PRODUCT SALES ANALYSIS

PHASE 3 : VISUALIZATION & PREPROCESSING

INTRODUCTION

Phase 3 of the Product Sales Analysis project focuses on visualizing and defining the analysis objectives and collect sales data from source shared and also process and clean the collected data to ensure its accuracy and reliability. The primary objective is to compare and contrast the trends , places , sales and profit based on cases and associated sales per day and by to develop sales trends based on consumer interest to make profit. This project aims to gain insights into the Product Sales situation, identify trends, and provide data-driven information to aid decision-makers in understanding the Sales process

OBJECTIVES

1. Define Analysis Objectives:

The first step in any data analysis endeavor is to clearly outline your objectives. What specific questions or challenges are you aiming to address through the analysis of sales data? Whether it's understanding customer behavior, optimizing pricing strategies, or evaluating the impact of marketing campaigns, a well-defined objective sets the stage for a focused and effective analysis.

2. Data Collection:

Identify the sources from which you will collect sales data. These sources may include sales databases, CRM systems, point-of-sale (POS) terminals, e-commerce platforms, or any other relevant systems where sales data is recorded.

3. Data Collection Methods:

Choose the most appropriate methods for data collection. This may involve extracting data from databases, using APIs to retrieve information from online platforms, or manual data entry. Ensure that data is collected consistently and comprehensively.

4. Data Cleaning:

Cleaning the collected data is a crucial step to ensure its accuracy and reliability. Common data cleaning tasks include:

- Removing duplicate entries.
- Correcting errors and inconsistencies in data entry.
- Handling missing values (e.g., using imputation methods).
- Standardizing data formats (e.g., date formats, product names, and customer information).
- Validating data against predefined criteria.
- Removing outliers that could skew the analysis.
-

5. Data Transformation:

Depending on your analysis objectives, you may need to transform the data. This could involve aggregating data to a different time granularity (e.g., monthly or quarterly), calculating key performance metrics, or creating new variables that are relevant to your analysis.

6. Data Integration:

If your sales data comes from multiple sources, you may need to integrate it into a single dataset. Ensure that the data is properly linked and that there are no conflicts or discrepancies between sources.

7. Data Validation:

Conduct data validation to ensure that the processed data aligns with your objectives. Cross-check the data for any irregularities or anomalies that may have been missed during cleaning and transformation.

8. Data Quality Assurance:

Implement quality assurance measures to maintain data accuracy and reliability over time. This may involve regular monitoring, setting up validation rules, and ensuring data consistency across all data sources.

9. Documentation:

Maintain clear and thorough documentation of the data processing steps and decisions made during the cleaning and transformation process. This documentation will be valuable for future reference and for replicating the analysis.

10. Analysis and Reporting:

With clean and reliable data in hand, proceed with the analysis based on your defined objectives. This may involve using statistical methods, data visualization, or machine learning techniques to derive insights and answer your questions.

PROCESS

Process and clean the collected data to ensure its accuracy and reliability .We have,

In [5]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

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

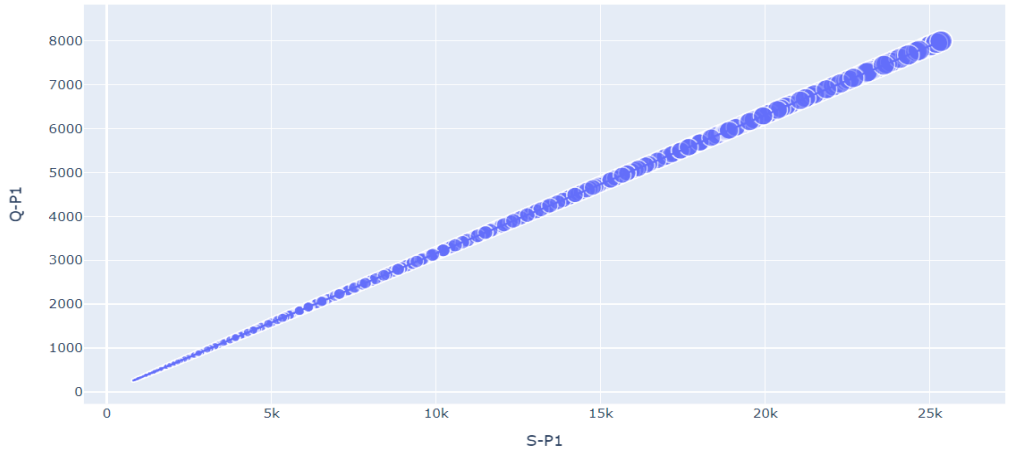
```
   Unnamed: 0        Date  Q-P1  Q-P2  Q-P3  Q-P4      S-P1      S-P2  \
0           0  13-06-2010  5422  3725   576   907  17187.74  23616.50
1           1  14-06-2010  7047   779  3578  1574  22338.99   4938.86
2           2  15-06-2010  1572  2082   595  1145   4983.24  13199.88
3           3  16-06-2010  5657  2399  3140  1672  17932.69  15209.66
4           4  17-06-2010  3668  3207  2184   708  11627.56  20332.38

       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
```

In [6]:
```
print(data.isnull().sum())
```

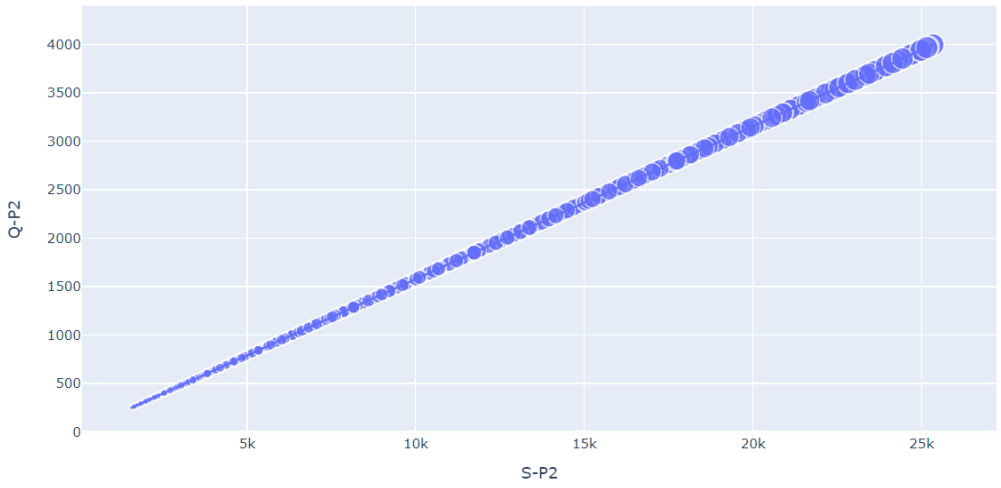
```
Unnamed: 0    0
Date          0
Q-P1          0
Q-P2          0
Q-P3          0
Q-P4          0
S-P1          0
S-P2          0
S-P3          0
S-P4          0
dtype: int64
```

Vizualizing the database by using jupyter,

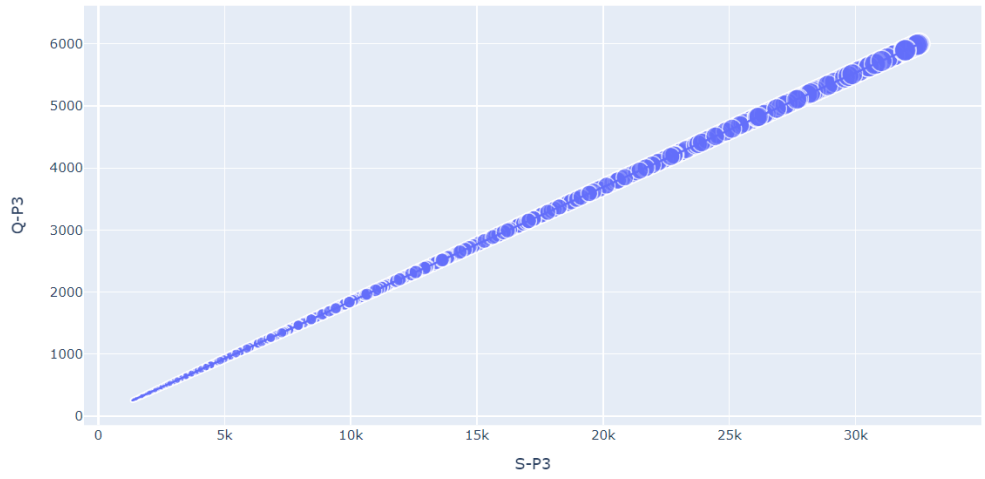
In [7]:
```
import plotly.express as px
import plotly.graph_objects as go
figure = px.scatter(data_frame = data, x="S-P1",
                    y="Q-P1", size="Q-P1", trendline="ols")
figure.show()
```



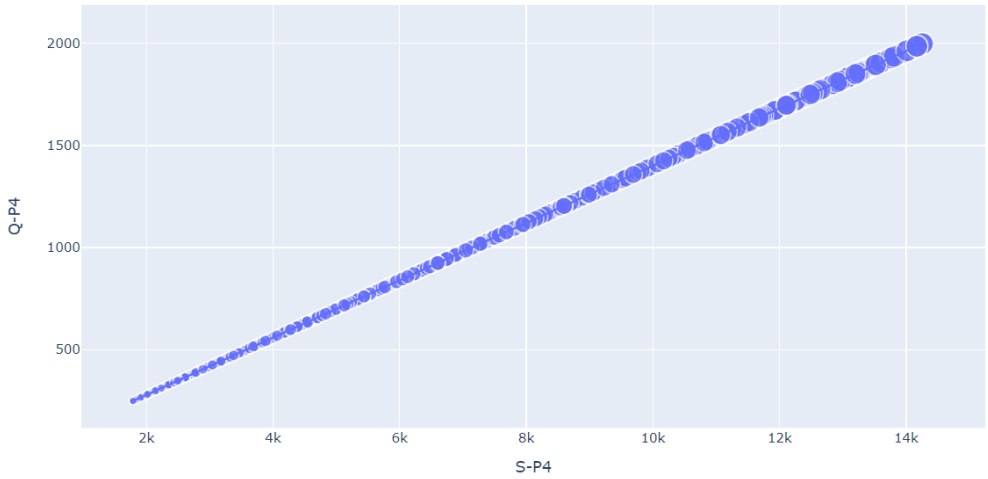
In [8]:
```
figure = px.scatter(data_frame = data, x="S-P2",
                    y="Q-P2", size="Q-P2", trendline="ols")
figure.show()
```



In [9]:
```
figure = px.scatter(data_frame = data, x="S-P3",
                    y="Q-P3", size="Q-P3", trendline="ols")
figure.show()
```



In [10]:
```
figure = px.scatter(data_frame = data, x="S-P4",
                    y="Q-P4", size="Q-P4", trendline="ols")
figure.show()
```



Performing correlation ,

```
del data['Date']
```

```
correlation = data.corr()
print(correlation["S-P1"].sort_values(ascending=False))
```

```
Q-P1          1.000000
S-P1          1.000000
S-P2          0.002422
Q-P2          0.002422
Unnamed: 0   -0.001120
S-P3         -0.005650
Q-P3         -0.005650
S-P4         -0.059365
Q-P4         -0.059365
Name: S-P1, dtype: float64
```

```
correlation = data.corr()
print(correlation["S-P2"].sort_values(ascending=False))
```

```
Q-P2          1.000000
S-P2          1.000000
S-P4          0.013082
Q-P4          0.013082
Unnamed: 0    0.008716
S-P3          0.003729
Q-P3          0.003729
S-P1          0.002422
Q-P1          0.002422
Name: S-P2, dtype: float64
```

```
correlation = data.corr()
print(correlation["S-P3"].sort_values(ascending=False))
```

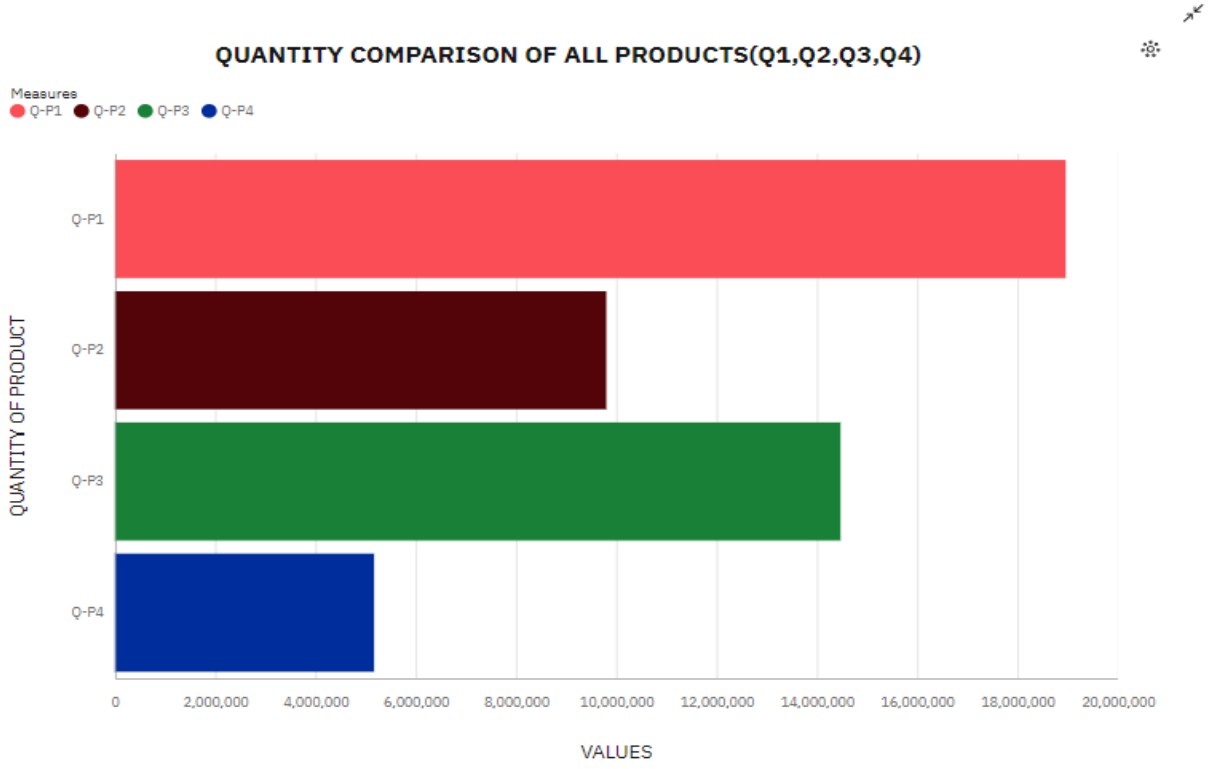
```
Q-P3          1.000000
S-P3          1.000000
Unnamed: 0    0.005255
S-P2          0.003729
Q-P2          0.003729
S-P1         -0.005650
Q-P1         -0.005650
S-P4         -0.006693
Q-P4         -0.006693
Name: S-P3, dtype: float64
```

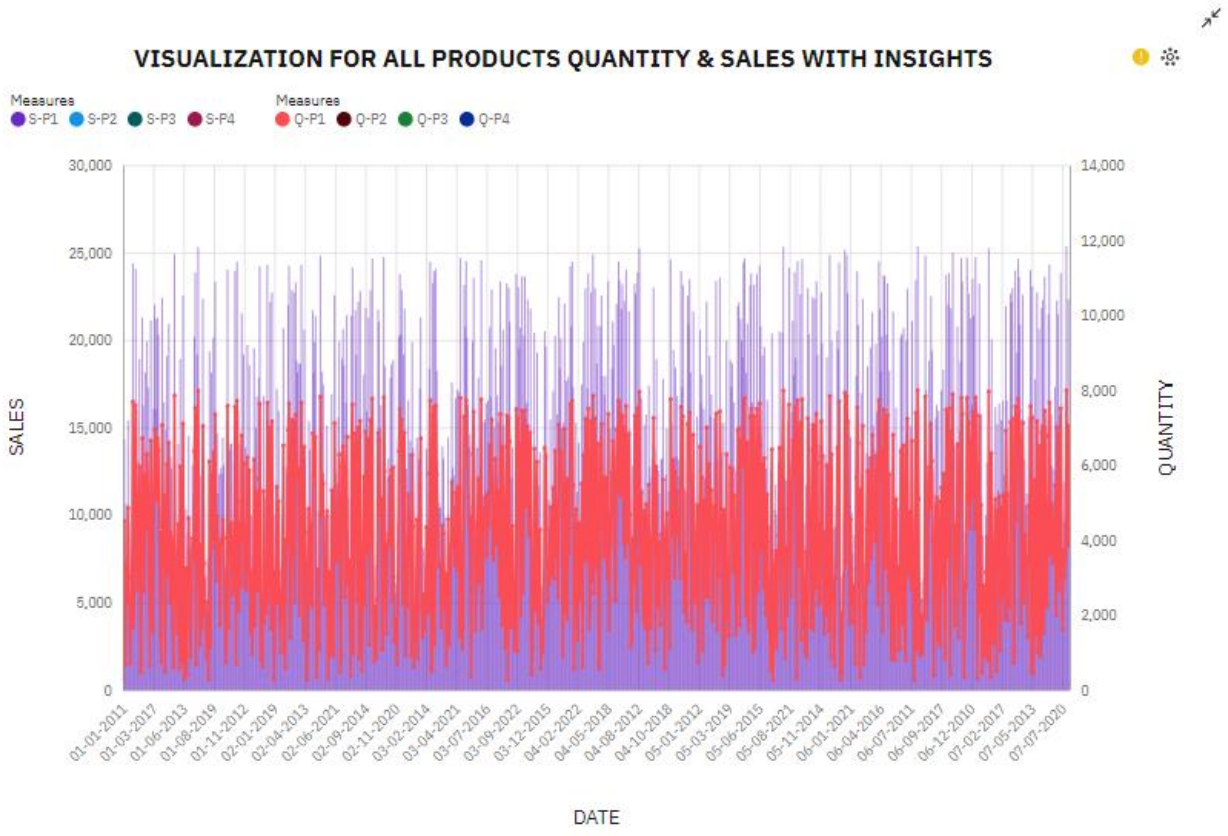
```
correlation = data.corr()
print(correlation["S-P4"].sort_values(ascending=False))
```

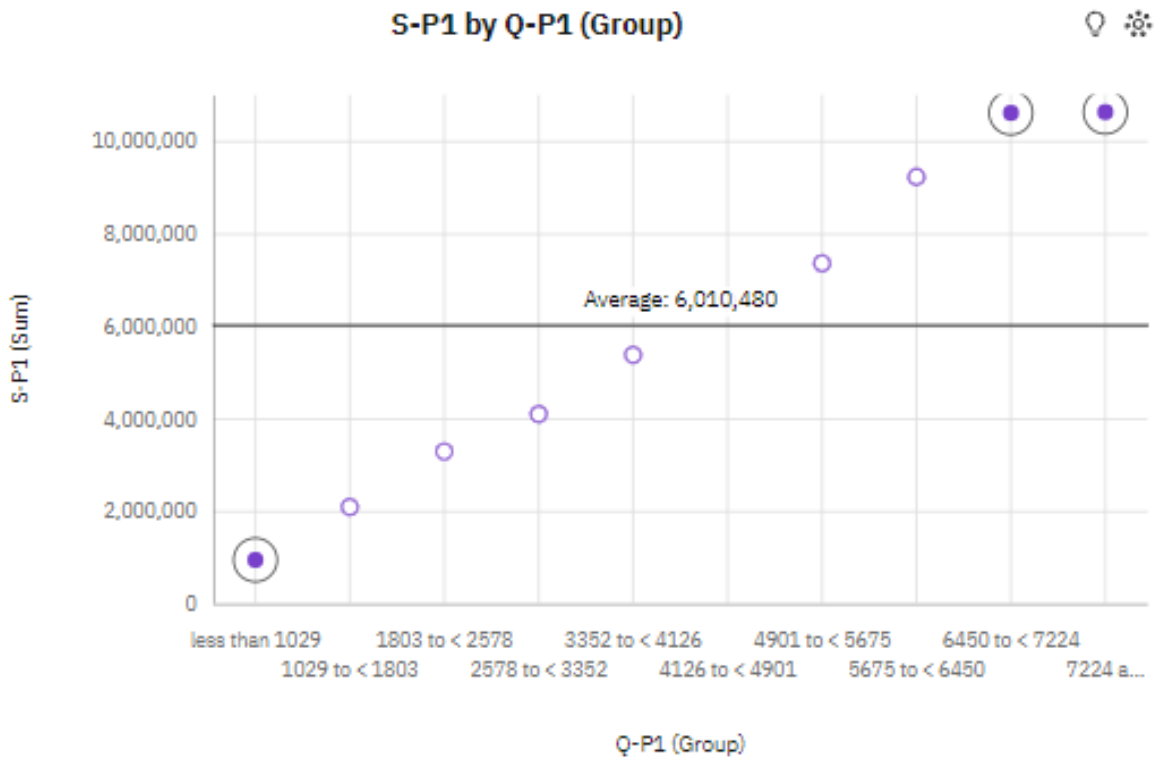
```
Q-P4          1.000000
S-P4          1.000000
S-P2          0.013082
Q-P2          0.013082
S-P3         -0.006693
Q-P3         -0.006693
Unnamed: 0   -0.009739
Q-P1         -0.059365
S-P1         -0.059365
Name: S-P4, dtype: float64
```

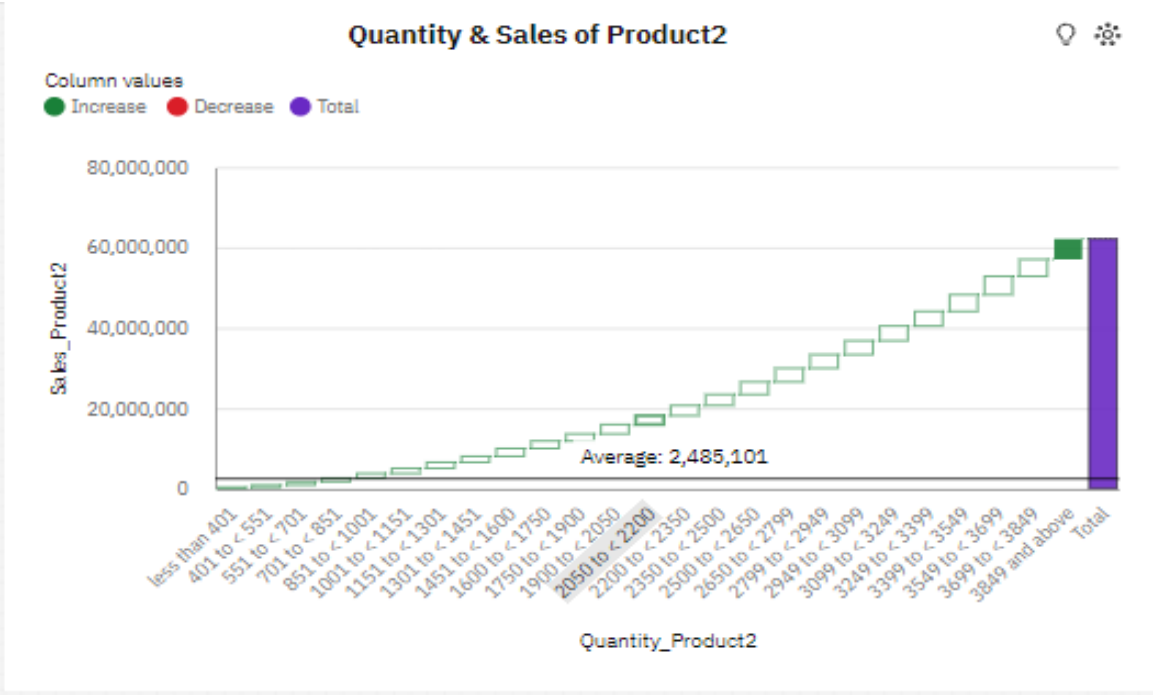
VISUALIZATION USING COGNOS

Let us building the product sales analysis using IBM Cognos for visualization.




VISUALIZATION FOR ALL PRODUCTS QUANTITY & SALES WITH INSIGHTS
Measures
S-P1 S-P2 S-P3 S-P4
Measures
Q-P1 Q-P2 Q-P3 Q-P4
SALES
QUANTITY
DATE


S-P1 by Q-P1 (Group)
S-P1 (Sum)
Average: 6,010,480
Q-P1 (Group)



CONCLUSION

Phase 2 of the Product Sales Analysis project focuses on visualizing and defining the analysis objectives and collect sales data from source shared and also process and clean the collected data to ensure its accuracy and reliability., you can ensure that your sales data analysis is based on accurate and reliable data, leading to more meaningful and actionable insights for your organization.

DATASET

Dataset Link :
(https://www.kaggle.com/datasets/ksabishek/product-sales-data)

Santhosh M 18.10.2023

Nandha College of Technology