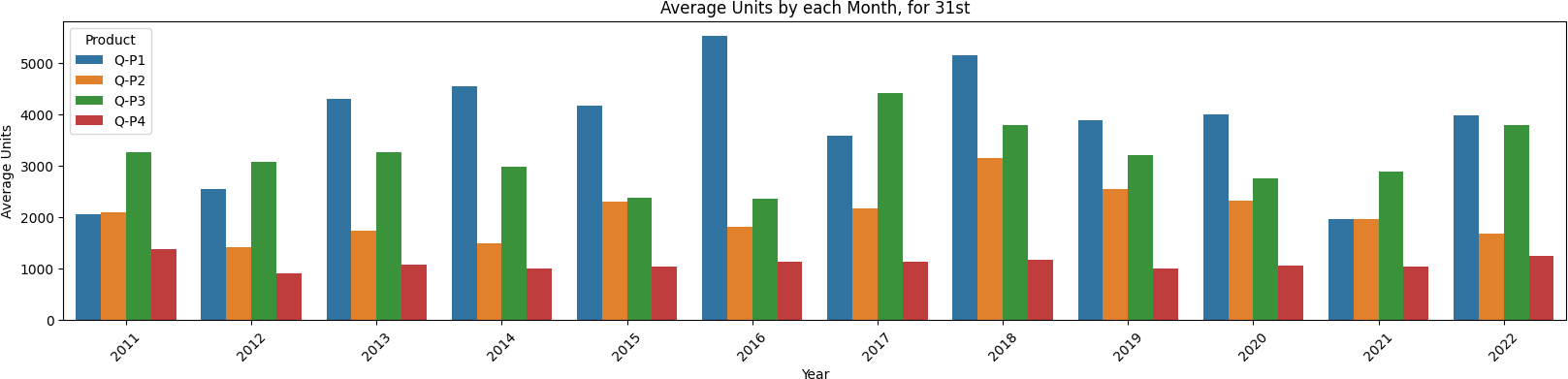
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | S-P3 | S-P4 | Day | Month | Year |
| 231 | 10097.46 | 11493.56 | 31 | 01 | 2011 |
| 290 | 2281.82 | 11857.19 | 31 | 03 | 2011 |
| 351 | 20682.72 | 8570.26 | 31 | 05 | 2011 |
| 412 | 25566.14 | 4755.71 | 31 | 07 | 2011 |
| 442 | 8650.32 | 14074.62 | 31 | 08 | 2011 |
| … | … | … .. | … | … |  |
| 4352 | 16623.14 | 11358.09 | 31 | 05 | 2022 |
| 4413 | 10119.14 | 9055.10 | 31 | 07 | 2022 |
| 4443 | 19826.36 | 5996.33 | 31 | 08 | 2022 |
| 4474 | 29707.02 | 9525.68 | 31 | 09 | 2022 |
| 4535 | 20574.32 | 10167.38 | 31 | 11 | 2022 |

[84 rows x 13 columns]

[27]:

plot\_bar\_chart(\_31\_months, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Average Units',␣

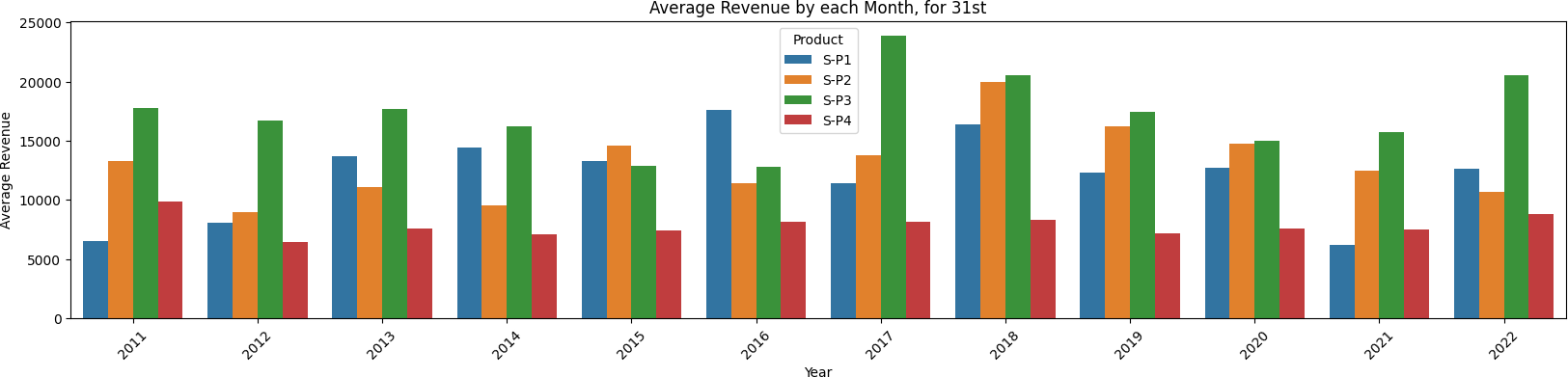
𝗌'each Month, for 31st', 'mean')



[28]:

plot\_bar\_chart(\_31\_months, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Average Revenue',␣

𝗌'each Month, for 31st', 'mean')



[29]:

df[df['Date'].isnull()]

[29]: Empty DataFrame

Columns: [Date, Q-P1, Q-P2, Q-P3, Q-P4, S-P1, S-P2, S-P3, S-P4]

Index: []

[32]:

**for** i **in** df.columns:

print(i," ",df[i].unique())

[36]:

df.sample()

Date --------- ['13-06-2010' '14-06-2010' '15-06-2010' … '01-02-2023'

'02-02-2023'

'03-02-2023']

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q-P1 --------- | [5422 7047 | 1572 | … | 1227 3122 | 1234] | | |
| Q-P2 --------- | [3725 779 | 2082 | … | 3404 841 | 3143] | | |
| Q-P3 --------- | [ 576 3578 | 595 | … | 4825 3588 | 5899] | | |
| Q-P4 --------- | [ 907 1574 | 1145 | … | 1161 1151 | 1112] | | |
| S-P1 --------- | [17187.74 22338.99 4983.24 | | | | … 3889.59 | 9896.74 | 3911.78] |
| S-P2 --------- | [23616.5 4938.86 13199.88 | | | | … 21581.36 | 5331.94 | 19926.62] |
| S-P3 --------- | [ 3121.92 19392.76 3224.9 | | | | … 26151.5 | 19446.96 | 31972.58] |
| S-P4 --------- | [ 6466.91 11222.62 8163.85 | | | | … 8277.93 | 8206.63 | 7928.56] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [36]: | Date Q-P1 Q-P2 Q-P3 Q-P4 | S-P1 | S-P2 | S-P3 | \ |
|  | 2296 05-10-2016 4718 3983 5502 739 | 14956.06 | 25252.22 | 29820.84 |  |

[37]:

*# 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))

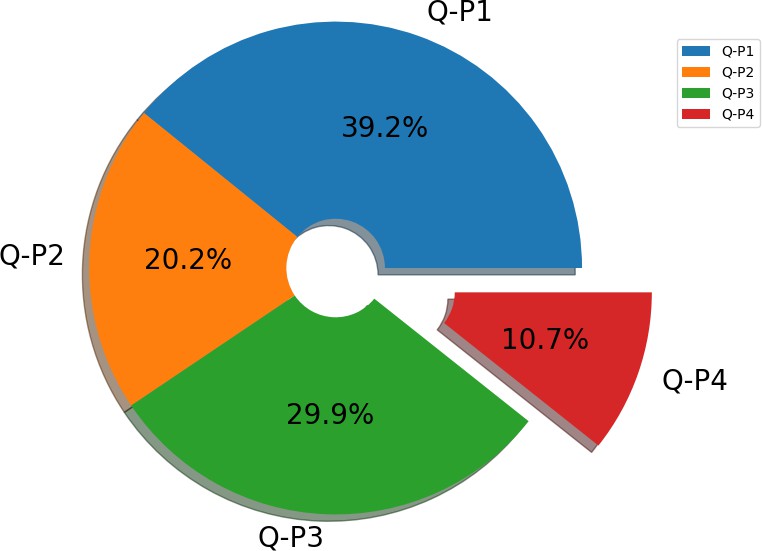
plt.pie(q,labels=df[["Q-P1","Q-P2","Q-P3","Q-P4"]].sum().

𝗌index,shadow=**True**,autopct="**%0.01f%%**",textprops={"fontsize":

𝗌20},wedgeprops={'width': 0.8},explode=[0,0,0,0.3]) plt.legend(loc='center right', bbox\_to\_anchor=(1.2, 0.8));

S-P4 2296 5269.07

|  |  |
| --- | --- |
| Q-P1 | 18960506 |
| Q-P2 | 9799295 |
| Q-P3 | 14470404 |
| Q-P4 | 5168100 |
| dtype: | int64 |



[40]:

**if** "month" **in** df.columns: print(df["month"].value\_counts()) plt.figure(figsize=(10, 10)) sns.countplot(x="month", data=df, edgecolor="black")

**else**:

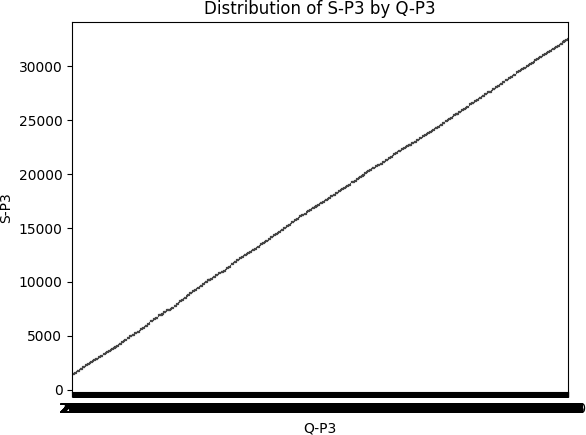
print("The column 'month' does not exist in the DataFrame.")

The column 'month' does not exist in the DataFrame.

[43]:

sns.boxplot(x='Q-P3', y='S-P3', data=df) plt.xlabel('Q-P3')

plt.ylabel('S-P3') plt.title('Distribution of S-P3 by Q-P3') plt.show()



[45]:

data = df

**try**:

df['Date'] = pd.to\_datetime(df['Date'], format='**%d**-%m-%Y', errors='coerce')

**except** pd.errors.ParserError:

**pass**

df = df.dropna(subset=['Date'])

df['Date'] = pd.to\_datetime(df['Date'], format='**%d**-%m-%Y') df.set\_index('Date', inplace=**True**)

products = ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']

**for** product **in** products: product\_data = df[product]

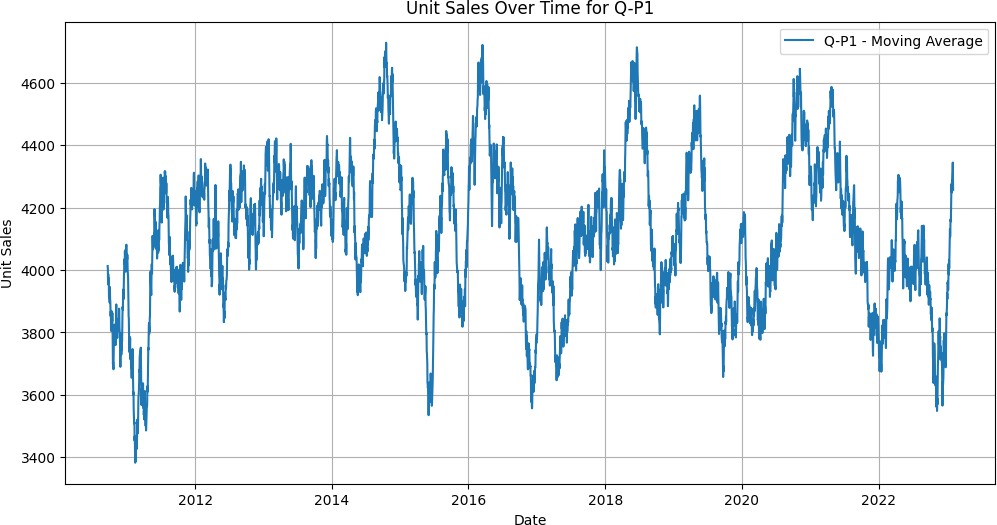
moving\_average = product\_data.rolling(window=100).mean() plt.figure(figsize=(12, 6))

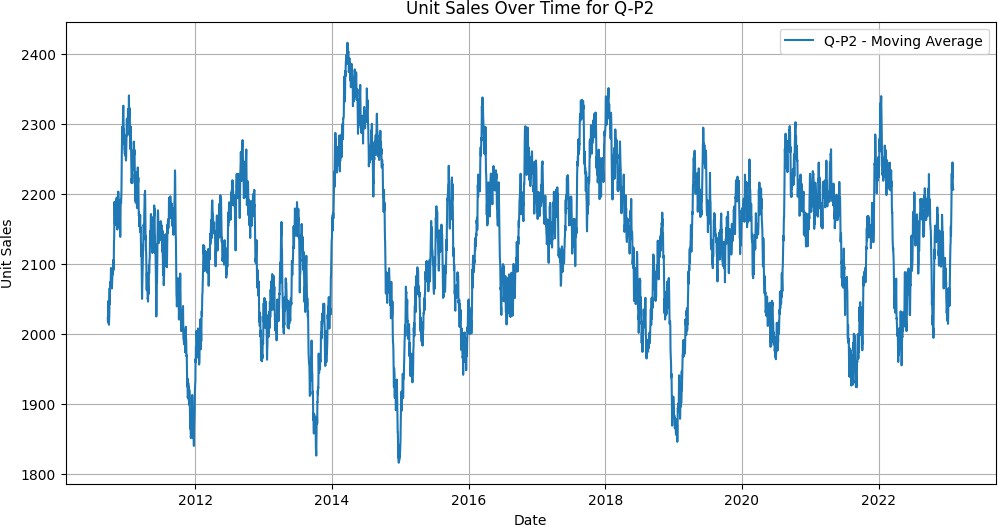
plt.plot(df.index, moving\_average, label=f'**{**product**}** - Moving Average') plt.xlabel('Date')

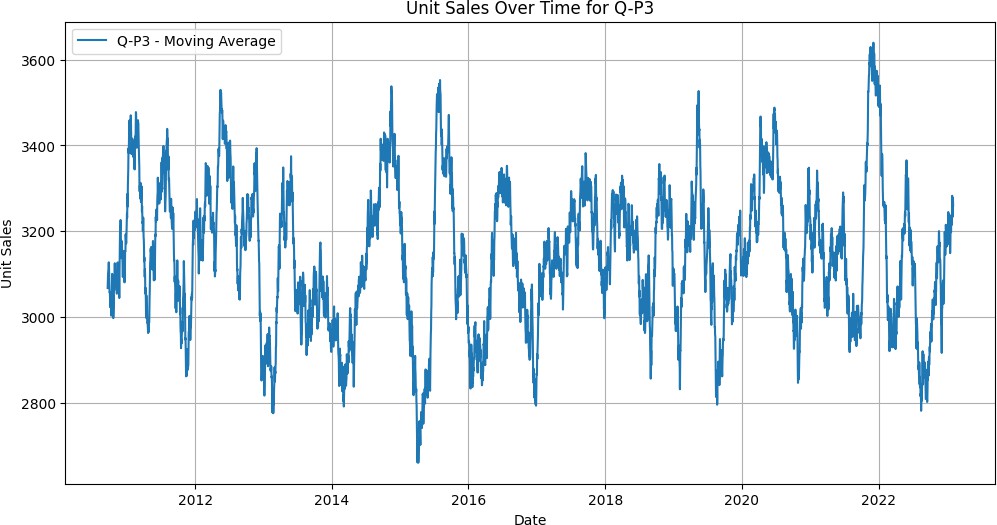
plt.ylabel('Unit Sales')

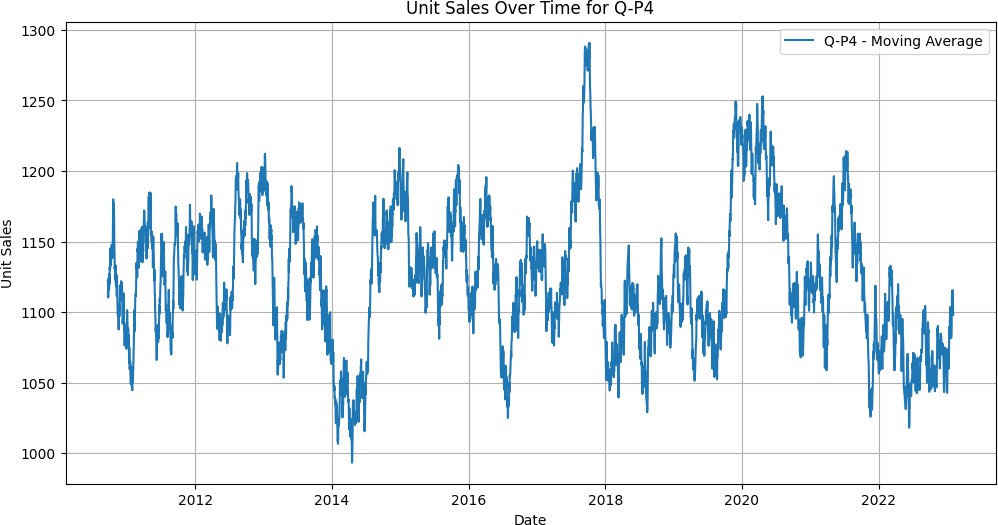
plt.title(f'Unit Sales Over Time for **{**product**}**') plt.grid(**True**)

plt.legend() plt.show()









[4

data = df

**if** 'Date' **in** df.columns:

**try**:

df['Date'] = pd.to\_datetime(df['Date'], format='**%d**-%m-%Y',␣

𝗌errors='coerce')

**except** pd.errors.ParserError:

**pass**