# **PROJECT:PRODUCT SALES ANALYSIS**

# **PHASE 3- DEVELOPMENT PHASE 3**

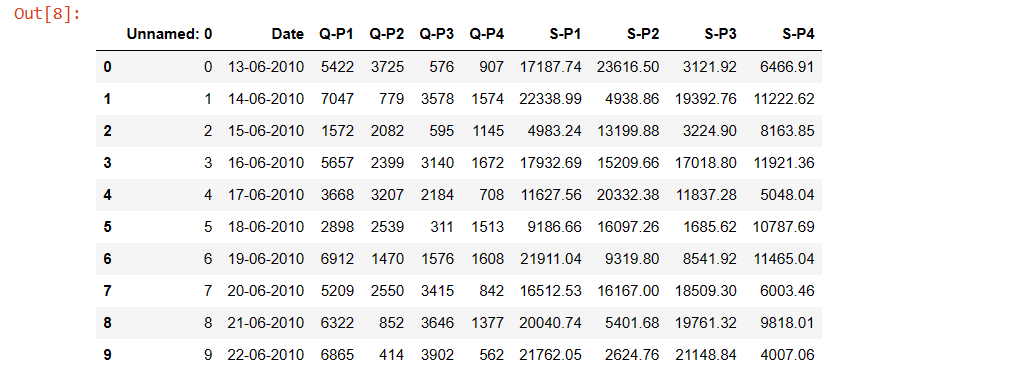
**PREPROCESSING AND CLEANSING OF DATA**

**CLEANING OF DATASET:**

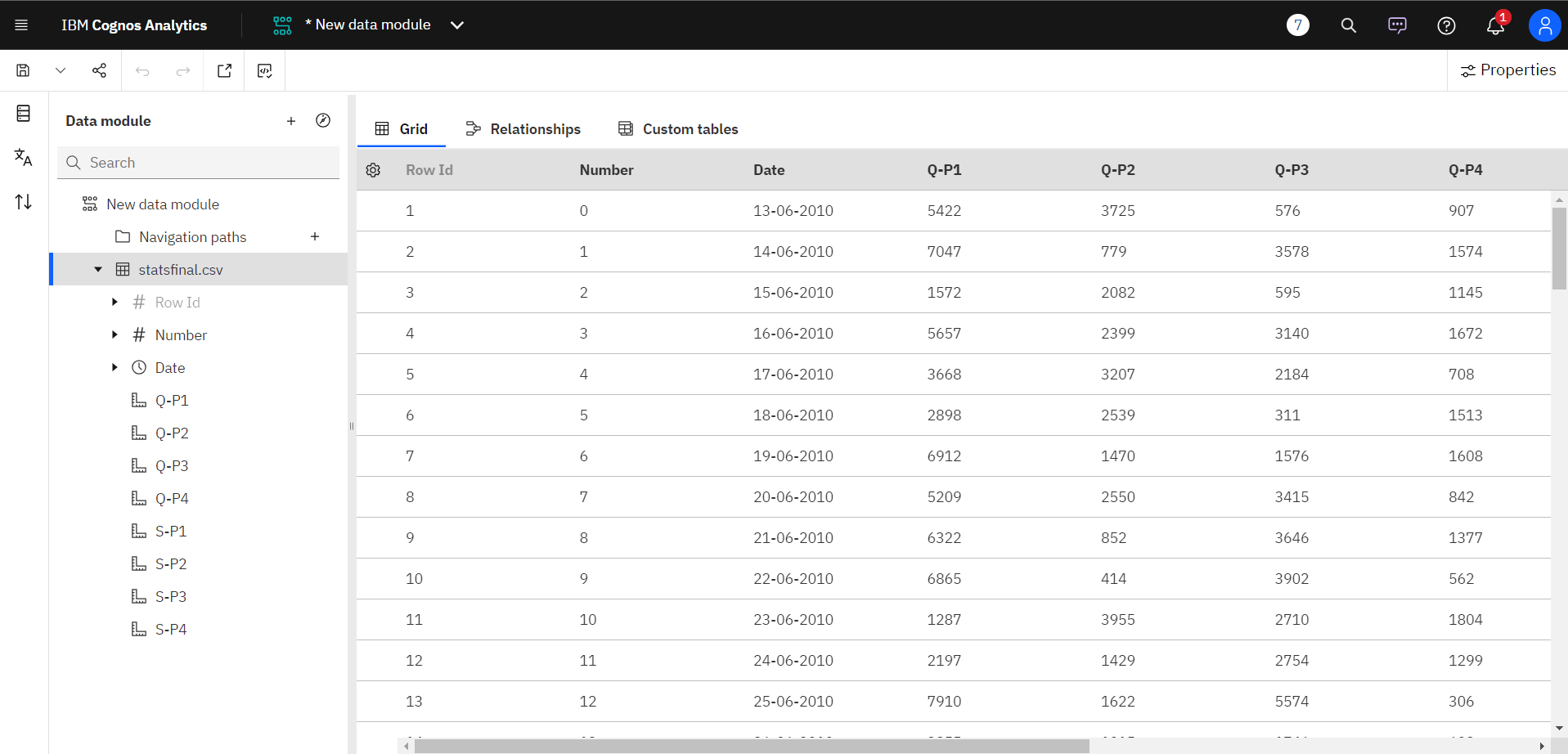
Cleaning of the dataset includes removing duplicates,handling the missing values, handling outliers,data scaling and normalization, data visualization,data splitting and data balancing if needed.

1.Removing duplicates:

data=data.dropna()



From IBM Cognos:



2. Handling outliers:

On checking outliers by scatter plot .

For product1:

import seaborn as sns

import matplotlib.pyplot as plt

# Set some default parameters of matplotlib

plt.rcParams['figure.figsize'] = (8, 6)

plt.rcParams['figure.dpi'] = 150

# Use style froms seaborn. Try to comment the next line and see the difference in graph

sns.set()

# A regular scatter plot

plt.scatter(x=data["Q-P1"], y=data["S-P1"])

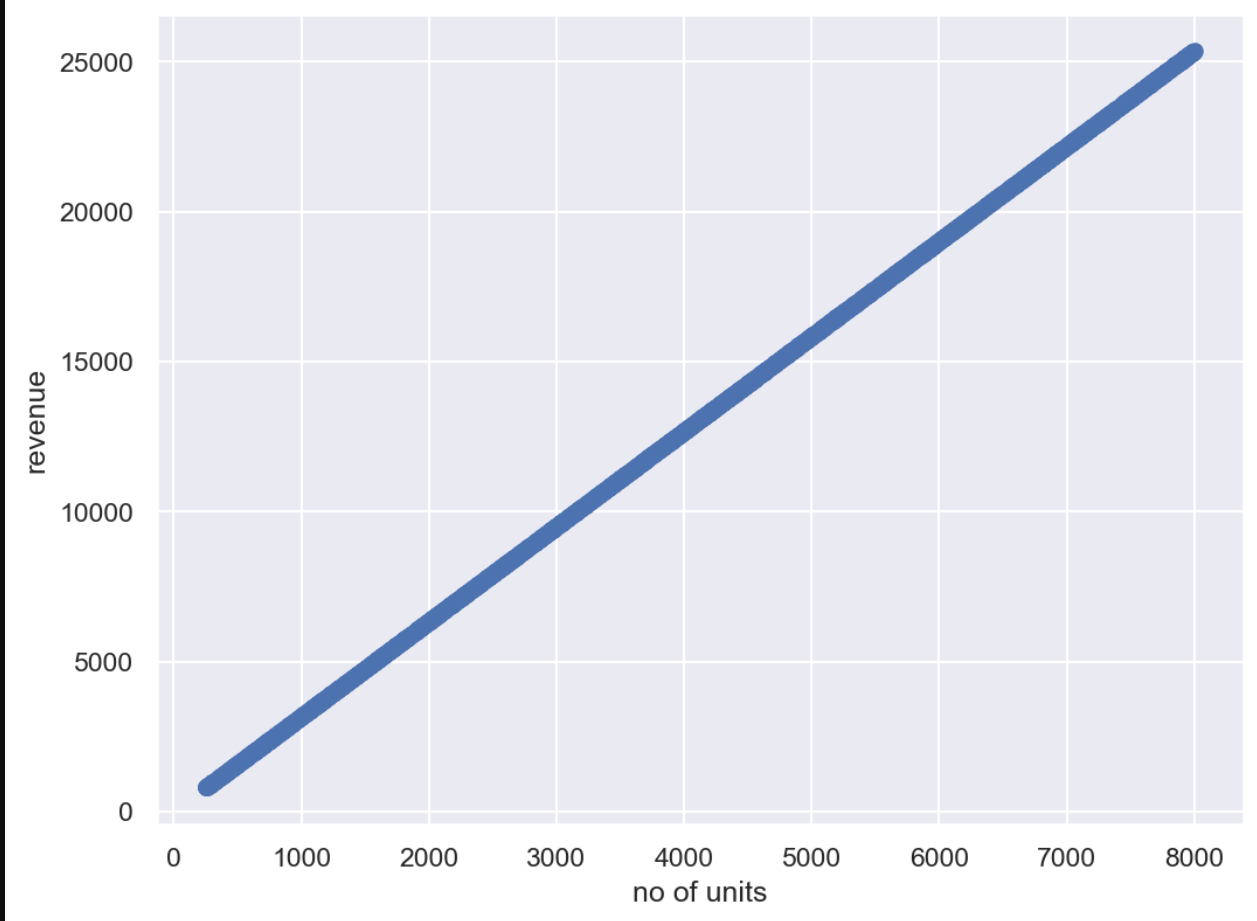
# Create labels for axises

plt.xlabel('no of units')

plt.ylabel('revenue')

# Display the plot on the screen

plt.show()



For product 2:

plt.rcParams['figure.figsize'] = (8, 6)

plt.rcParams['figure.dpi'] = 150

# Use style froms seaborn. Try to comment the next line and see the difference in graph

sns.set()

# A regular scatter plot

plt.scatter(x=data["Q-P2"], y=data["S-P2"])

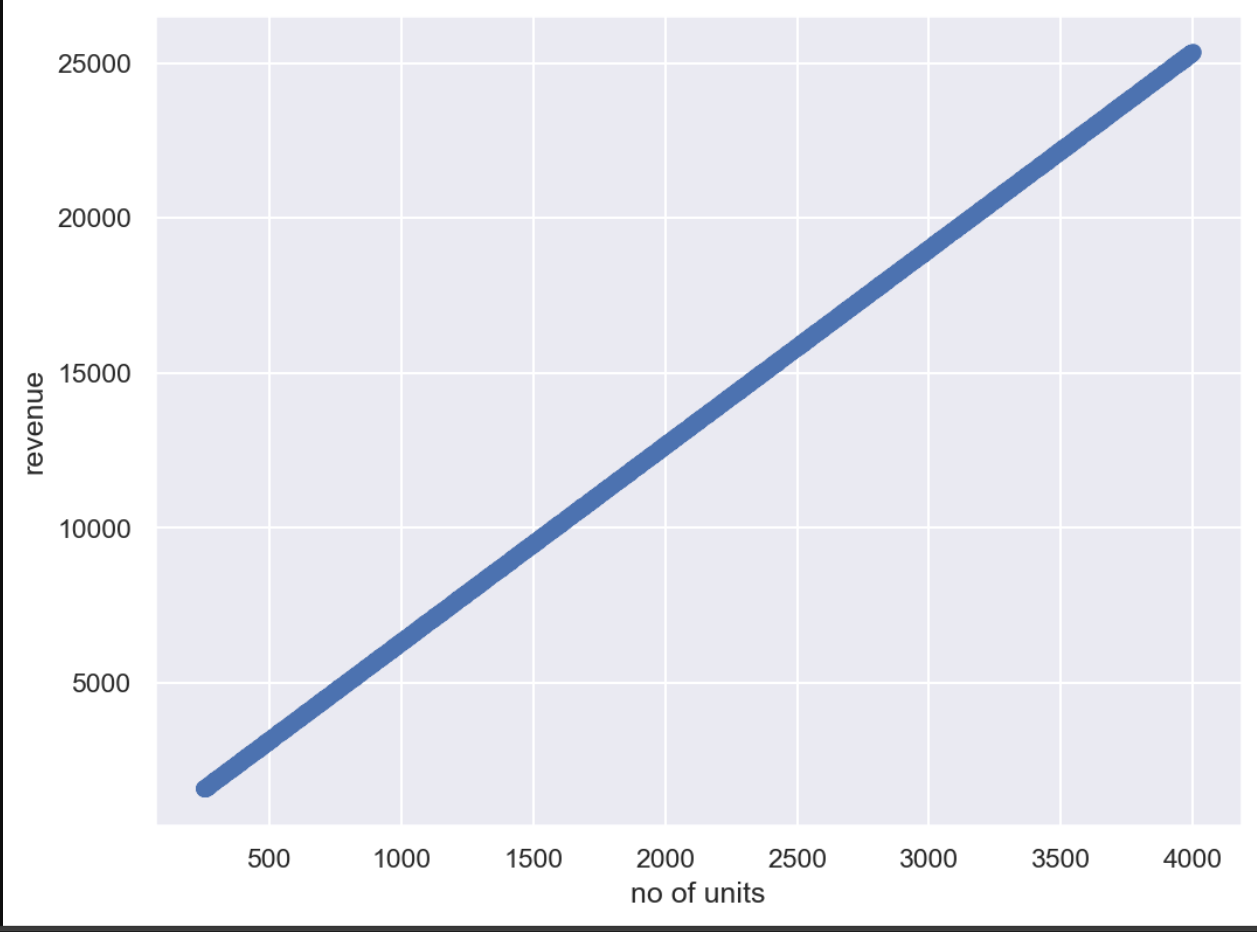
# Create labels for axises

plt.xlabel('no of units')

plt.ylabel('revenue')

# Display the plot on the screen

plt.show()



For product 3:

plt.rcParams['figure.figsize'] = (8, 6)

plt.rcParams['figure.dpi'] = 150

# Use style froms seaborn. Try to comment the next line and see the difference in graph

sns.set()

# A regular scatter plot

plt.scatter(x=data["Q-P3"], y=data["S-P3"])

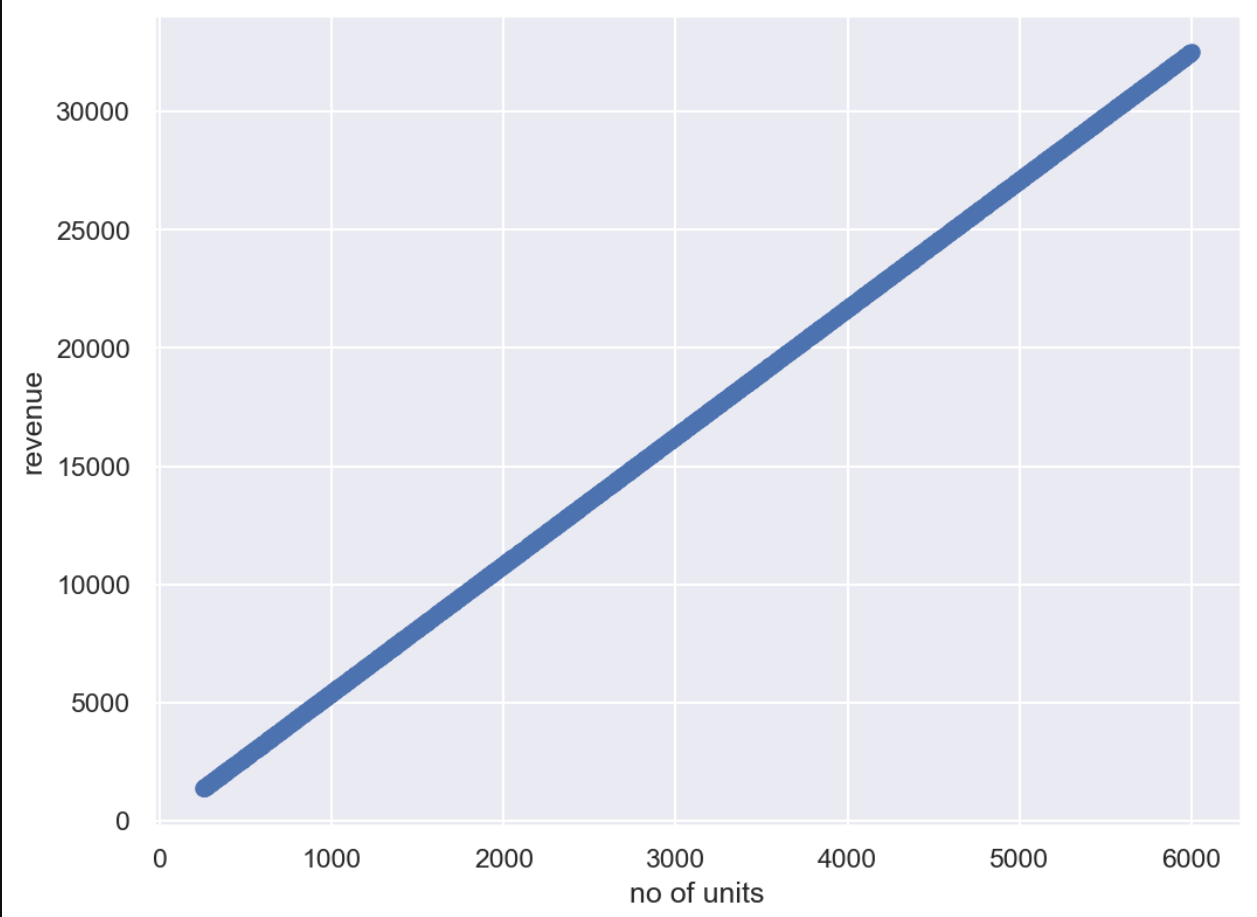
# Create labels for axises

plt.xlabel('no of units')

plt.ylabel('revenue')

# Display the plot on the screen

plt.show()



For product 4:

plt.rcParams['figure.figsize'] = (8, 6)

plt.rcParams['figure.dpi'] = 150

# Use style froms seaborn. Try to comment the next line and see the difference in graph

sns.set()

# A regular scatter plot

plt.scatter(x=data["Q-P4"], y=data["S-P4"])

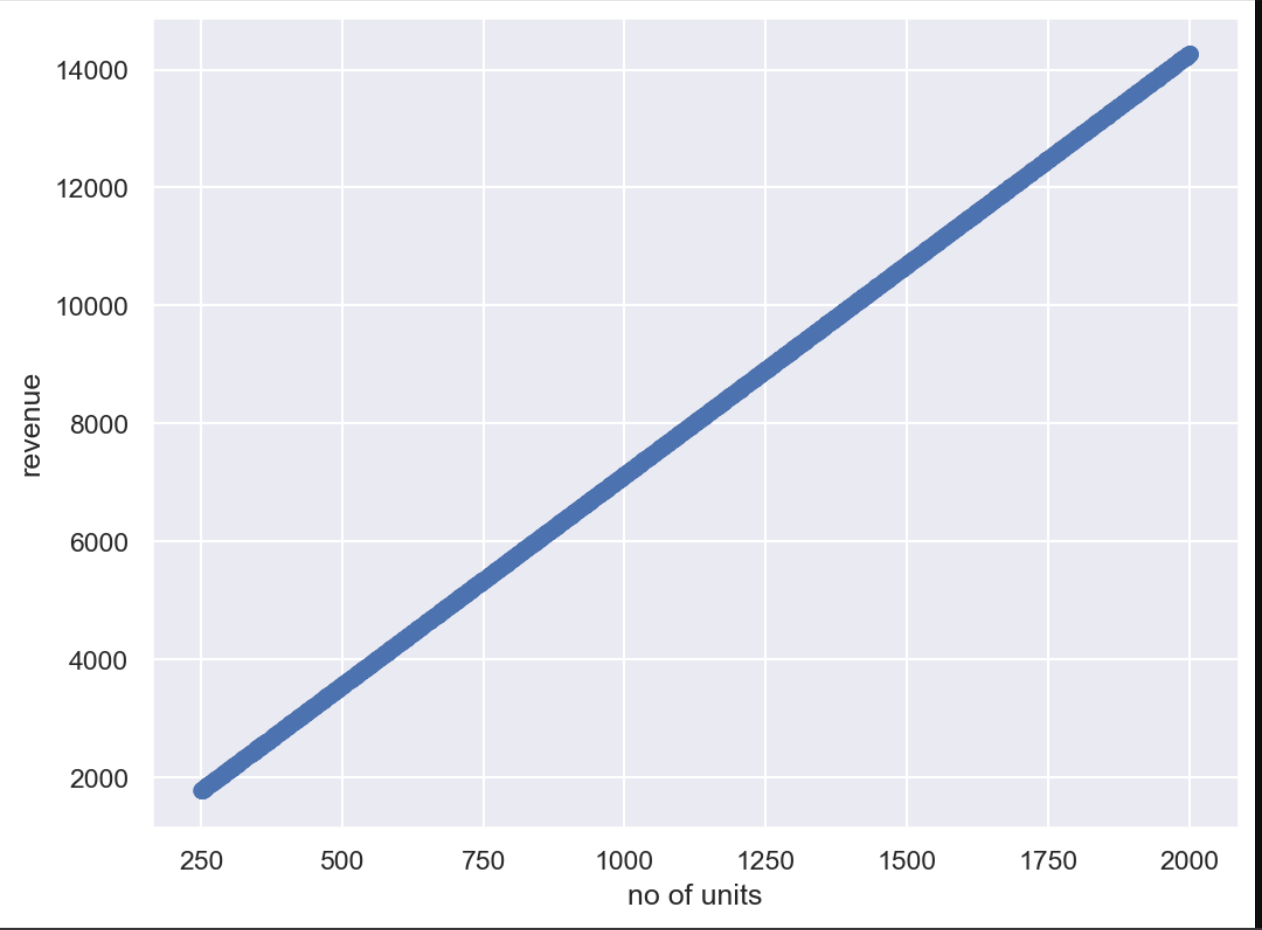
# Create labels for axises

plt.xlabel('no of units')

plt.ylabel('revenue')

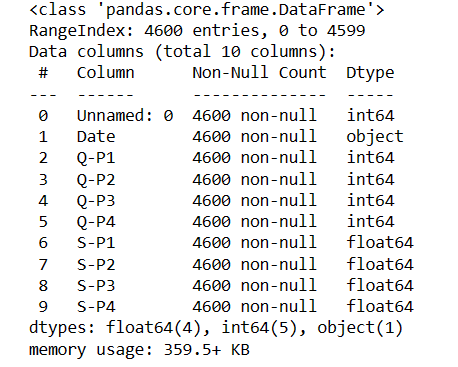
# Display the plot on the screen

plt.show()



SUMMARY OF OUR CLEANSED DATA:

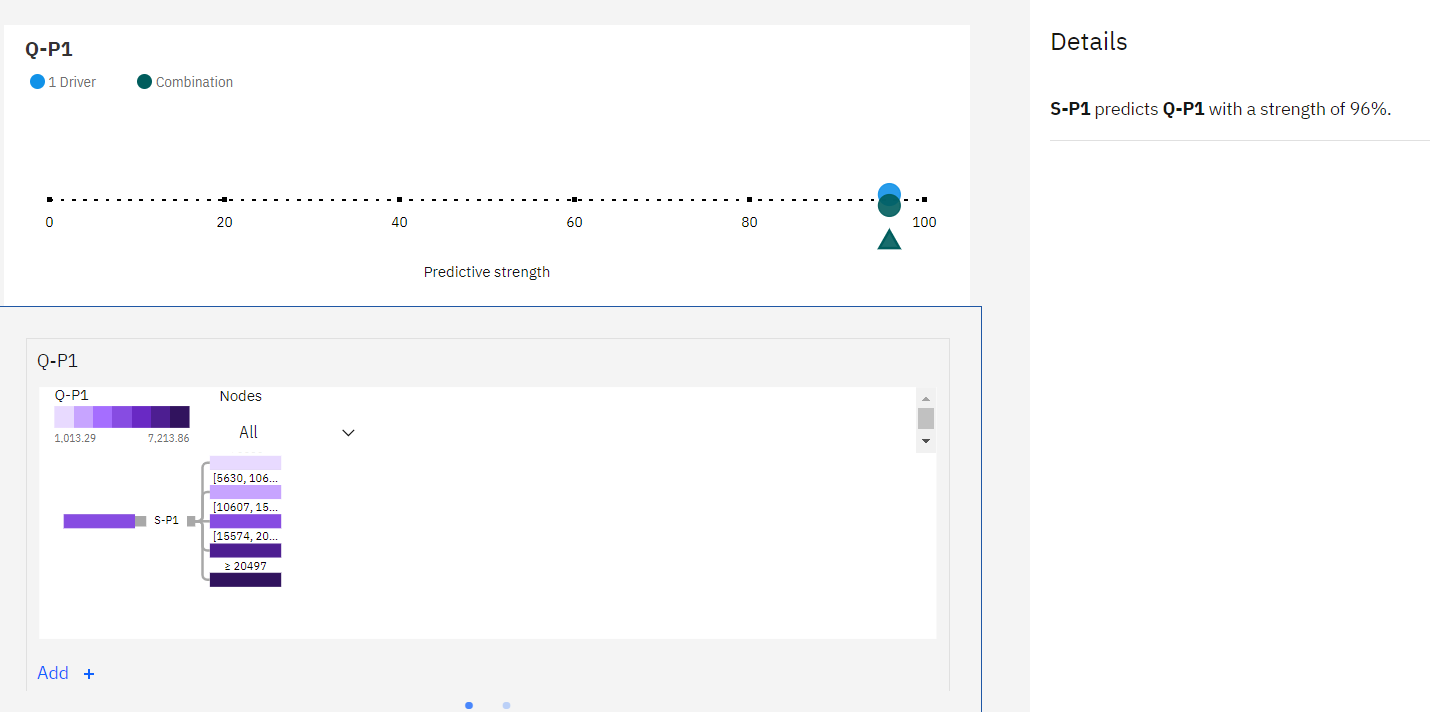
data.info()

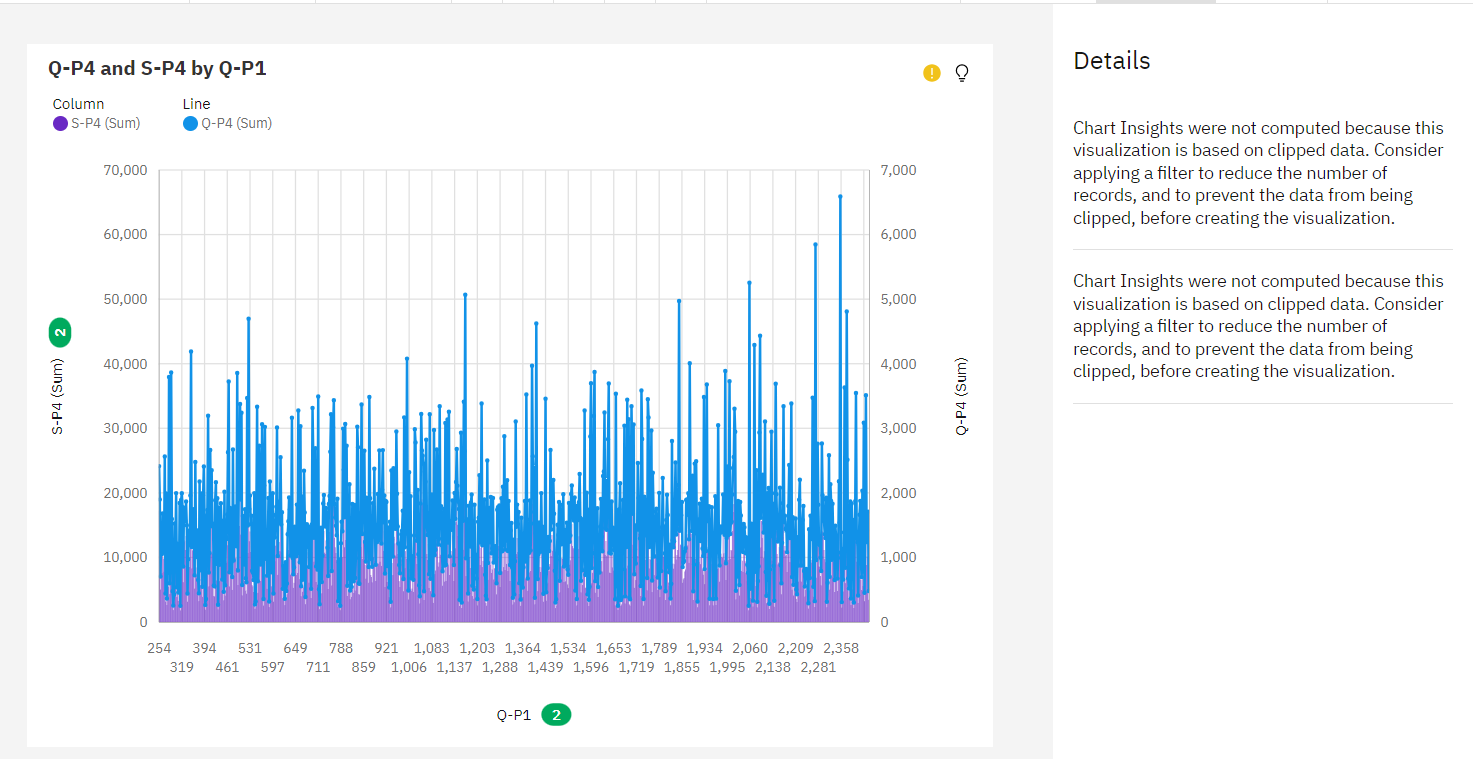


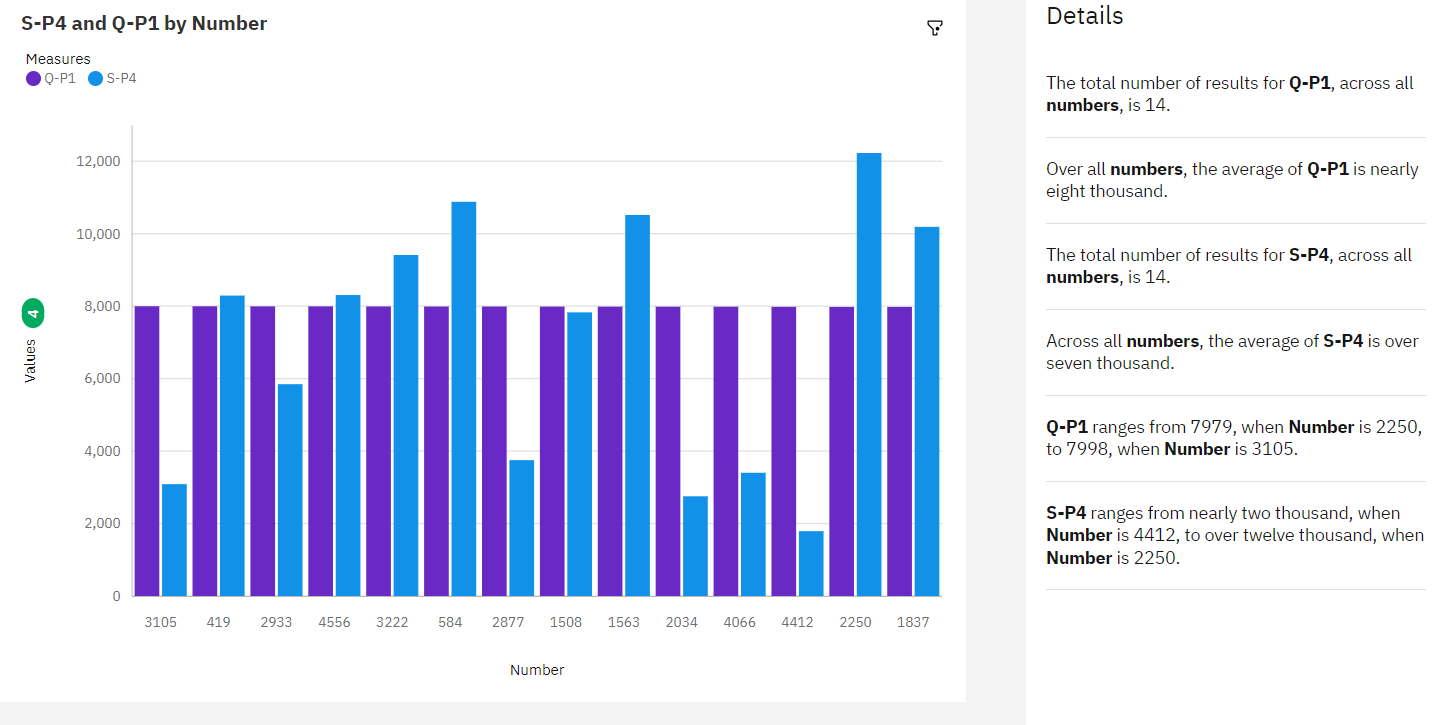
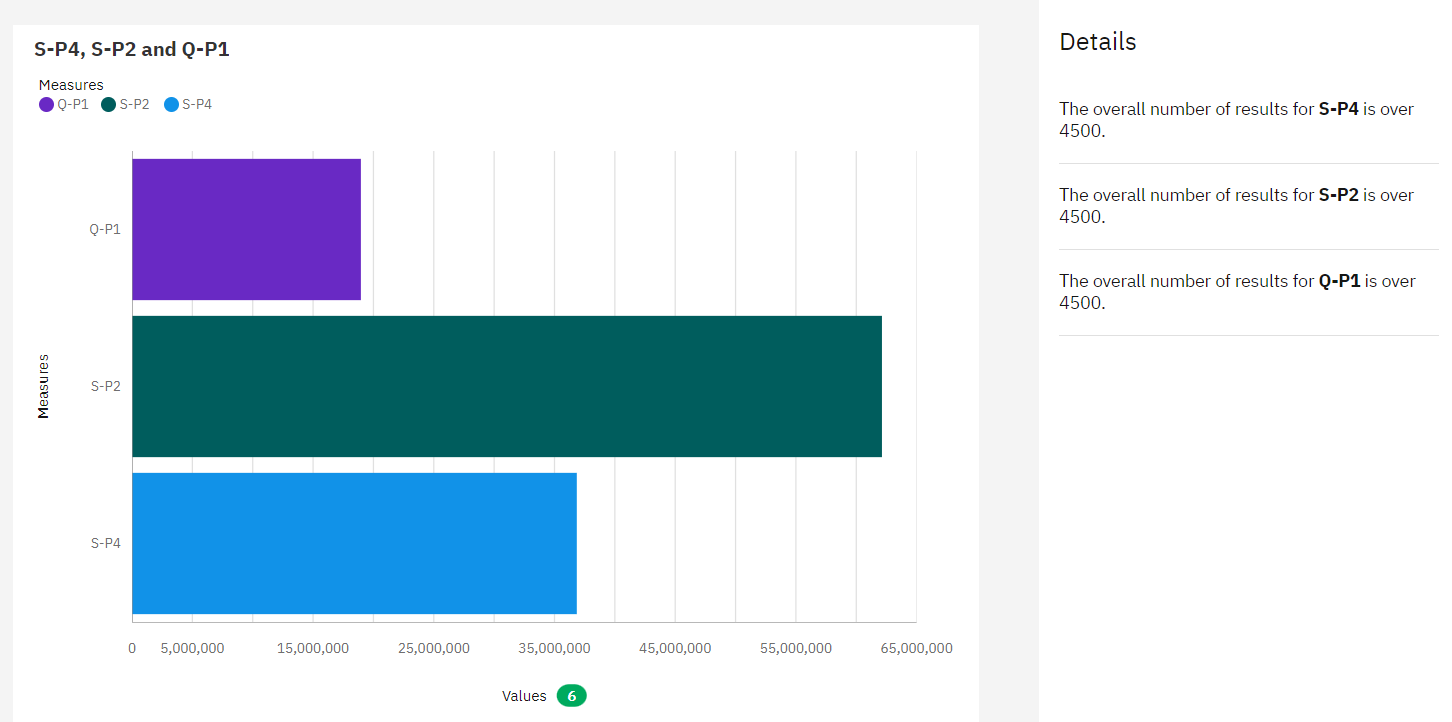
data.describe()

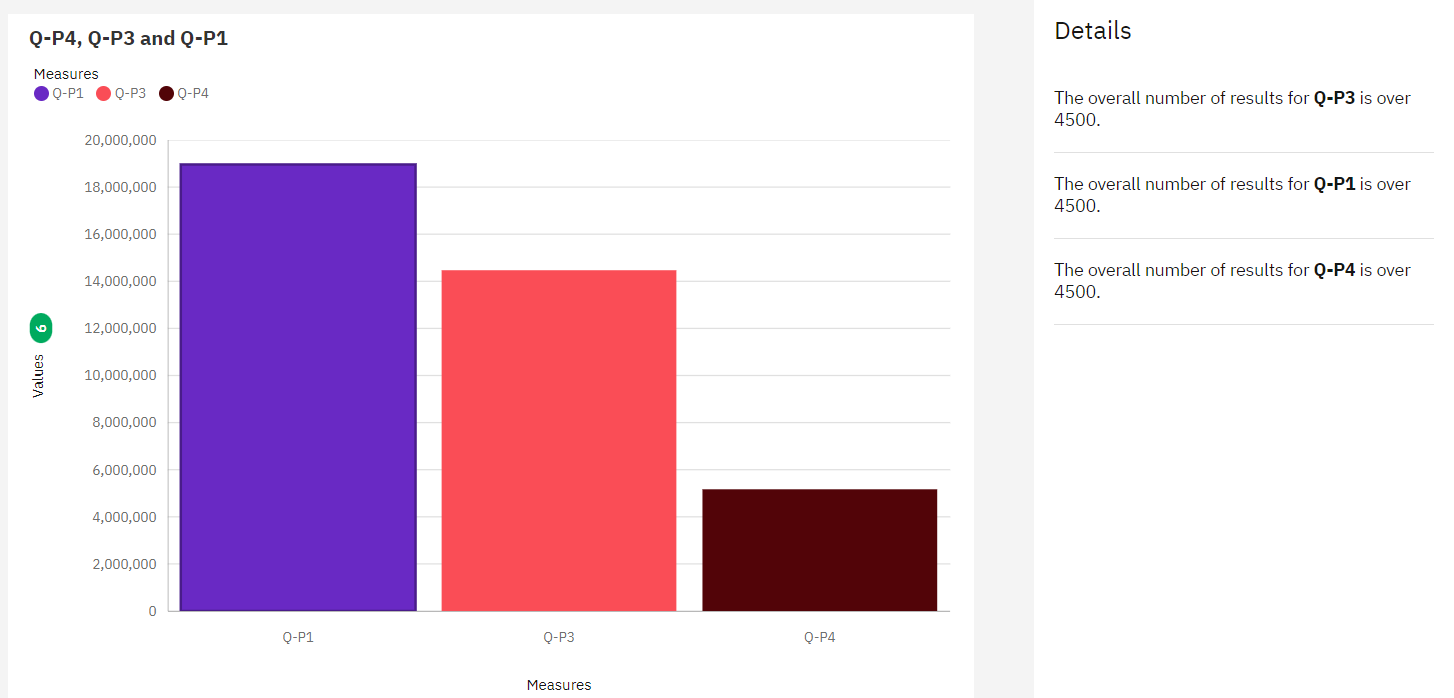
Accuracy of dataset performed by IBM cognos:

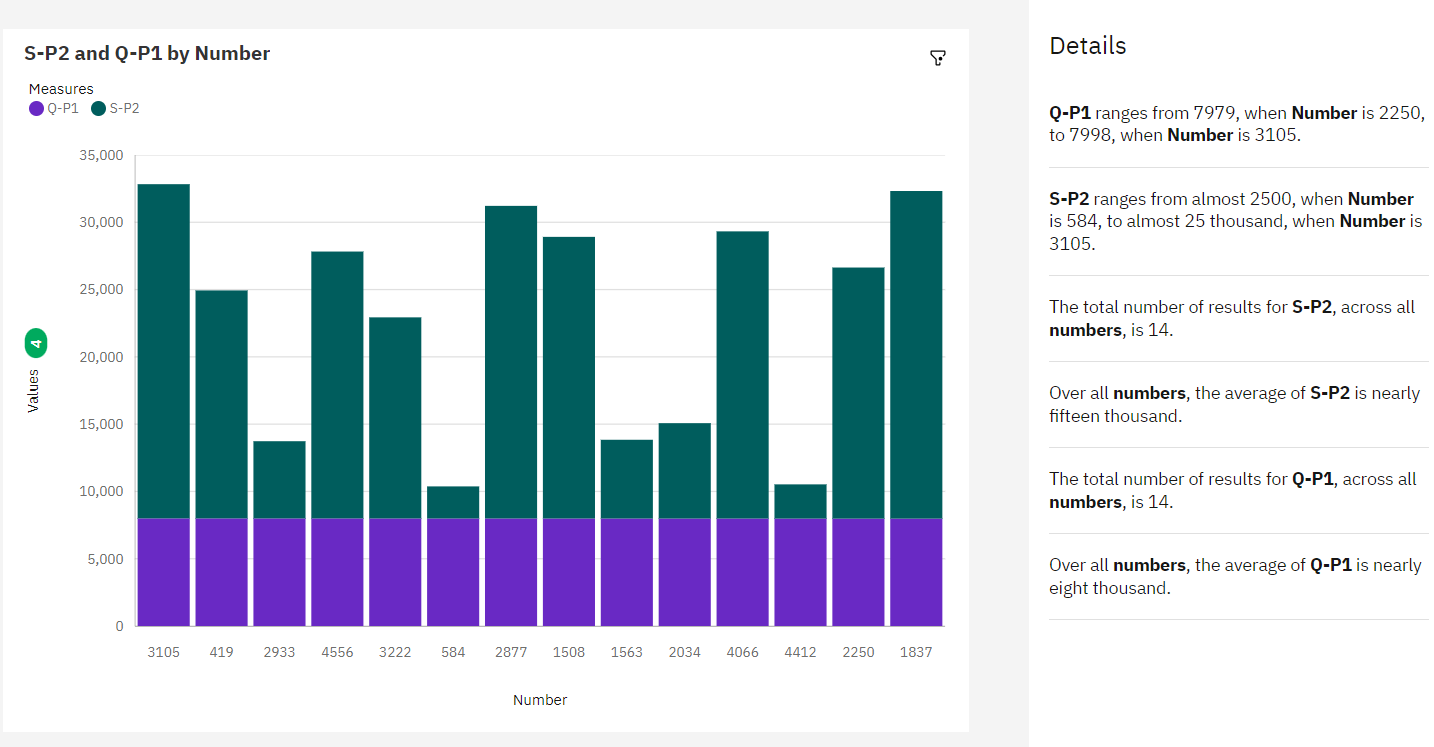
Q-P1





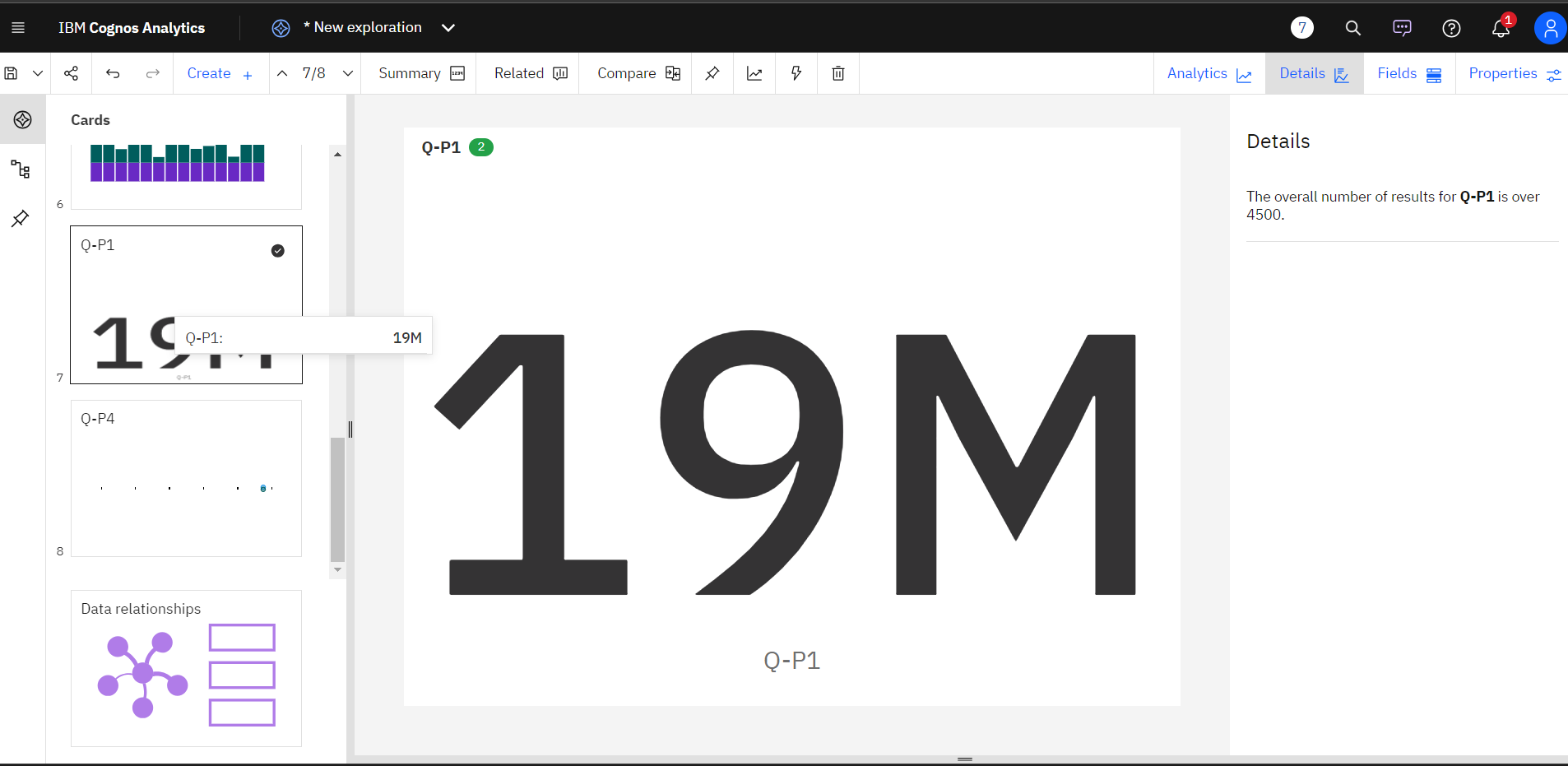




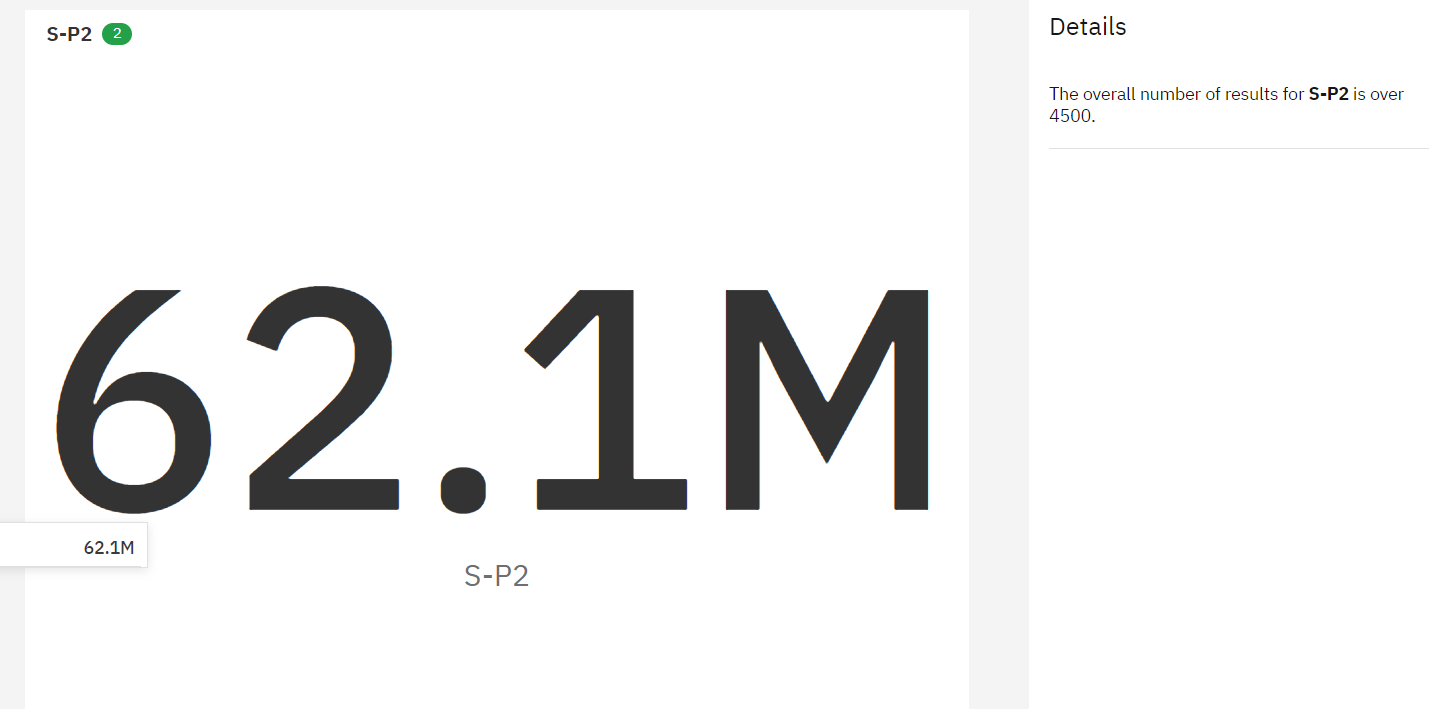


**OVERALL SUMMARY IN IBM COGNOS:**

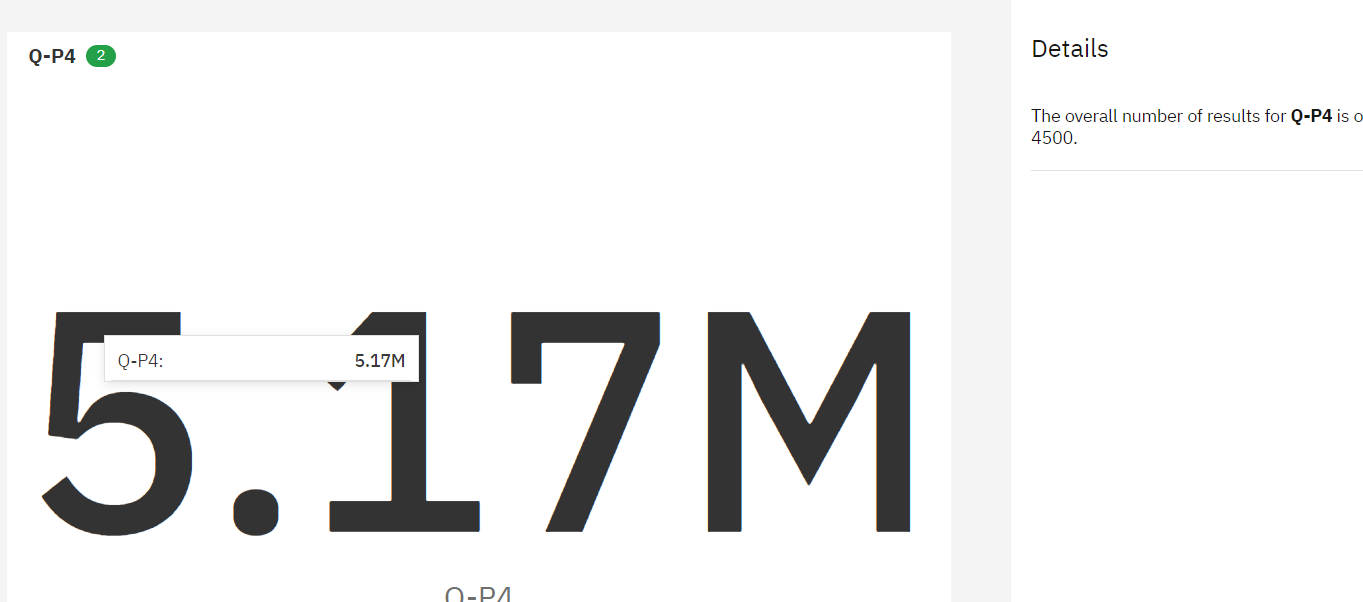
**Q-P1**

****

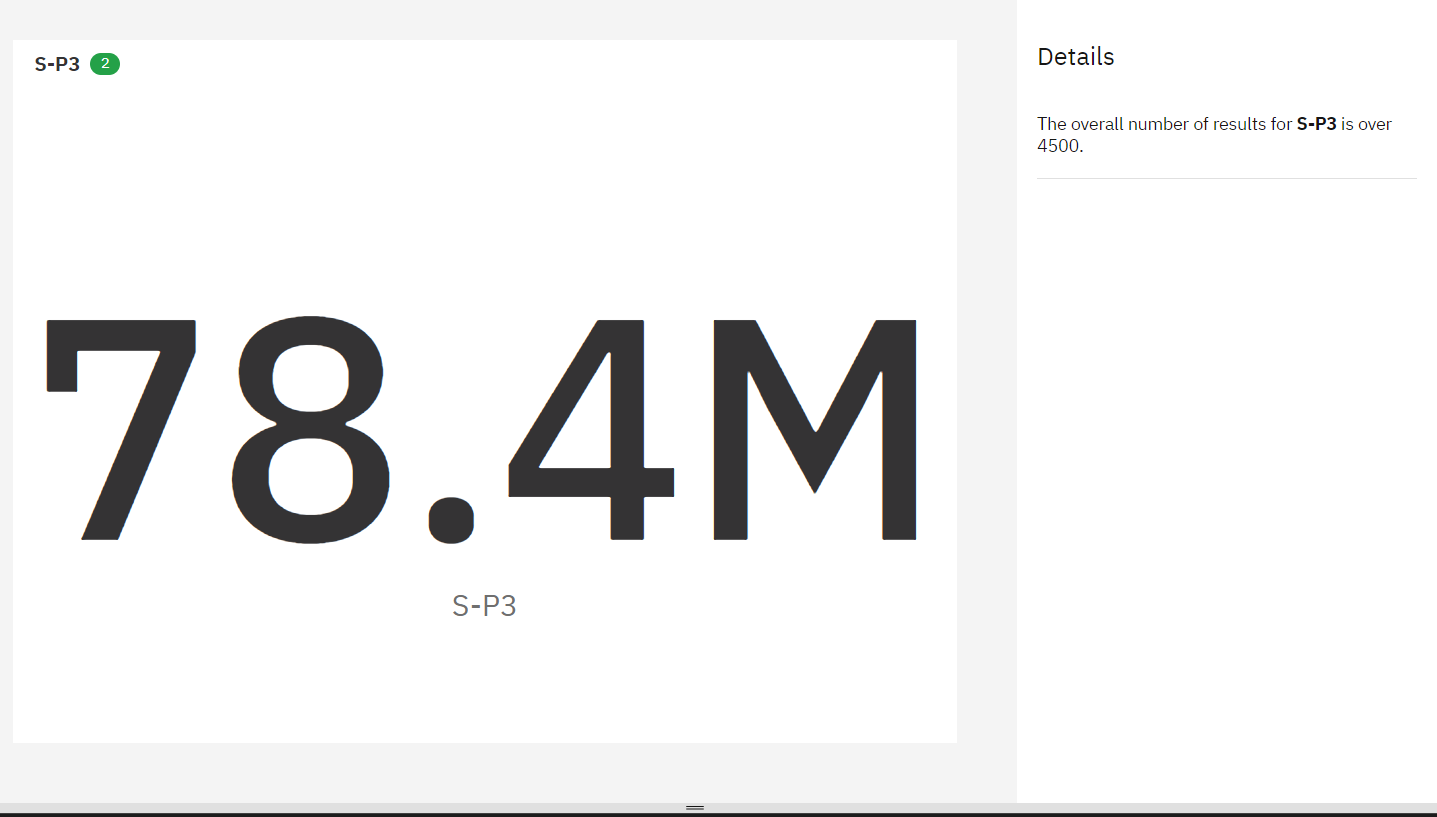
**Q-P2**

****

**Q-P3**

****

**Q-P4**

****

**CONCLUSION:**

In conclusion, cleaning and preprocessing a dataset are essential steps in the data analysis and machine learning process. These steps help ensure that your data is accurate, consistent, and ready for analysis or modeling. Clean, well-preprocessed data is the foundation for meaningful and actionable insights