

# PRODUCT SALES ANALYSIS

PHASE4: **Development Part 2**

SUBMITTED BY

NAME: Thammineni Vamsi krishna

NM ID: au723921244048

MAIL ID: tvamsikrishnanaidutvamsi@gmail.com

# Product Sales Analysis

## Phase-4 Document Submission

**Project:** Product Sales Analysis

**Phase 3: Development Part 2**

**Topic:** The Development Part 2, is a crucial stage in your project where you Continue building the analysis by creating visualizations using IBM Cognos and generating actionable insights from the visualizations, such as identifying products with the highest sales, peak sales periods, and customer preferences for specific products.

### Product Sales Analysis



**40**

**Product A**

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.



**650**

**Product B**

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.



**380**

**Product C**

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.



**55**

**Product D**

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

## Introduction:

In an ever-evolving business landscape, understanding sales patterns, consumer behavior, and product performance is paramount to the sustained growth and success of a company. Product sales analysis is the critical tool that equips organizations with the insights needed to navigate this landscape.

## Given dataset:

|      | 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  |

## Building the analysis by creating visualizations using IBM Cognos and generating actionable insights.

### Connect to Data Sources:

Launch IBM Cognos and connect it to your data sources containing product sales data.

### Create a Data Module:

Define a data module to organize and prepare your data for analysis. Select relevant tables, join data, and create calculated fields if necessary.

### Design Visualizations:

Create visualizations that address specific aspects of product sales analysis:

- **Top-Selling Products:** Create a bar chart or a list that displays the top-selling products based on sales revenue.
- **Sales Trends:** Develop a line chart or area chart to visualize sales trends over time. Break it down by month, quarter, or year.
- **Customer Preferences:** Use filters to allow users to select customer preferences (e.g., region, product category) and link them to relevant visualizations.

### **Interactive Dashboards:**

Design a dashboard and add the visualizations you created. Arrange them logically for easy consumption.

### **Add Interactivity:**

Implement interactive elements like drop-down menus, sliders, and date pickers to enable users to dynamically explore the data.

### **Apply Contextual Filters:**

Apply filters that allow users to refine the data displayed in real-time. For example, let users select a specific time frame or product category.

### **Incorporate Drill-Downs:**

Enable users to drill down into specific data points for more detailed insights. For instance, from a regional view to a country-level view.

### **Utilize Conditional Formatting:**

Apply color coding and formatting to highlight important information and trends in the visualizations.

### **Add Tooltips and Labels:**

Provide additional context by adding tooltips and labels to your visualizations. This helps users understand the data points.

### **Set Up Alerts:**

Configure alerts to notify users when specific thresholds or conditions are met (e.g., sales exceeding a target).

### Apply Statistical Analysis:

Incorporate statistical functions to identify significant trends, correlations, or outliers in the data.

### Generate Reports:

Create detailed reports for stakeholders who may prefer a more structured view of the data.

### Schedule and Distribute Reports:

Set up schedules to automatically generate and distribute reports to relevant stakeholders via email or a shared platform.

### Monitor and Analyze Usage:

Keep track of how users interact with the dashboards and reports. Use this feedback to make improvements or updates.

### Derive Actionable Insights:

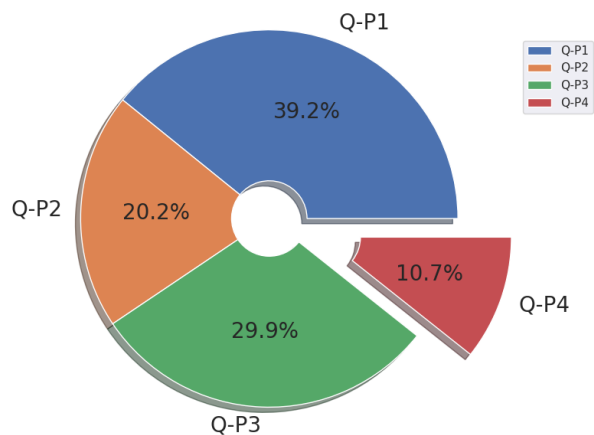
Analyze the visualizations and reports to draw conclusions and recommendations. Identify areas for improvement, marketing strategies, and product development opportunities.

Remember to engage with end-users or stakeholders to gather feedback and refine the analysis and dashboards over time. This iterative process ensures that the insights provided are relevant and valuable for decision-making.

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

```
# 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));
```

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



## Total Revenue percent from sales from Product 1, Product 2, Product 3, Product 4

*# Total Revenue percent from sales from Product 1, Product 2, Product 3, Product 4*

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

```
print(s)
```

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

```
plt.pie(s,labels=df[["S-P1","S-P2","S-P3","S-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))
```

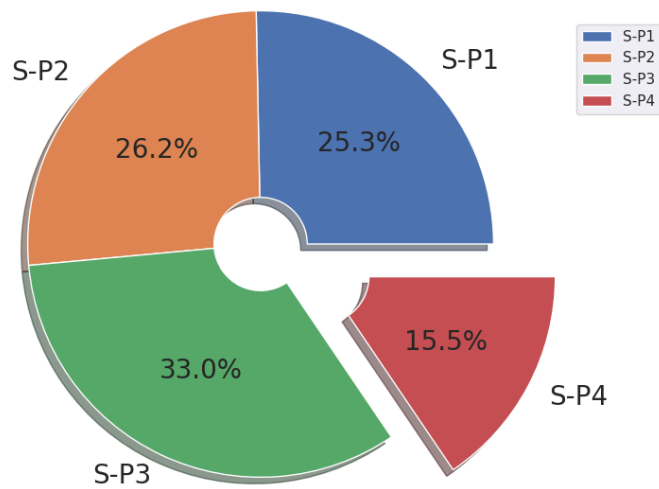
```
S-P1    60104804.02
```

```
S-P2    62127530.30
```

```
S-P3    78429589.68
```

```
S-P4    36848553.00
```

```
dtype: float64
```



## Most sales occuring month :

# which is the most occuring month

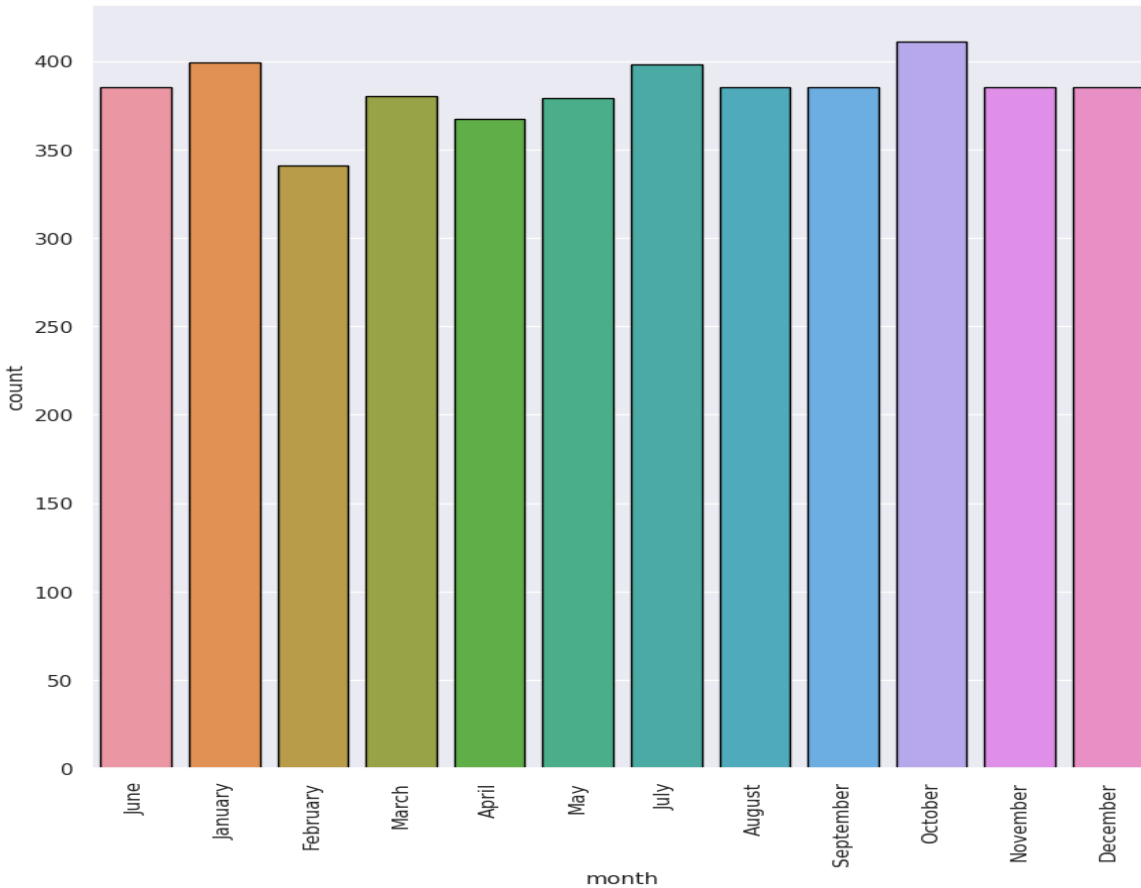
```
print(df["month"].value_counts())
```

```
plt.figure(figsize=(10,10))
```

```
sns.countplot(x="month",data=df,edgecolor="black")
```

```
plt.xticks(rotation=90);
```

```
October    411
January    399
July       398
June       385
August     385
September  385
November   385
December   385
March      380
May        379
April      367
February   341
Name: month, dtype: int64
```



## Most sales occuring Day

# which is the most occuring Day

```
print(df["day"].value_counts())
```

```
plt.figure(figsize=(10,10))
```

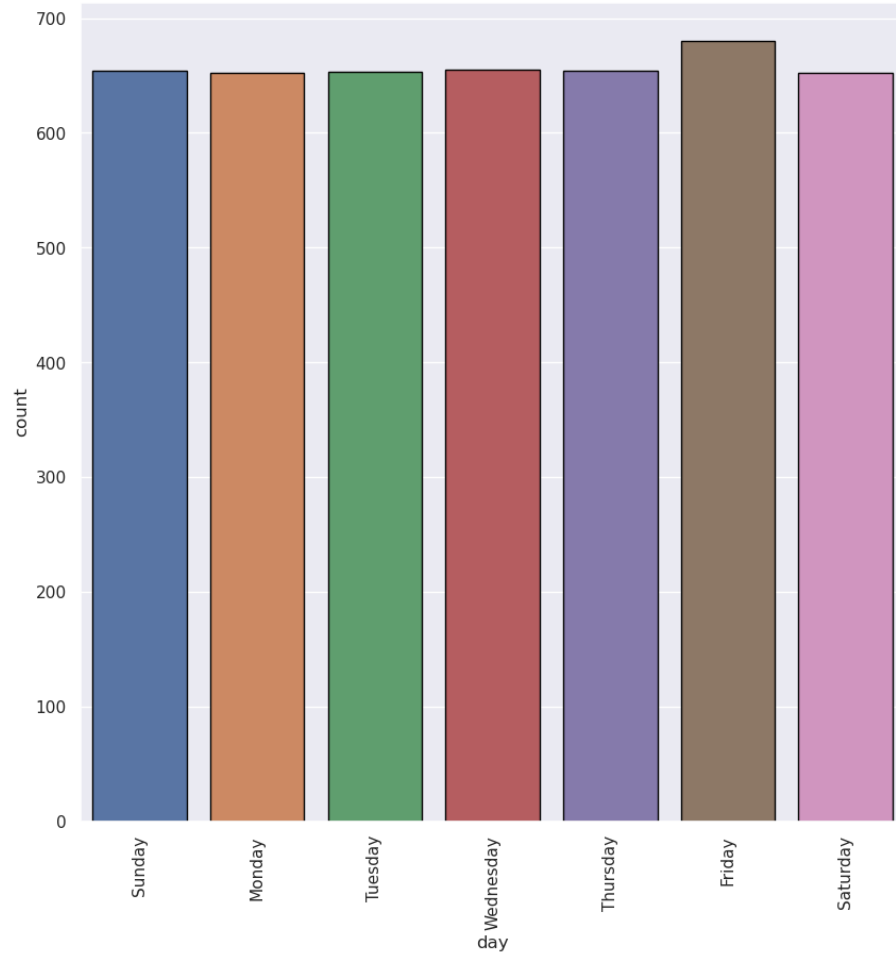
```
sns.countplot(x="day",data=df,edgecolor="black")
```

```
plt.xticks(rotation=90);
```

```

Friday      680
Wednesday   655
Sunday      654
Thursday    654
Tuesday     653
Monday      652
Saturday    652
Name: day, dtype: int64

```



### Most sales occuring year:

*# which is the most occuring year*

```

print(df["year"].value_counts())
plt.figure(figsize=(10,10))
sns.countplot(x="year",data=df,edgecolor="black")
plt.xticks(rotation=90);

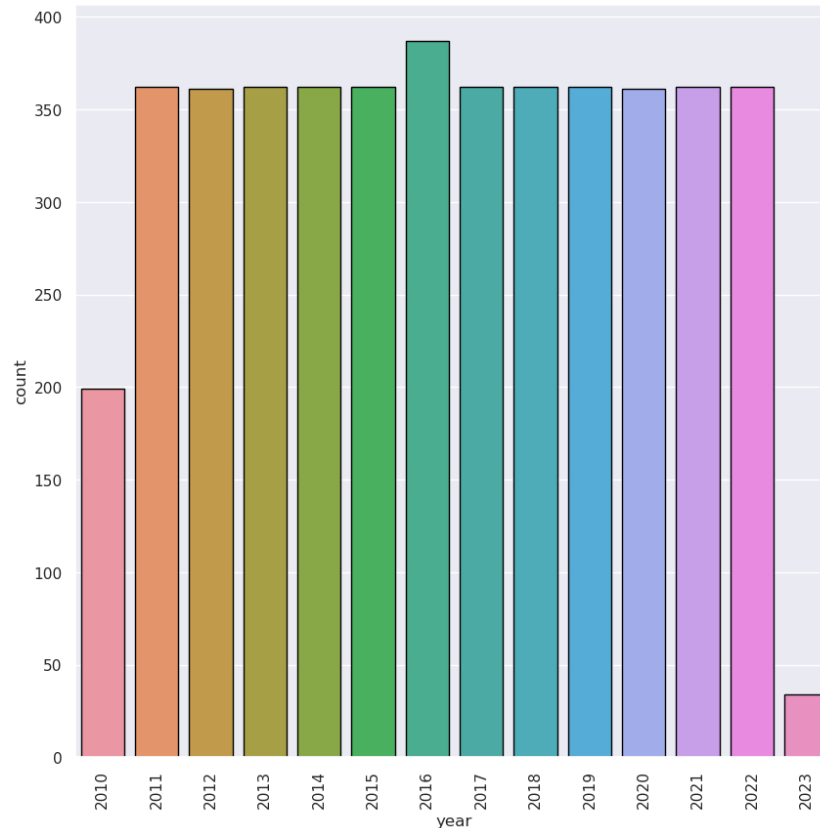
```

```

2016      387
2011      362
2013      362
2014      362
2015      362
2017      362
2018      362
2019      362
2021      362
2022      362
2012      361
2020      361
2010      199
2023       34
Name: year, dtype: int64

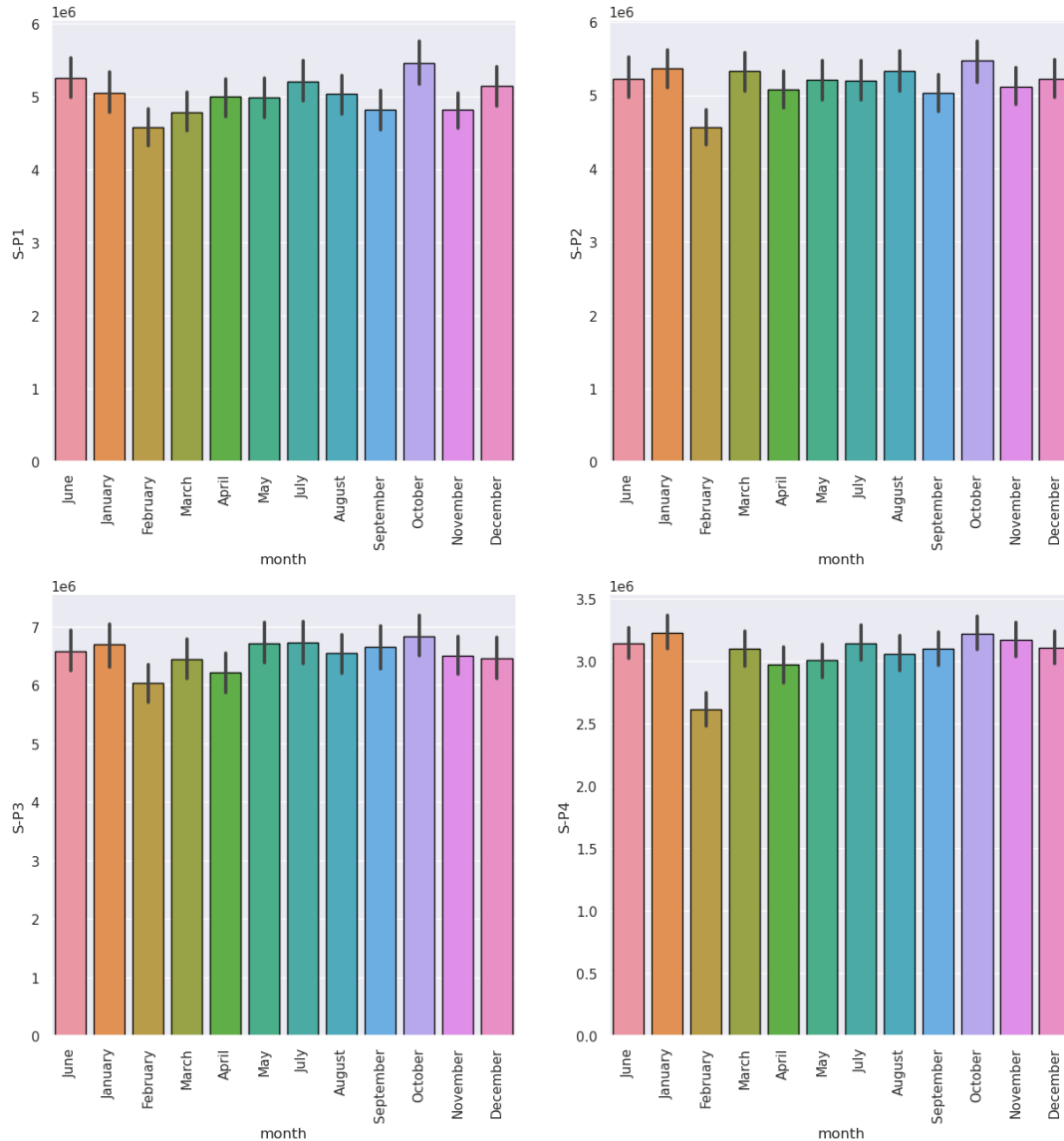
```





### Peak sales in a month:

```
# In which month revenue was it peak
df.groupby("month")[["S-P1", "S-P2", "S-P3", "S-P4"]].sum()
plt.figure(figsize=(15,15),dpi=100)
plt.subplot(2,2,1)
sns.barplot(x="month",y="S-P1",data=df,edgecolor="black",estimator=sum)
plt.xticks(rotation=90);
plt.subplot(2,2,2)
sns.barplot(x="month",y="S-P2",data=df,edgecolor="black",estimator=sum)
plt.xticks(rotation=90);
plt.subplot(2,2,3)
sns.barplot(x="month",y="S-P3",data=df,edgecolor="black",estimator=sum)
plt.xticks(rotation=90);
plt.subplot(2,2,4)
sns.barplot(x="month",y="S-P4",data=df,edgecolor="black",estimator=sum)
plt.xticks(rotation=90)
plt.subplots_adjust(hspace=0.3);
```



## Sales per week:

```
week_t=df[df["dayoftheweek"]<5]
weekend_t=df[df["dayoftheweek"]>=5]
print(week_t.groupby("day")[["S-P1","S-P2","S-P3","S-P4"]].sum())
```

| day       | S-P1       | S-P2       | S-P3        | S-P4       |
|-----------|------------|------------|-------------|------------|
| Friday    | 8913637.41 | 9267831.02 | 11428877.58 | 5463169.99 |
| Monday    | 8636791.80 | 8864347.08 | 11064892.06 | 5292577.61 |
| Thursday  | 8577981.96 | 8909481.54 | 10951554.44 | 5043013.35 |
| Tuesday   | 8433525.06 | 8738326.90 | 11156338.30 | 5384854.07 |
| Wednesday | 8693537.97 | 8908067.72 | 11017830.20 | 5086827.20 |

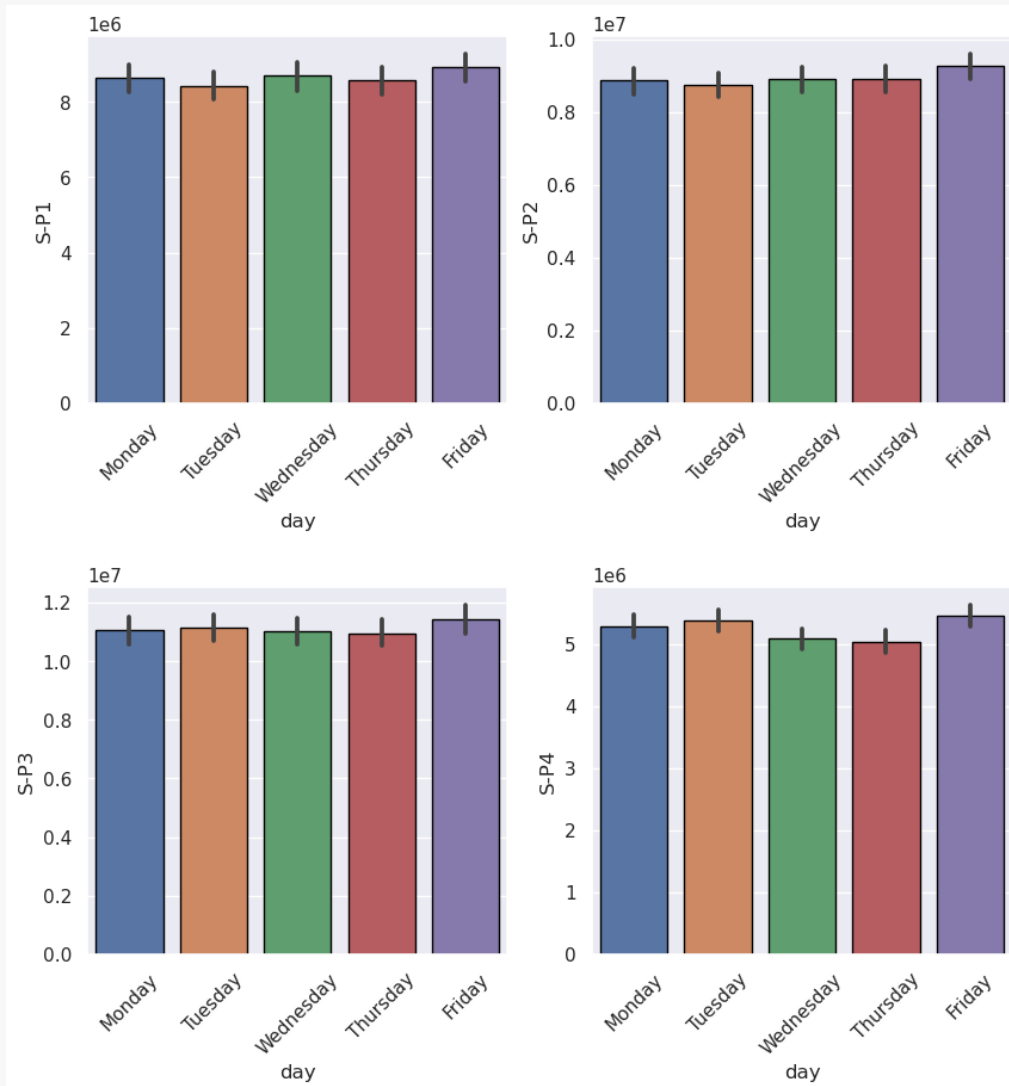
```
plt.figure(figsize=(10,10),dpi=100)
plt.subplot(2,2,1)
sns.barplot(x="day",y="S-P1",data=week_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,2)
```

```

sns.barplot(x="day",y="S-P2",data=week_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,3)
sns.barplot(x="day",y="S-P3",data=week_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,4)

sns.barplot(x="day",y="S-P4",data=week_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45)
plt.subplots_adjust(hspace=0.5);

```



### Sales at Weekend:

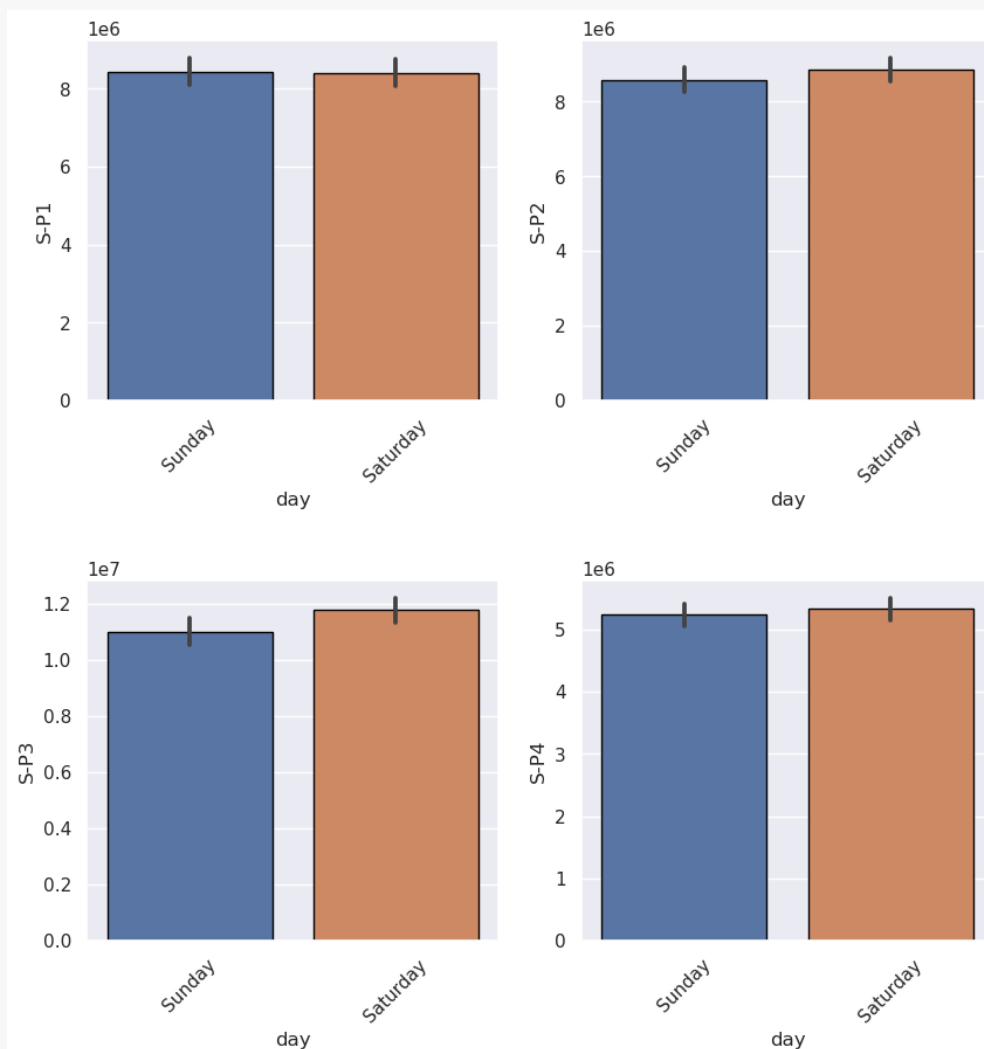
```
print(weekend_t.groupby("day")[["S-P1","S-P2","S-P3","S-P4"]].sum())
```

|          | S - P1     | S - P2     | S - P3      | S - P4     |
|----------|------------|------------|-------------|------------|
| day      |            |            |             |            |
| Saturday | 8409578.88 | 8853201.36 | 11796375.26 | 5339977.85 |
| Sunday   | 8439750.94 | 8586274.68 | 11013721.84 | 5238132.93 |

```

plt.figure(figsize=(10,10),dpi=100)
plt.subplot(2,2,1)
sns.barplot(x="day",y="S-P1",data=weekend_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,2)
sns.barplot(x="day",y="S-P2",data=weekend_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,3)
sns.barplot(x="day",y="S-P3",data=weekend_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45);
plt.subplot(2,2,4)
sns.barplot(x="day",y="S-P4",data=weekend_t,edgecolor="black",estimator=sum)
plt.xticks(rotation=45)
plt.subplots_adjust(hspace=0.5);

```



## **Conclusion:**

The product sales analysis has yielded valuable insights into our business's performance and customer behavior. Through a detailed examination of sales data, we have uncovered several key findings such as Top-Selling Products, Sales Trends, Customer Preferences, Profitability Analysis and so on....