

# **Data Analytics with Cognos Project**

## **PROJECT TITLE:**

**PRODUCT SALES ANALYSIS**

**INNOVATION PHASE**

## **TEAM MEMBERS:**

|                              |            |
|------------------------------|------------|
| MUTHU RATHNASHRI C           | 2021115068 |
| NITHYA SREE K                | 2021115071 |
| MUTHAMIZH VAANJINATHAN M     | 2021115067 |
| NAREN KARTHICK THIRUVENGADAM | 2021115069 |
| PRASAAD S                    | 2021115307 |

## **OVERVIEW**

Innovation in product sales analysis involves a series of essential steps to optimize performance and gain valuable insights. First, data collection and aggregation is paramount, as it forms the foundation of analysis. This includes gathering data on sales transactions, customer demographics, and market trends. The next step is data cleaning and preprocessing, ensuring accuracy and consistency. Once the data is ready, advanced analytics techniques, such as data mining and machine learning, can be employed to identify patterns, correlations, and outliers. Visualization tools are then used to present these findings in a digestible format. To foster innovation, predictive and prescriptive analytics can be applied, allowing for forecasting and actionable recommendations. Regular monitoring and feedback loops complete the process, enabling continuous improvement and adaptation to changing market dynamics.

## **DATA CLEANING**

### **1. Understanding the Data:**

Begin by understanding the dataset's structure, column names, data types, and identifying any missing or duplicate values. This step helps in formulating a strategy for cleaning the data effectively.

### **2. Handling Missing Values:**

Decide how to handle missing values in the dataset. Options include removing rows with missing values or filling missing values with appropriate measures like the mean or median.

### **3. Handling Duplicates:**

Identify and remove duplicate rows to ensure that the analysis is performed on unique data points, preventing skewed results.

### **4. Saving Cleaned Data:**

After completing the cleaning and preprocessing steps, save the cleaned dataset. This ensures that the cleaned data is available for further analysis and modeling without the need to repeat the cleaning process every time the analysis is performed.

About Dataset

- 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

# Data Cleaning

The screenshot shows a Jupyter Notebook in VS Code. The first cell imports pandas. The second cell reads a CSV file named 'statsfinal.csv'. The third cell displays the data, which is a table with 10 columns and 4600 rows.

```
1 import pandas as pd
```

```
1 data = pd.read_csv('statsfinal.csv')
2
```

```
1 display(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  |

4600 rows x 10 columns

The screenshot shows the same Jupyter Notebook with additional cells for data cleaning. The first cell drops rows with missing values. The second cell fills missing values with 0. The third cell drops duplicate rows.

```
1 data.dropna(inplace=True)
2
```

```
1 data.fillna(0, inplace=True)
2
```

```
1 data.drop_duplicates(inplace=True)
```

| 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  |

4600 rows x 10 columns

File Edit Selection View Go Run Terminal Help

Welcome

Code

```
1 data.fillna(0, inplace=True)
2
```

[14] ✓ 0.0s

```
1 data.drop_duplicates(inplace=True)
```

[15] ✓ 0.0s

```
1 data.to_csv('cleaned_data.csv', index=False)
```

[16] ✓ 0.0s

```
1 display(data)
```

[17] ✓ 0.0s

| 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  |

4600 rows x 10 columns

Ln 1, Col 45 CRLF Cell B

File Edit Selection View Go Run Terminal Help

phase\_3.ipynb

statsfinal.csv

cleaned\_data.csv

Code

```
1 # Extract year from the 'Day' 'Month' 'year' from the 'Date' column using a lambda function
2 # We need to get the year from the data to analyse sales year to year
3 data['Day'] = data['Date'].apply(lambda x: x.split('-')[0])
4 data['Month'] = data['Date'].apply(lambda x: x.split('-')[1])
5 data['Year'] = data['Date'].apply(lambda x: x.split('-')[2])
6 data
```

[54] ✓ 0.0s

| Date       | Q-P1 | Q-P2 | Q-P3 | Q-P4 | S-P1     | S-P2     | S-P3     | S-P4     | Day | Month | Year |
|------------|------|------|------|------|----------|----------|----------|----------|-----|-------|------|
| 13-06-2010 | 5422 | 3725 | 576  | 907  | 17187.74 | 23616.50 | 3121.92  | 6466.91  | 13  | 06    | 2010 |
| 14-06-2010 | 7047 | 779  | 3578 | 1574 | 22338.99 | 4938.86  | 19392.76 | 11222.62 | 14  | 06    | 2010 |
| 15-06-2010 | 1572 | 2082 | 595  | 1145 | 4983.24  | 13199.88 | 3224.90  | 8163.85  | 15  | 06    | 2010 |
| 16-06-2010 | 5657 | 2399 | 3140 | 1672 | 17932.69 | 15209.66 | 17018.80 | 11921.36 | 16  | 06    | 2010 |
| 17-06-2010 | 3668 | 3207 | 2184 | 708  | 11627.56 | 20332.38 | 11837.28 | 5048.04  | 17  | 06    | 2010 |
| ...        | ...  | ...  | ...  | ...  | ...      | ...      | ...      | ...      | ... | ...   | ...  |
| 30-01-2023 | 2476 | 3419 | 525  | 1359 | 7848.92  | 21676.46 | 2845.50  | 9689.67  | 30  | 01    | 2023 |
| 31-01-2023 | 7446 | 841  | 4825 | 1311 | 23603.82 | 5331.94  | 26151.50 | 9347.43  | 31  | 01    | 2023 |
| 01-02-2023 | 6289 | 3143 | 3588 | 474  | 19936.13 | 19926.62 | 19446.96 | 3379.62  | 01  | 02    | 2023 |
| 02-02-2023 | 3122 | 1188 | 5899 | 517  | 9896.74  | 7531.92  | 31972.58 | 3686.21  | 02  | 02    | 2023 |
| 03-02-2023 | 1234 | 3854 | 2321 | 406  | 3911.78  | 24434.36 | 12579.82 | 2894.78  | 03  | 02    | 2023 |

4600 rows x 12 columns

File Edit Selection View Go Run Terminal Help

phase\_3.ipynb statsfinal.csv cleaned\_data.csv

Python 3.9.6

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 pd.options.display.max_columns=50
6 sns.set(style="darkgrid")
```

[48]

1 data.head(5)

[49]

|   | 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 |

1 # Fetching rows and columns  
2 data.shape

[50]

(4600, 12)

1 # fetching column names  
2 data.columns

[51]

File Edit Selection View Go Run Terminal Help

phase\_3.ipynb statsfinal.csv cleaned\_data.csv

Python 3.9.6

1 # fetching column names  
2 data.columns

[51]

Index(['Date', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4', 'S-P1', 'S-P2', 'S-P3', 'S-P4',  
 'Day', 'Month', 'Year'],  
 dtype='object')

1 # basic info  
2 data.info()

[53]

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 12 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 
 9   Day         4600 non-null   object 
10  Month       4600 non-null   object 
11  Year        4600 non-null   object 
dtypes: float64(4), int64(4), object(4)
memory usage: 431.4+ KB
```

File Edit Selection View Go Run Terminal Help

phase\_3.ipynb statsfinal.csv cleaned\_data.csv

Python 3.9.6

```
1 # Checking null values
2 data.isnull().sum()
```

[55]

```
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
Day        0
Month      0
Year       0
dtype: int64
```

```
1 # Checking Dtypes
2 data.dtypes
```

[56]

```
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
Day        object
Month      object
Year       object
dtype: object
```

EXPLORER

OPEN EDITORS

- Welcome
- phase\_3.ipynb
- statsfinal.csv
- cleaned\_data.csv

NM

- a1.png
- a2.png
- a3.png
- c1.png
- c2.png
- cleaned\_data.csv
- p1.png
- p2.png
- p3.png
- phase\_3.ipynb
- start.py
- statsfinal.csv
- v1.png
- v2.png
- v3.png
- v4.png
- v5.png
- v6.png

OUTLINE

TIMELINE

File Edit Selection View Go Run Terminal Help

phase\_3.ipynb statsfinal.csv cleaned\_data.csv

Python 3.9.6

```
1 data.duplicated().sum()
```

[57]

```
0
```

```
1 ## Basic statistical info
2 data.describe().T
```

[58]

|      | count  | mean         | std         | min     | 25%      | 50%       | 75%       | max      |
|------|--------|--------------|-------------|---------|----------|-----------|-----------|----------|
| Q-P1 | 4600.0 | 4121.849130  | 2244.271323 | 254.00  | 2150.500 | 4137.000  | 6072.000  | 7998.00  |
| Q-P2 | 4600.0 | 2130.281522  | 1089.783705 | 251.00  | 1167.750 | 2134.000  | 3070.250  | 3998.00  |
| Q-P3 | 4600.0 | 3145.740000  | 1671.832231 | 250.00  | 1695.750 | 3202.500  | 4569.000  | 6000.00  |
| Q-P4 | 4600.0 | 1123.500000  | 497.385676  | 250.00  | 696.000  | 1136.500  | 1544.000  | 2000.00  |
| S-P1 | 4600.0 | 13066.261743 | 7114.340094 | 805.18  | 6817.085 | 13114.290 | 19248.240 | 25353.66 |
| S-P2 | 4600.0 | 13505.984848 | 6909.228687 | 1591.34 | 7403.535 | 13529.560 | 19465.385 | 25347.32 |
| S-P3 | 4600.0 | 17049.910800 | 9061.330694 | 1355.00 | 9190.965 | 17357.550 | 24763.980 | 32520.00 |
| S-P4 | 4600.0 | 8010.555000  | 3546.359869 | 1782.50 | 4962.480 | 8103.245  | 11008.720 | 14260.00 |

```
1 data.sample(2)
```

[59]

|      | Date       | Q-P1 | Q-P2 | Q-P3 | Q-P4 | S-P1     | S-P2     | S-P3     | S-P4    | Day | Month | Year |
|------|------------|------|------|------|------|----------|----------|----------|---------|-----|-------|------|
| 1628 | 03-12-2014 | 4261 | 844  | 600  | 434  | 13507.37 | 5350.96  | 3252.00  | 3094.42 | 03  | 12    | 2014 |
| 49   | 01-08-2010 | 2062 | 3659 | 2194 | 803  | 6536.54  | 23198.06 | 11891.48 | 5725.39 | 01  | 08    | 2010 |

EXPLORER

OPEN EDITORS

- Welcome
- phase\_3.ipynb
- statsfinal.csv
- cleaned\_data.csv

NM

- a1.png
- a2.png
- a3.png
- a4.png
- c1.png
- c2.png
- cleaned\_data.csv
- p1.png
- p2.png
- p3.png
- phase\_3.ipynb
- start.py
- statsfinal.csv
- v1.png
- v2.png
- v3.png
- v4.png
- v5.png
- v6.png

OUTLINE

TIMELINE

```
File Edit Selection View Go Run Terminal Help
Welcome phase_3.ipynb x a5.png statsfinal.csv cleaned_data.csv
phase_3.ipynb > # Checking null values
+ Code + Markdown | ▶ Run All ⌂ Restart Clear All Outputs | Variables Outline ...

1 # Changing dtype
2 from datetime import datetime as dt
3 data[data["Date"]=="31-9-2010"]

[60]
...
    Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year
109 31-9-2010 4986 342 4978 558 15805.62 2168.28 26980.76 3978.54 31 9 2010

1 data["Date"] = pd.to_datetime(data["Date"], errors='coerce')

[62]

▶ 1 data[data['Date'].isnull()]

[63]
...
    Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year
109 NaT 4986 342 4978 558 15805.62 2168.28 26980.76 3978.54 31 9 2010
170 NaT 4632 3930 523 1581 14683.44 24916.20 2834.66 11272.53 31 11 2010
473 NaT 2242 401 5926 789 7107.14 2542.34 32118.92 5625.57 31 9 2011
534 NaT 325 3476 4588 1771 1030.25 22037.84 24866.96 12627.23 31 11 2011
836 NaT 1003 256 1346 1449 3179.51 1623.04 7295.32 10331.37 31 9 2012
897 NaT 2509 2666 4146 593 7953.53 16902.44 22471.32 4228.09 31 11 2012
1200 NaT 597 709 5470 1994 1892.49 4495.06 29647.40 14217.22 31 9 2013
1261 NaT 7681 1235 347 1087 24348.77 7829.90 1880.74 7750.31 31 11 2013
1564 NaT 5333 833 3494 618 16905.61 5281.22 18937.48 4406.34 31 9 2014
1625 NaT 3870 2779 3246 1290 12267.90 17618.86 17593.32 9197.70 31 11 2014
1928 NaT 3583 2111 4225 1401 11358.11 13383.74 22899.50 9989.13 31 9 2015
1989 NaT 7516 3423 3116 458 23825.72 21701.82 16888.72 3265.54 31 11 2015
2291 NaT 7891 741 2280 1068 25014.47 4697.94 12357.60 7614.84 31 9 2016
2352 NaT 2457 3144 533 1184 7788.69 19932.96 2888.86 8441.92 31 11 2016
2655 NaT 3512 2851 4072 1597 11133.04 18075.34 22070.24 11386.61 31 9 2017
```

```
File Edit Selection View Go Run Terminal Help
Welcome phase_3.ipynb x a5.png statsfinal.csv cleaned_data.csv
phase_3.ipynb > # Checking null values
+ Code + Markdown | ▶ Run All ⌂ Restart Clear All Outputs | Variables Outline ... Python 3.9.6

1 data[data["Date"].isnull()]

[63]
...
    Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year
109 NaT 4986 342 4978 558 15805.62 2168.28 26980.76 3978.54 31 9 2010
170 NaT 4632 3930 523 1581 14683.44 24916.20 2834.66 11272.53 31 11 2010
473 NaT 2242 401 5926 789 7107.14 2542.34 32118.92 5625.57 31 9 2011
534 NaT 325 3476 4588 1771 1030.25 22037.84 24866.96 12627.23 31 11 2011
836 NaT 1003 256 1346 1449 3179.51 1623.04 7295.32 10331.37 31 9 2012
897 NaT 2509 2666 4146 593 7953.53 16902.44 22471.32 4228.09 31 11 2012
1200 NaT 597 709 5470 1994 1892.49 4495.06 29647.40 14217.22 31 9 2013
1261 NaT 7681 1235 347 1087 24348.77 7829.90 1880.74 7750.31 31 11 2013
1564 NaT 5333 833 3494 618 16905.61 5281.22 18937.48 4406.34 31 9 2014
1625 NaT 3870 2779 3246 1290 12267.90 17618.86 17593.32 9197.70 31 11 2014
1928 NaT 3583 2111 4225 1401 11358.11 13383.74 22899.50 9989.13 31 9 2015
1989 NaT 7516 3423 3116 458 23825.72 21701.82 16888.72 3265.54 31 11 2015
2291 NaT 7891 741 2280 1068 25014.47 4697.94 12357.60 7614.84 31 9 2016
2352 NaT 2457 3144 533 1184 7788.69 19932.96 2888.86 8441.92 31 11 2016
2655 NaT 3512 2851 4072 1597 11133.04 18075.34 22070.24 11386.61 31 9 2017
2716 NaT 6094 3798 5849 881 19317.98 24079.32 31701.58 6281.53 31 11 2017
3019 NaT 1727 2645 5715 1295 5474.59 16769.30 30975.30 9233.35 31 9 2018
3080 NaT 7360 2974 2717 1127 23331.20 18855.16 14726.14 8035.51 31 11 2018
3383 NaT 3195 2525 5918 1003 10128.15 16008.50 32075.56 7151.39 31 9 2019
3444 NaT 2660 2674 2732 934 8432.20 16953.16 14807.44 6659.42 31 11 2019
3746 NaT 4713 1227 4065 403 14940.21 7779.18 22032.30 2873.39 31 9 2020
3807 NaT 870 3463 798 851 2757.90 21955.42 4325.16 6067.63 31 11 2020
4110 NaT 3511 2609 1543 853 11129.87 16541.06 8363.06 6081.89 31 9 2021
4171 NaT 506 3333 3897 574 1604.02 21131.22 21121.74 4092.62 31 11 2021
4474 NaT 6964 1873 5481 1336 22075.88 11874.82 29707.02 9525.68 31 9 2022
4535 NaT 4600 2006 3796 1426 14582.00 12718.04 20574.32 10167.38 31 11 2022
```

```

File Edit Selection View Go Run Terminal Help
Welcome phase_3.ipynb a5.png statsfinal.csv cleaned_data.csv
phase_3.ipynb > # Checking null values
Code + Markdown Run All Restart Clear All Outputs Variables Outline Python 3.9.6
4535 NaN 4600 2006 3796 1426 14582.00 12718.04 20574.32 10167.38 31 11 2022

1 ## Filling the NaN values with average of time
2 data["Date"].fillna(data["Date"].mean(),inplace=True)

1 data["Date"].isnull().sum()

0

1 data.dtypes

Date datetime64[ns]
Q-P1 int64
Q-P2 int64
Q-P3 int64
Q-P4 int64
S-P1 float64
S-P2 float64
S-P3 float64
S-P4 float64
Day object
Month object
Year object
dtype: object

1 #fetching weekday
2 data["dayoftheweek"]=data["Date"].dt.weekday
3 data.sample()

Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year dayoftheweek
1352 2014-03-02 2357 3234 1781 1716 7471.69 20503.56 9653.02 12235.08 02 03 2014 6

```

```

File Edit Selection View Go Run Terminal Help
Welcome phase_3.ipynb a5.png statsfinal.csv cleaned_data.csv
phase_3.ipynb > # Checking null values
Code + Markdown Run All Restart Clear All Outputs Variables Outline Python 3.9.6

1 #fetching weekday
2 data["dayoftheweek"]=data["Date"].dt.weekday
3 data.sample()

Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year dayoftheweek
1352 2014-03-02 2357 3234 1781 1716 7471.69 20503.56 9653.02 12235.08 02 03 2014 6

1 data.corr().T

Date Q-P1 Q-P2 Q-P3 Q-P4 S-P1 S-P2 S-P3 S-P4 Day Month Year dayoftheweek
Date 1.000000 -0.001607 0.007008 0.004984 -0.008926 -0.001607 0.007008 0.004984 -0.008926 0.000672 -0.000456 0.993923 -0.001171
Q-P1 -0.001607 1.000000 0.002422 -0.005650 -0.059365 1.000000 0.002422 -0.005650 -0.059365 -0.000912 -0.009342 -0.000381 -0.002597
Q-P2 0.007008 0.002422 1.000000 0.003729 0.013082 0.002422 1.000000 0.003729 0.013082 -0.010996 -0.017790 0.010162 0.009255
Q-P3 0.004984 -0.005650 0.003729 1.000000 -0.006693 -0.005650 0.003729 1.000000 -0.006693 -0.028226 -0.006541 0.005943 0.007870
Q-P4 -0.008926 -0.059365 0.013082 -0.006693 1.000000 -0.059365 0.013082 -0.006693 1.000000 0.014732 0.005417 -0.010227 0.006432
S-P1 -0.001607 1.000000 0.002422 -0.005650 -0.059365 1.000000 0.002422 -0.005650 -0.059365 -0.000912 -0.009342 -0.000381 -0.002597
S-P2 0.007008 0.002422 1.000000 0.003729 0.013082 0.002422 1.000000 0.003729 0.013082 -0.010996 -0.017790 0.010162 0.009255
S-P3 0.004984 -0.005650 0.003729 1.000000 -0.006693 -0.005650 0.003729 1.000000 -0.006693 -0.028226 -0.006541 0.005943 0.007870
S-P4 -0.008926 -0.059365 0.013082 -0.006693 1.000000 -0.059365 0.013082 -0.006693 1.000000 0.014732 0.005417 -0.010227 0.006432
Day 0.000672 -0.000912 -0.010996 -0.028226 0.014732 -0.000912 -0.010996 -0.028226 0.014732 1.000000 0.008722 -0.006398 0.008808
Month -0.000456 -0.009342 -0.017790 -0.006541 0.005417 -0.009342 -0.017790 -0.006541 0.005417 0.008722 1.000000 -0.079013 0.005892
Year 0.993923 -0.000381 0.010162 0.005943 -0.010227 -0.000381 0.010162 0.005943 -0.010227 -0.006398 -0.079013 1.000000 -0.001590
dayoftheweek -0.001171 -0.002597 0.009255 0.007870 0.006432 -0.002597 0.009255 0.007870 0.006432 0.008808 0.005892 -0.001590 1.000000

1 plt.figure(figsize=(10,10))
2 sns.heatmap(data.corr(),annot=True)

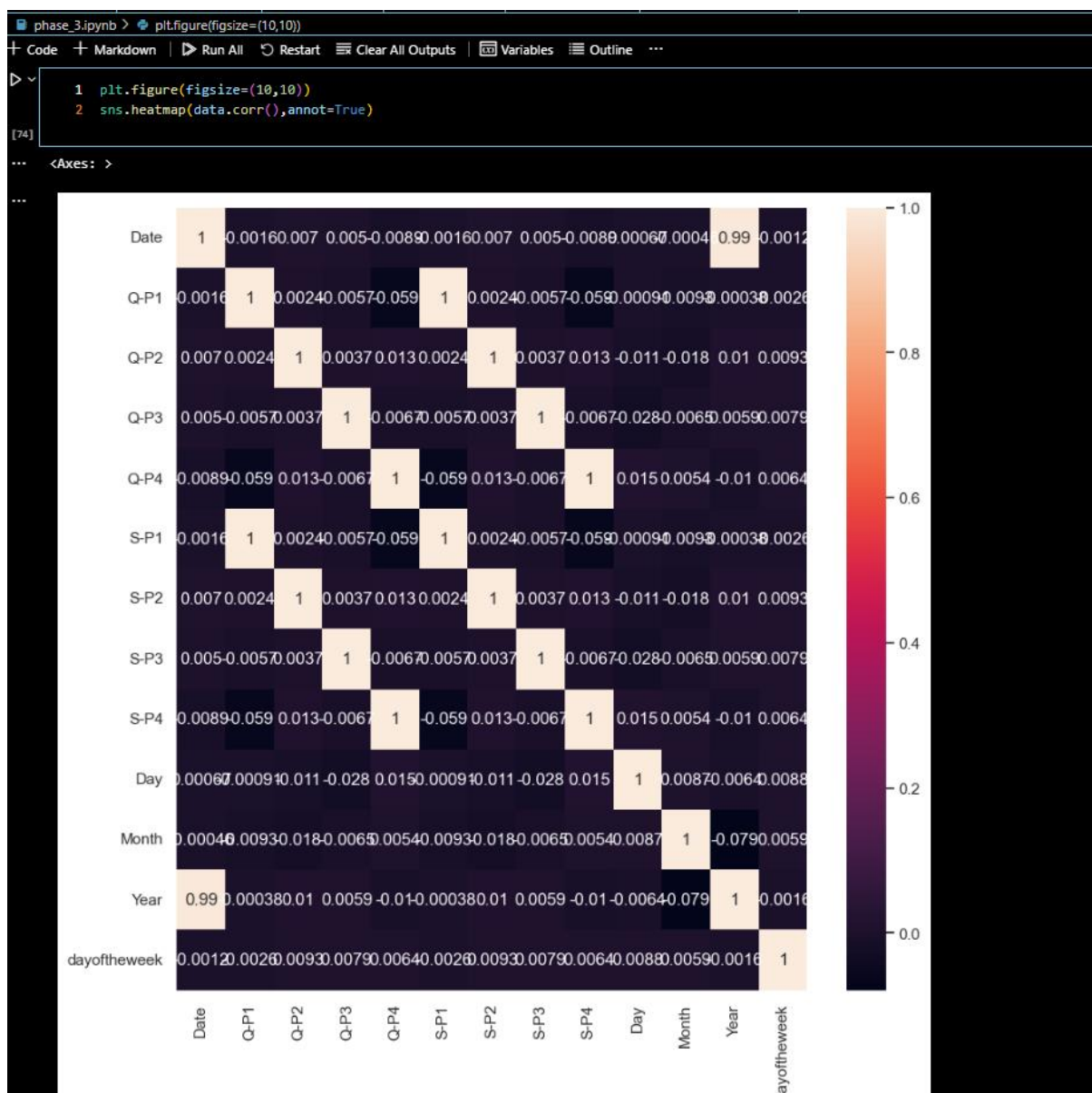
<Axes: >

```



## Visualization

In the product sales analysis project, visualization plays a pivotal role in transforming complex data into clear, actionable insights. Utilizing IBM Cognos, a powerful business intelligence tool, enables the creation of intuitive and interactive visualizations. Through compelling charts, graphs, and dashboards, patterns and trends in product sales data become immediately apparent. Visualizations allow stakeholders to grasp sales performance across products, regions, and time periods effortlessly. This visual representation not only enhances data-driven decision-making but also facilitates the communication of key findings, empowering businesses to strategize effectively, optimize inventory, identify market opportunities, and enhance overall sales performance.



There is no strong correlation between the columns

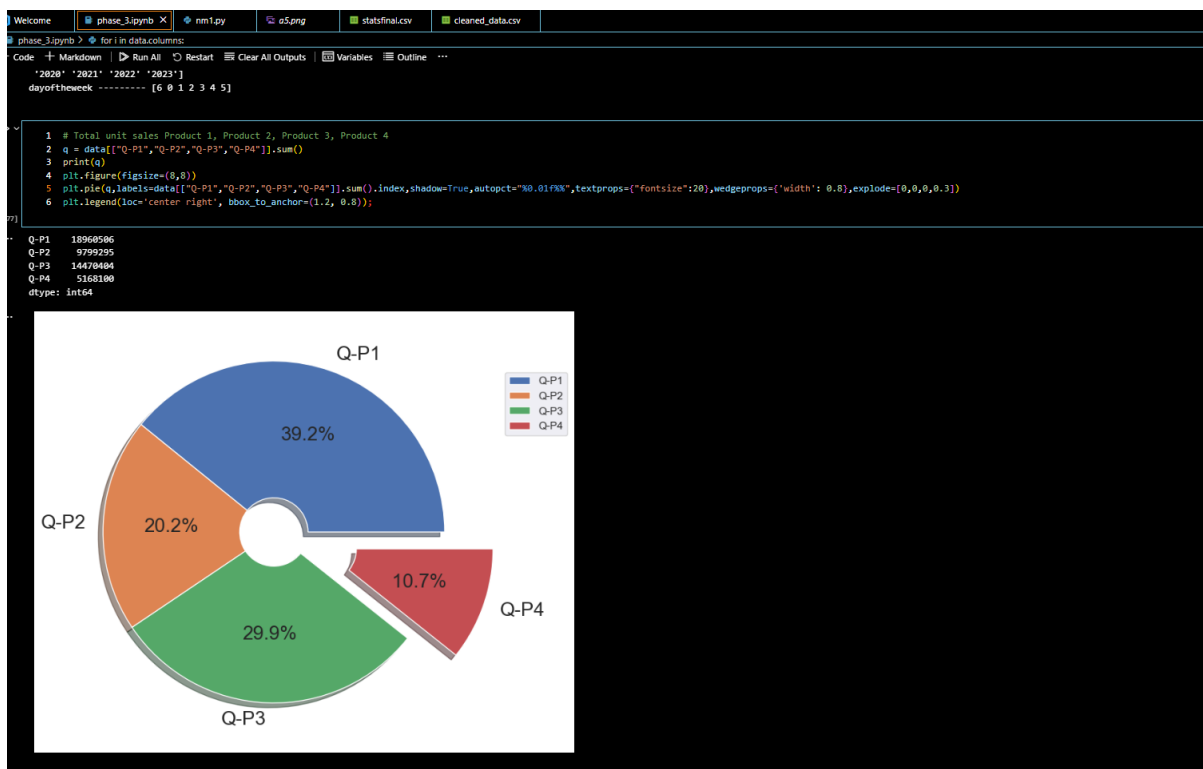
```
phase_3.ipynb > for i in data.columns:
1   print(i, "-----", data[i].unique())
2

Date -----> DatetimeArray
['2018-06-13 00:00:00', '2018-06-14 00:00:00', '2018-06-15 00:00:00',
 '2018-06-16 00:00:00', '2018-06-17 00:00:00', '2018-06-18 00:00:00',
 '2018-06-19 00:00:00', '2018-06-20 00:00:00', '2018-06-21 00:00:00',
 '2018-06-22 00:00:00',
 ...
 '2023-01-25 00:00:00', '2023-01-26 00:00:00', '2023-01-27 00:00:00',
 '2023-01-28 00:00:00', '2023-01-29 00:00:00', '2023-01-30 00:00:00',
 '2023-01-31 00:00:00', '2023-02-01 00:00:00', '2023-02-02 00:00:00',
 '2023-02-03 00:00:00']
Length: 4575, dtype: datetime64[ns]
Q-P1 -----> [5422 7047 1572 ... 1227 3122 1234]
Q-P2 -----> [3725 779 2802 ... 3404 042 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 -----> [22616.5  4936.86 13199.88 ... 21581.36 5331.94 18926.62]
S-P3 -----> [ 3121.92 19392.76 3224.9 ... 26151.5 19446.96 31972.58]
S-P4 -----> [ 6466.91 11222.62 8163.85 ... 8277.93 8286.63 7928.56]
Day -----> ['13' '14' '15' '16' '17' '18' '19' '20' '21' '22' '23' '24' '25' '26'
 '27' '28' '29' '30' '01' '02' '03' '04' '05' '06' '07' '08' '09' '10'
 '11' '12' '31']
Month -----> ['06' '07' '08' '09' '9' '10' '11' '12' '01' '02' '03' '04' '05']
Year -----> ['2018' '2011' '2012' '2013' '2014' '2015' '2016' '2017' '2018' '2019'
 '2020' '2021' '2022' '2023']
dayoftheweek -----> [ 6 0 1 2 3 4 5]

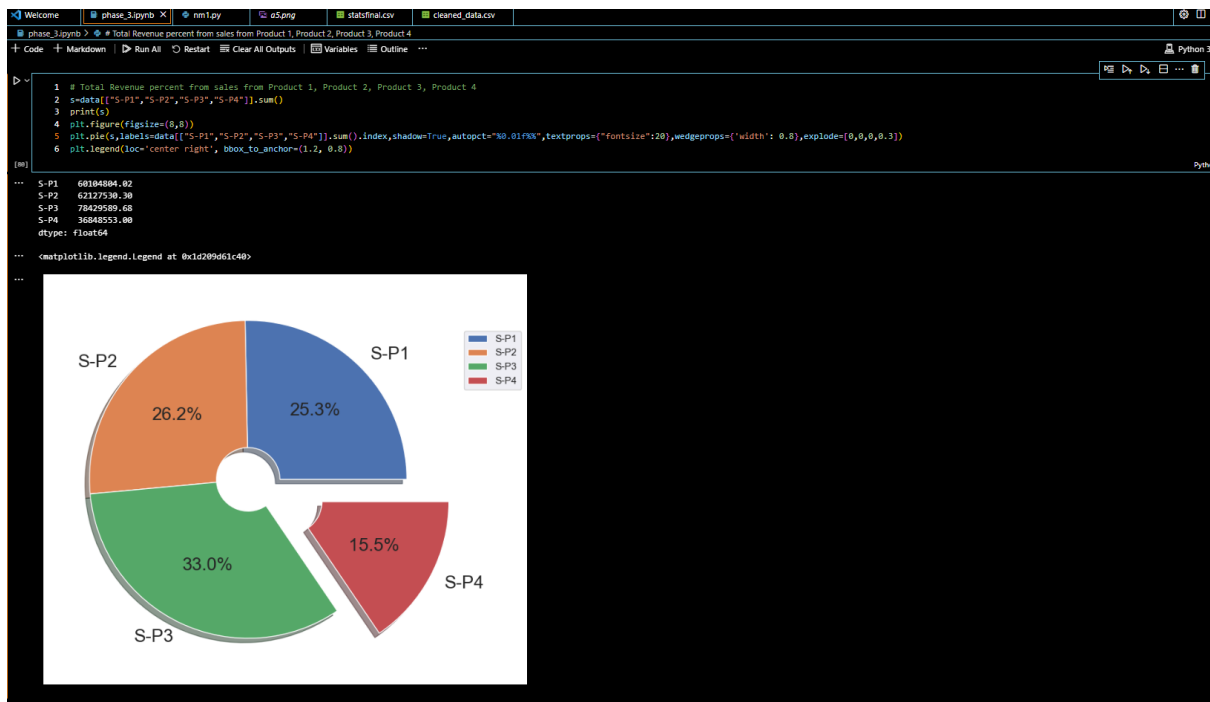
1 # Total unit sales Product 1, Product 2, Product 3, Product 4
2 q = data[["Q-P1", "Q-P2", "Q-P3", "Q-P4"]].sum()
3 print(q)
4 plt.figure(figsize=(8,8))
5 plt.pie(q, labels=data[["Q-P1", "Q-P2", "Q-P3", "Q-P4"]].sum().index, shadow=True, autopct="%0.1f%%", textprops={"fontSize":20}, wedgeprops={"width": 0.8}, explode=[0,0,0,0.3])
6 plt.legend(loc="center right", bbox_to_anchor=(1.2, 0.8));

Q-P1 18960596
Q-P2 9799295
Q-P3 14470404
Q-P4 5168100
dtype: int64
```

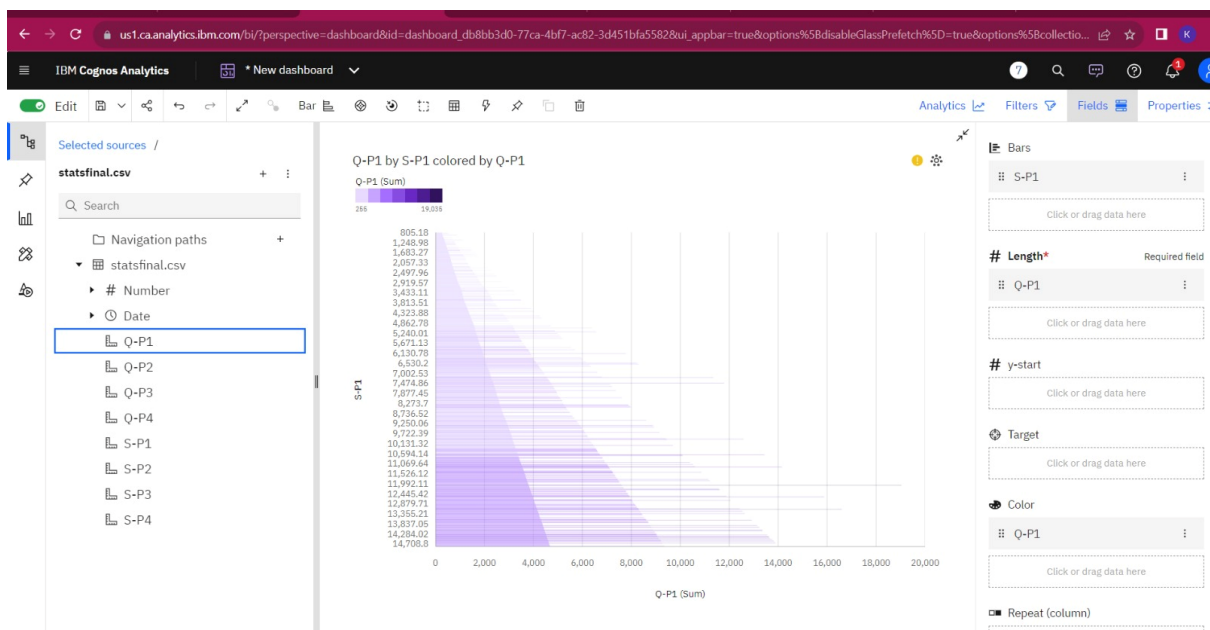
- Total unit sales Product 1, Product 2, Product 3, Product 4



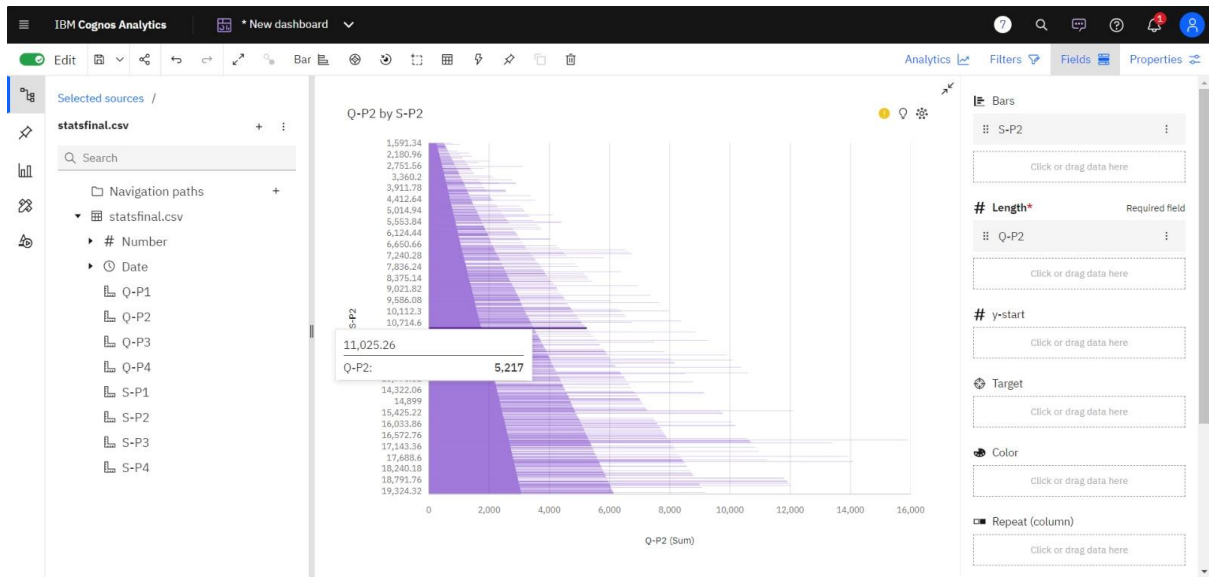
- Total Revenue percent from sales from Product 1, Product 2, Product 3, Product 4



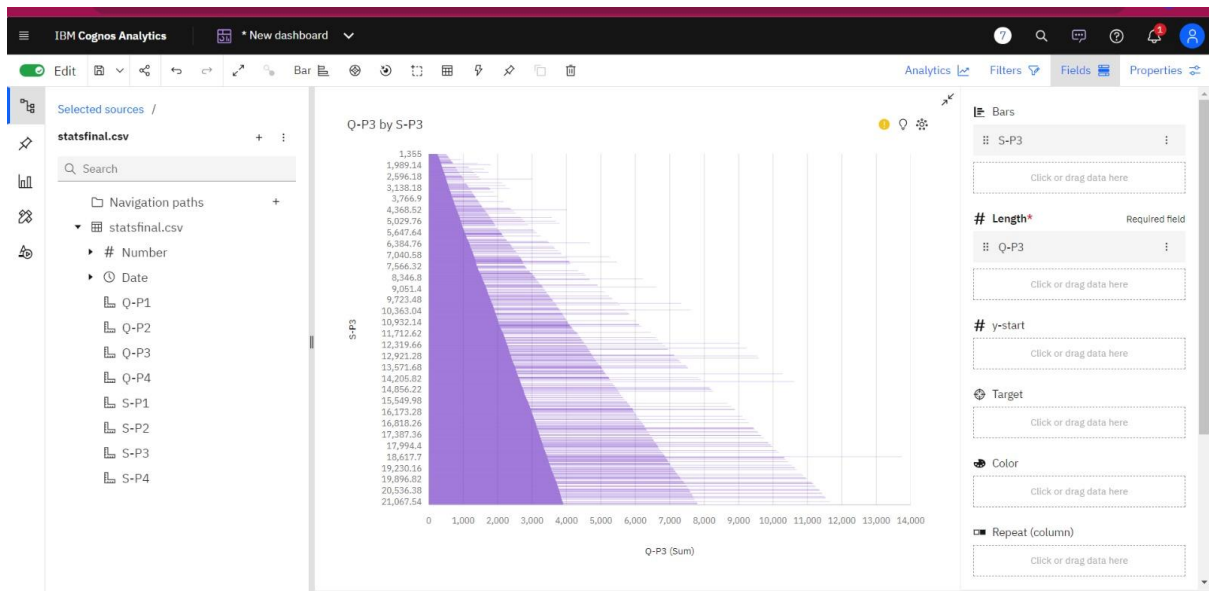
- Graph between Q-P1 and S-P1



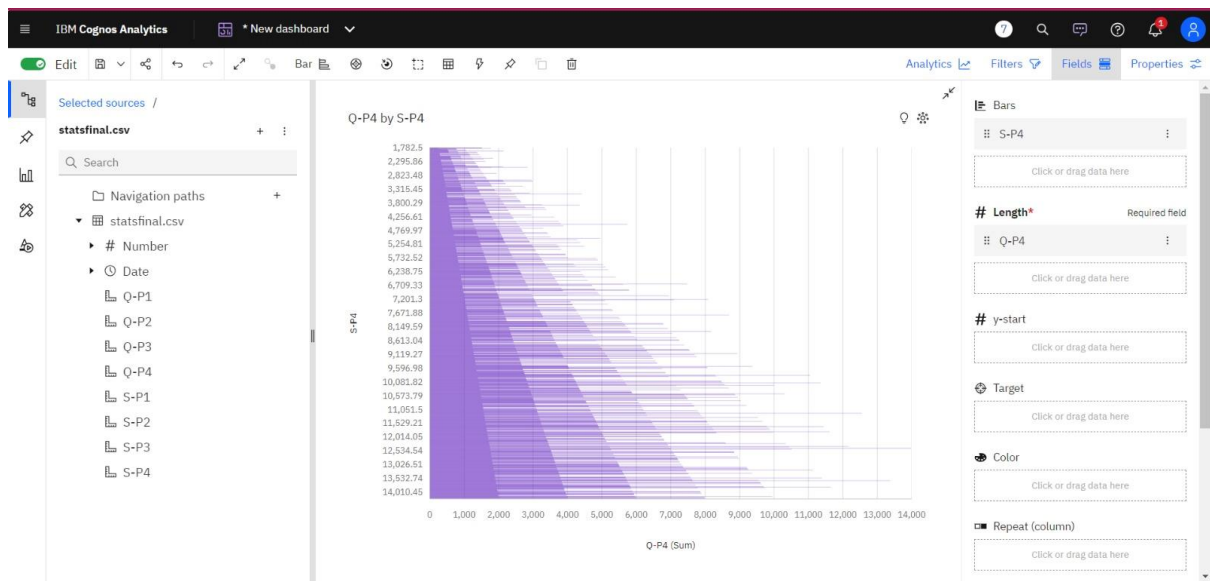
- Graph between Q-P2 and S-P2



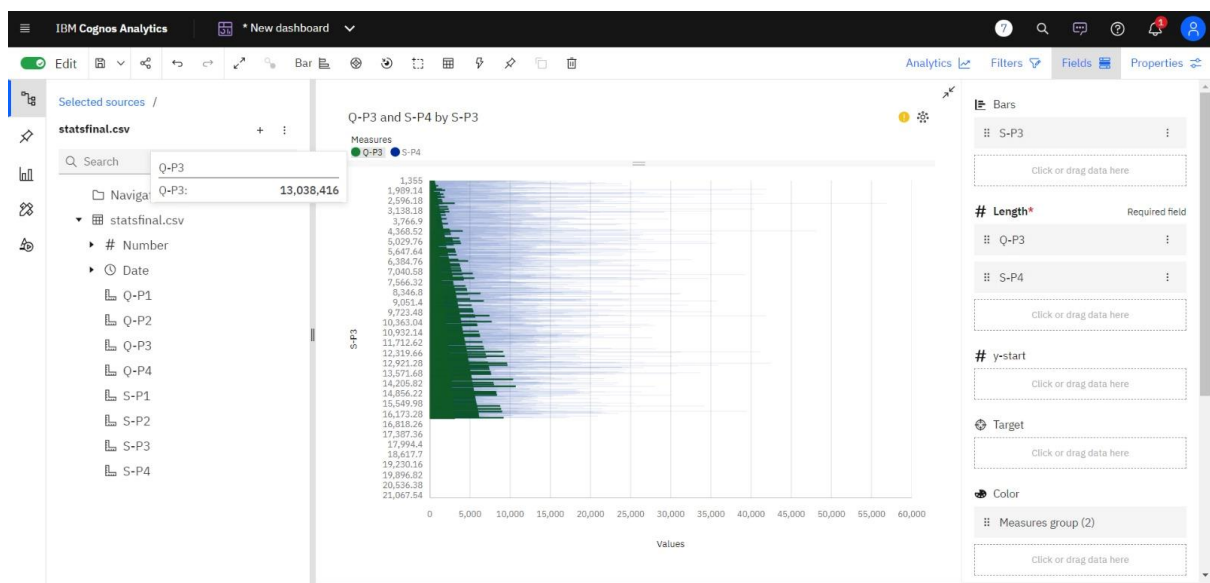
- Graph between Q-P3 and S-P3

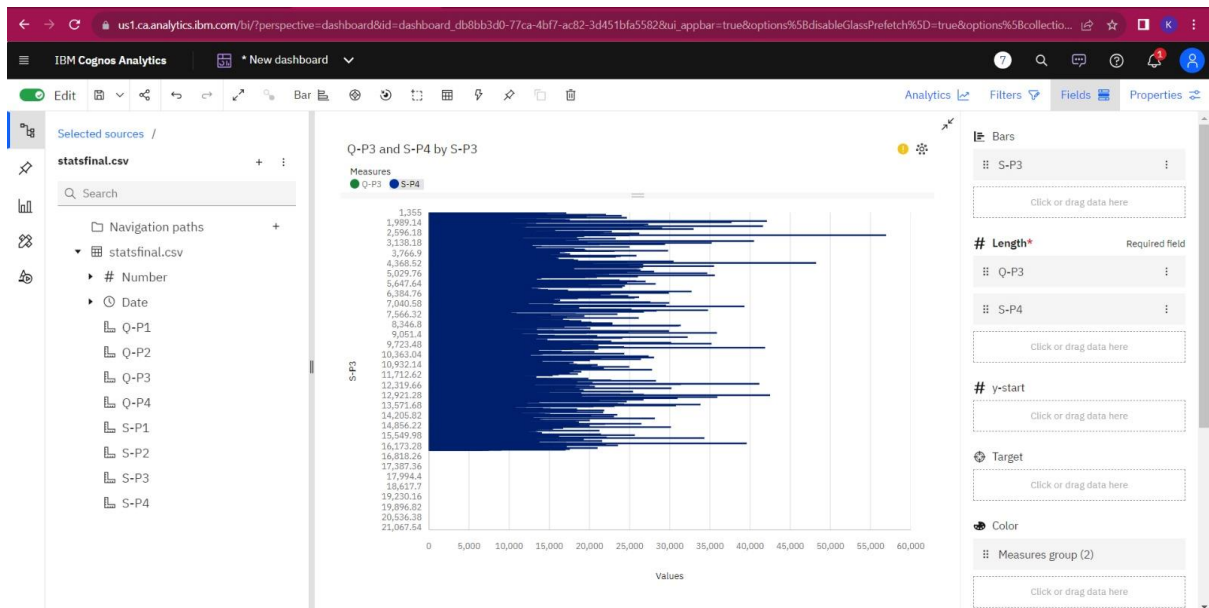


- Graph between Q-P4 and S-P4

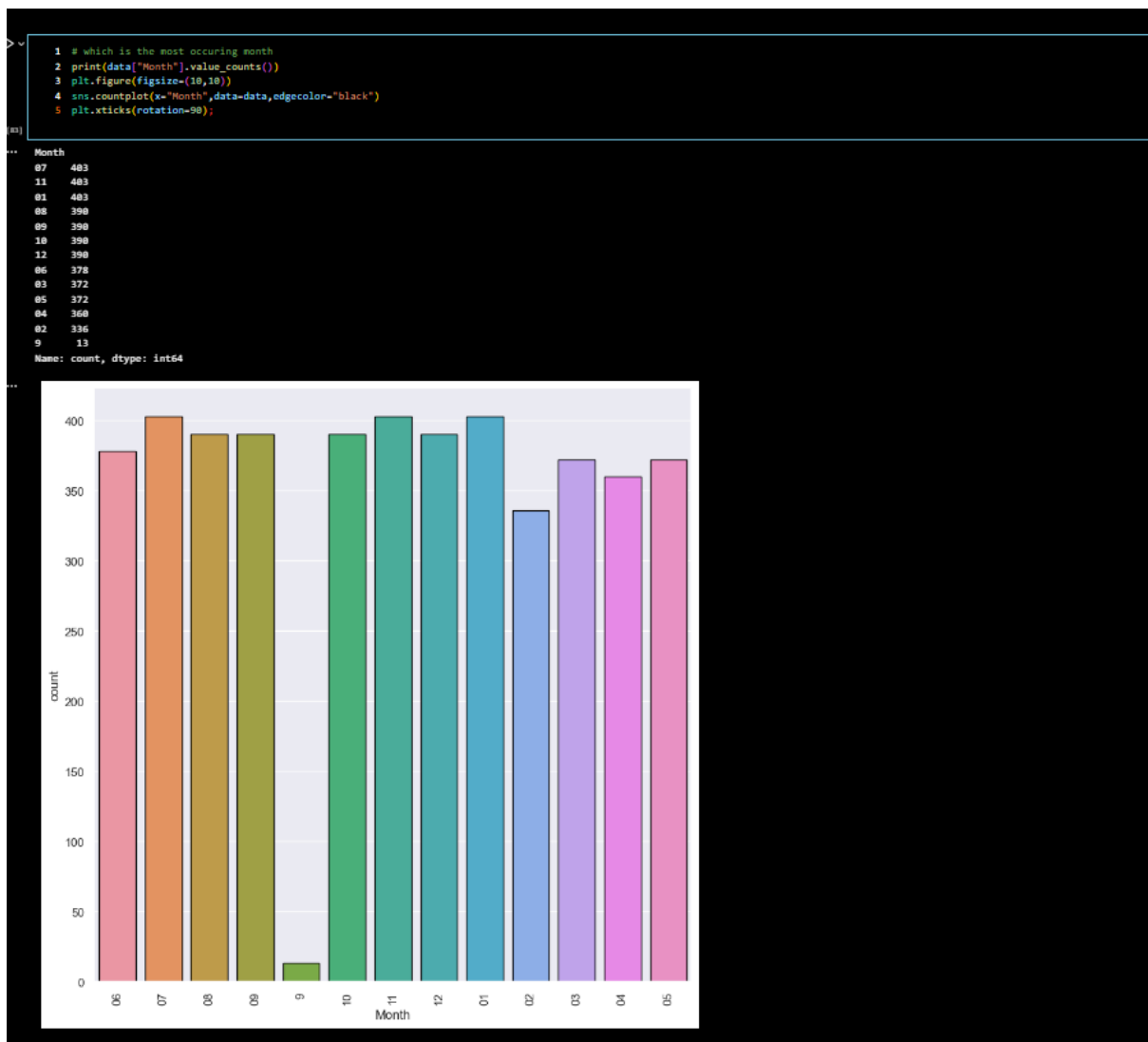


- Graph between Q-P3 and S-P3

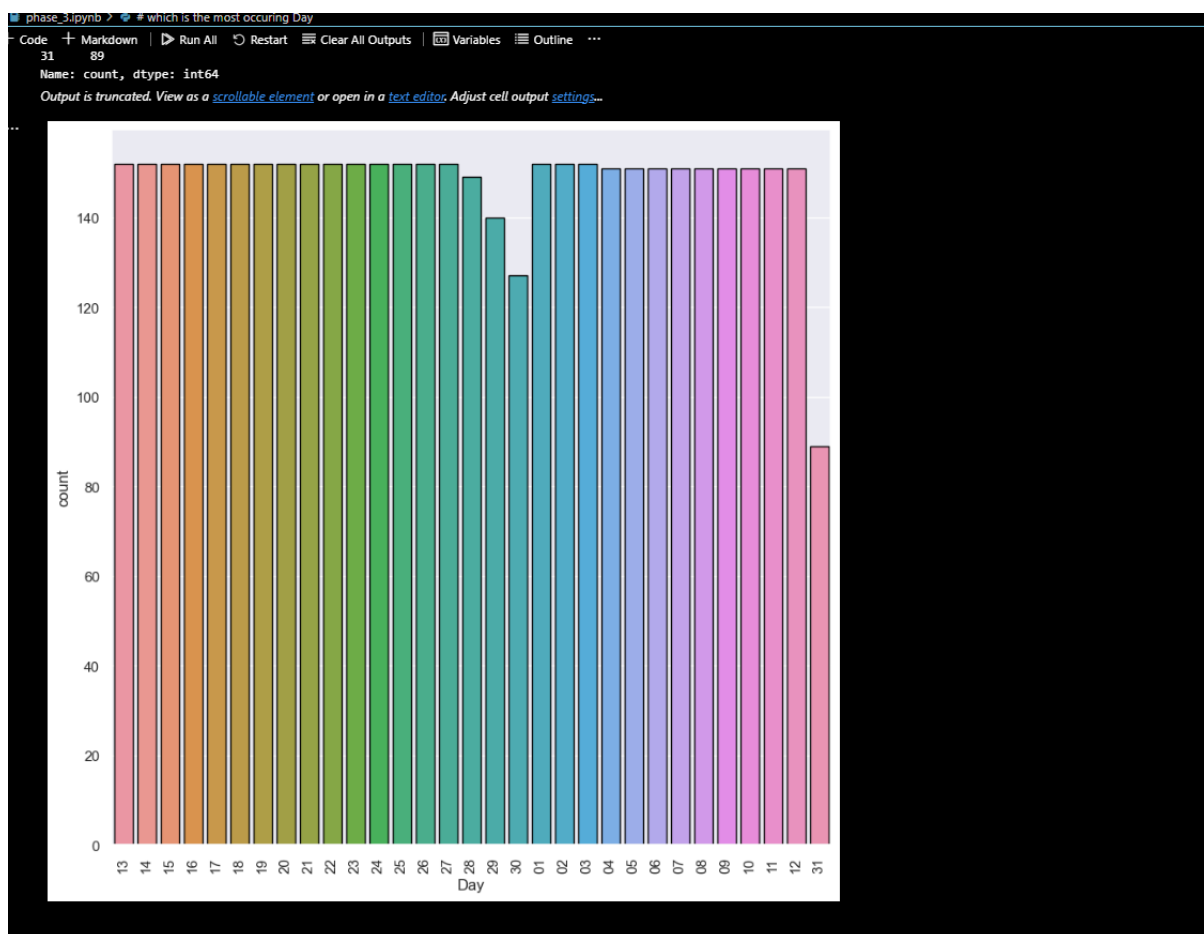
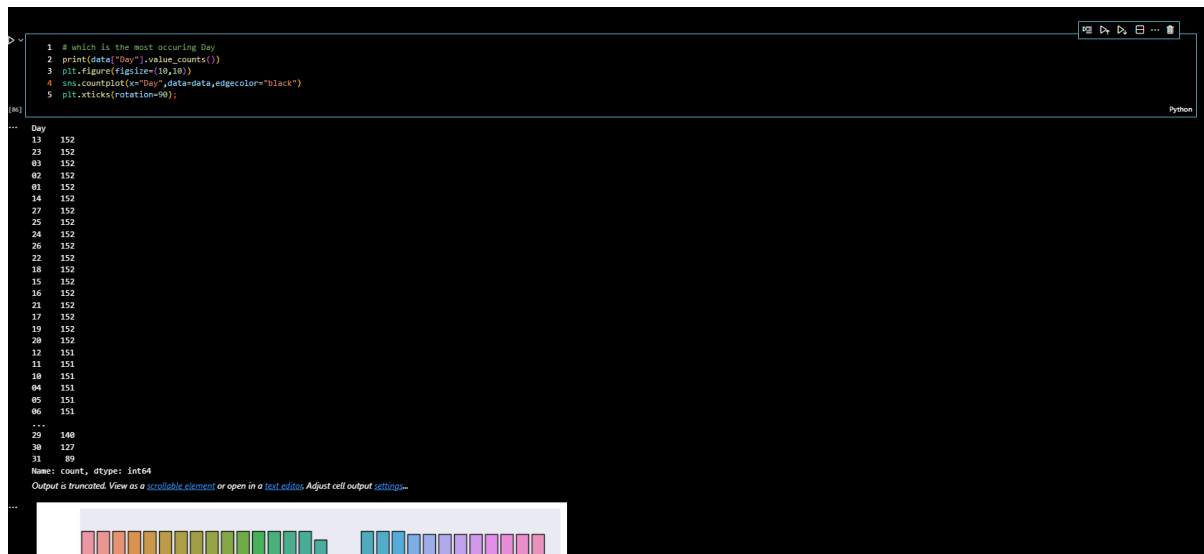




- Bar graph represents which is most occurring month



- Bar graph represents which is most occurring day



- Bar graph represents which is most occurring year

