PRODUCT SALES ANALYSIS

Date	26-10-2023
Team ID	715
Project Name	Product Sales Analysis

Table of Contents:

1. Introduction:

The project utilizes IBM Cognos to analyze sales data to identify top-selling products, peak sales periods, and customer preferences. The goal is to enhance inventory management and marketing strategies, reducing overstocking and customer dissatisfaction. The structured approach includes data collection, visualization design, and actionable insights, enabling businesses to make informed decisions and optimize operations.

2. Problem Statement:

The objectives of this project are to identify top-selling products, pinpoint peak sales periods, analyze customer preferences, optimize inventory management, and enhance marketing strategies to improve business performance. The project involves analyzing sales data using IBM Cognos, identifying top-selling products, peak sales periods, and customer preferences, and designing relevant visualizations.

3. Design and Innovation Strategies:

3.1. Define Analysis Objectives:

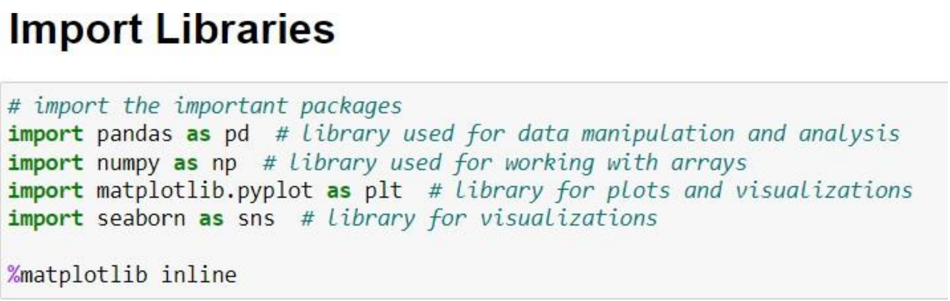
- Start by clearly defining the specific objectives of your analysis, which include identifying top-selling products, peak sales periods, and customer preferences. Having well-defined goals will guide your data collection and analysis.

3.2. Collect Sales Data:

- Gather the necessary sales data from a trusted source. This source can be your organization's sales database, data provider, or any other reliable data repository. Ensure that the data includes relevant fields, such as product information, sales dates, quantities sold, prices, and customer details.

3.3. Import Libraries:

- Start by importing the required libraries. In this case, you'll use Pandas for data manipulation.

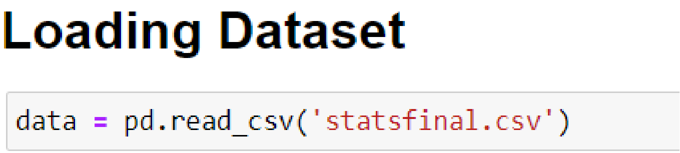


Import Libraries

```
# 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
```

3.4. Load the Dataset:

- This step involves loading your air quality dataset into your Python environment. The dataset should be in a format that Pandas can easily handle, such as a CSV file.



Loading Dataset

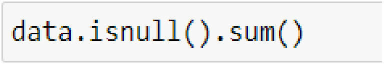
```
data = pd.read_csv('statsfinal.csv')
```

3.5. Data Preprocessing:

- Data preprocessing is crucial for ensuring the accuracy and reliability of your analysis. In this phase, you should:

Data Cleaning:

- Remove or handle missing values, duplicates, and any inconsistencies in the dataset. This step is essential for data integrity.



```
data.isnull().sum()
```

Data Transformation:

- Standardize and format data elements as needed. For instance, ensure that date and time formats are consistent and consider encoding categorical data into numerical values for analysis.

Data Integration:

- If your sales data is spread across multiple sources or tables, integrate them into a single dataset using common identifiers, like product IDs and customer IDs.

3.6. Data Visualization with IBM Cognos:

- ➢ IBM Cognos is a powerful tool for data visualization and reporting. Use it to create visually appealing and informative representations of your sales data. This phase includes:

  Loading Data:

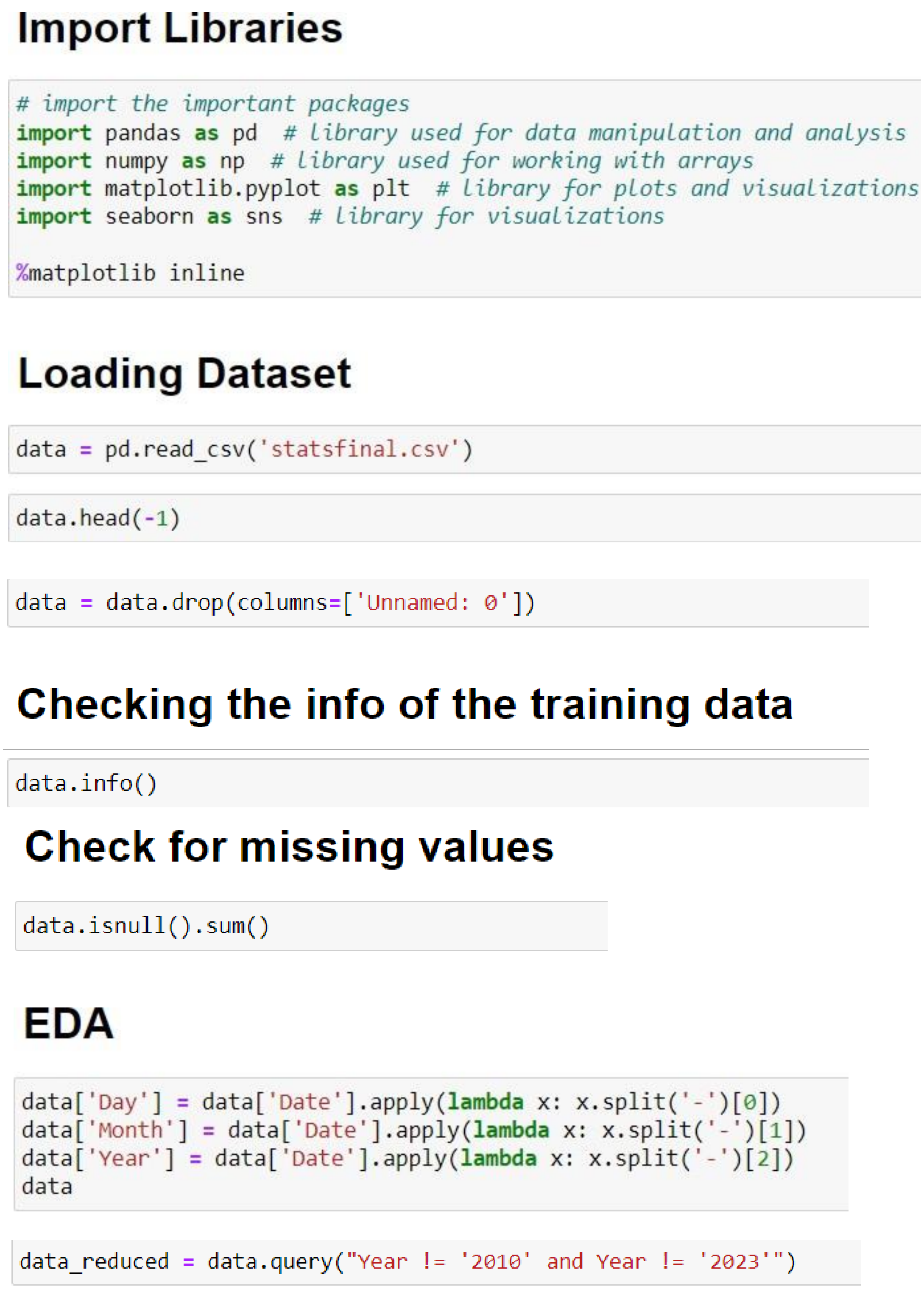
  - Import the preprocessed sales data into IBM Cognos.

  Creating Visualizations:

  - Use IBM Cognos to generate charts, graphs, dashboards, and reports to visualize the sales data. For instance, you can create bar charts to showcase top-selling products and line charts to display sales trends over time.

` **Interactive Exploration:**

  - To explore and analyze your data, use IBM Cognos' interactive features.
  - Apply filters and drill down into specific aspects of the data to gain insights.



Import Libraries

```
# 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
```

Loading Dataset

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

```
data.head(-1)
```

```
data = data.drop(columns=['Unnamed: 0'])
```

Checking the info of the training data

```
data.info()
```

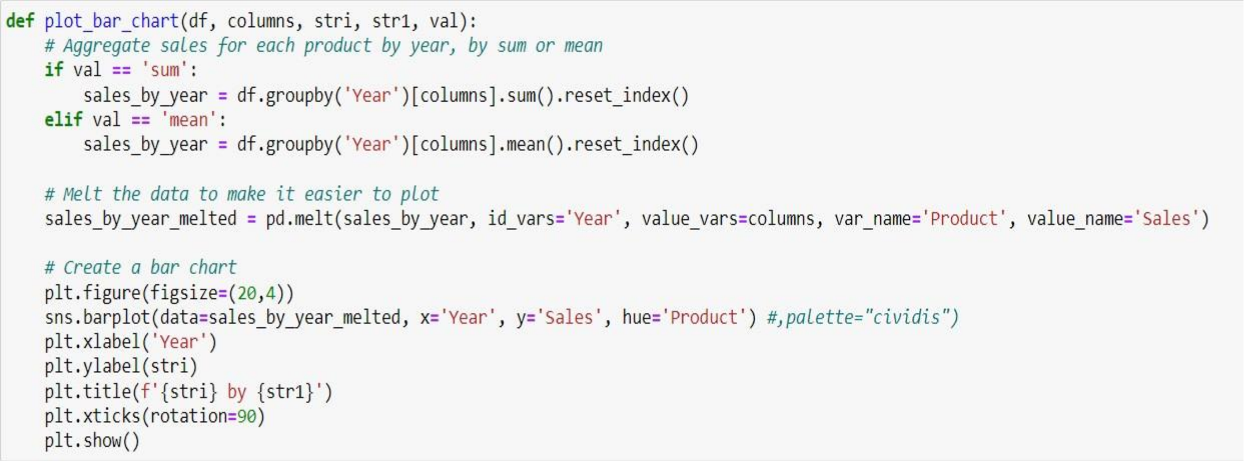
Check for missing values

```
data.isnull().sum()
```

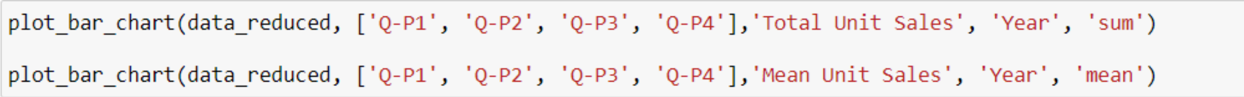
EDA

```
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
```

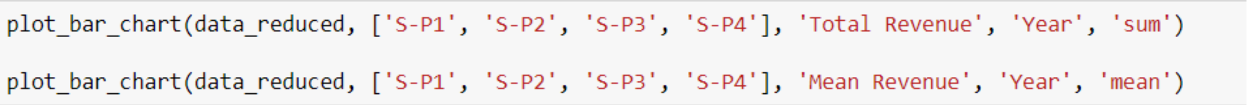
```
data_reduced = data.query("Year != '2010' and Year != '2023'")
```



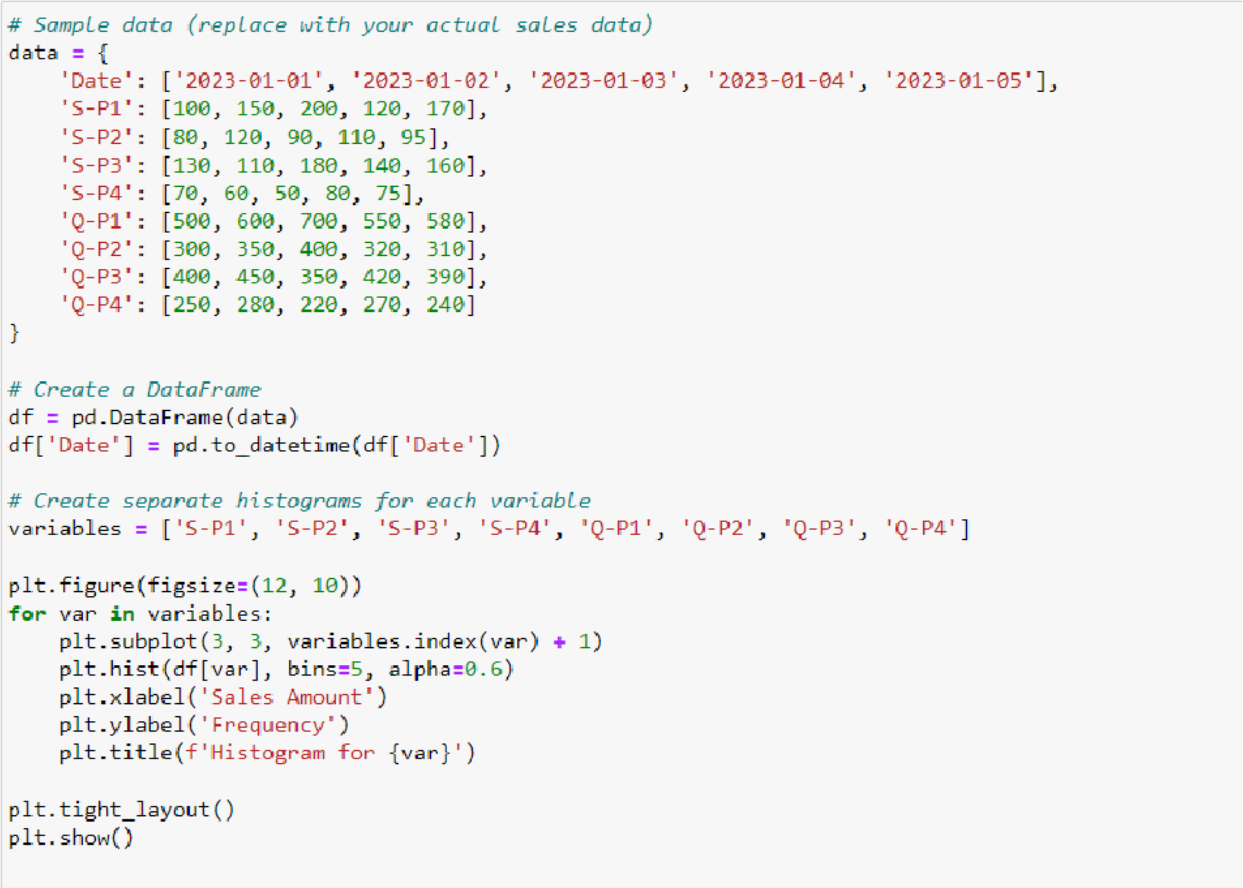
```python
def plot_bar_chart(df, columns, stri, str1, val):
    # Aggregate sales for each product by year, by sum or mean
    if val == 'sum':
        sales_by_year = df.groupby('Year')[columns].sum().reset_index()
    elif val == 'mean':
        sales_by_year = df.groupby('Year')[columns].mean().reset_index()

    # Melt the data to make it easier to plot
    sales_by_year_melted = pd.melt(sales_by_year, id_vars='Year', value_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') #,palette="cividis")
    plt.xlabel('Year')
    plt.ylabel(stri)
    plt.title(f'{stri} by {str1}')
    plt.xticks(rotation=90)
    plt.show()
```



```python
plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Total Unit Sales', 'Year', 'sum')

plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Mean Unit Sales', 'Year', 'mean')
```

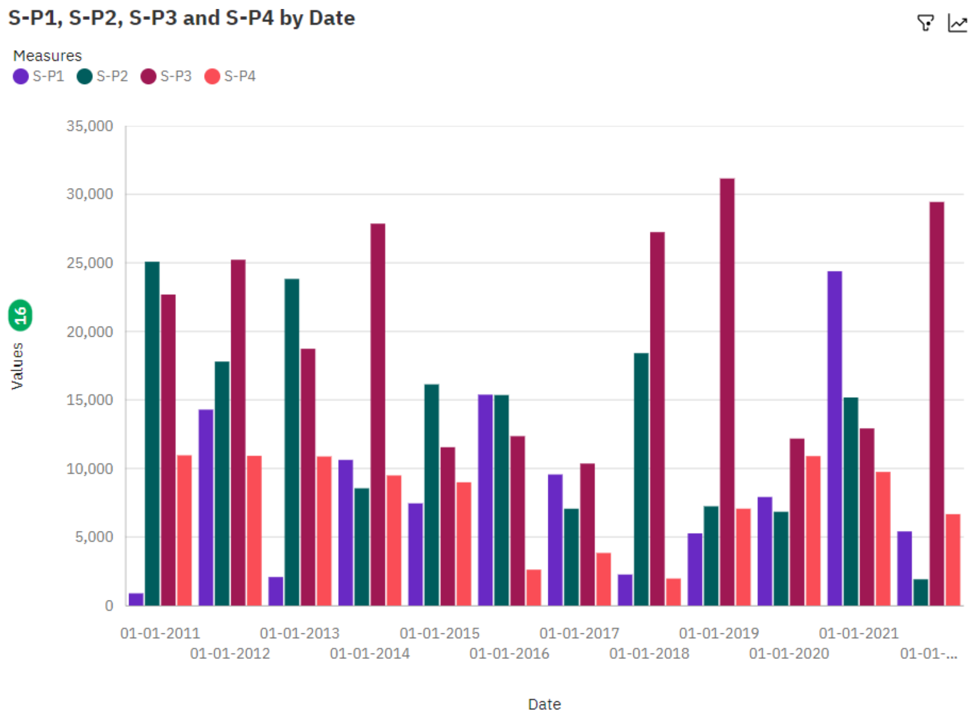


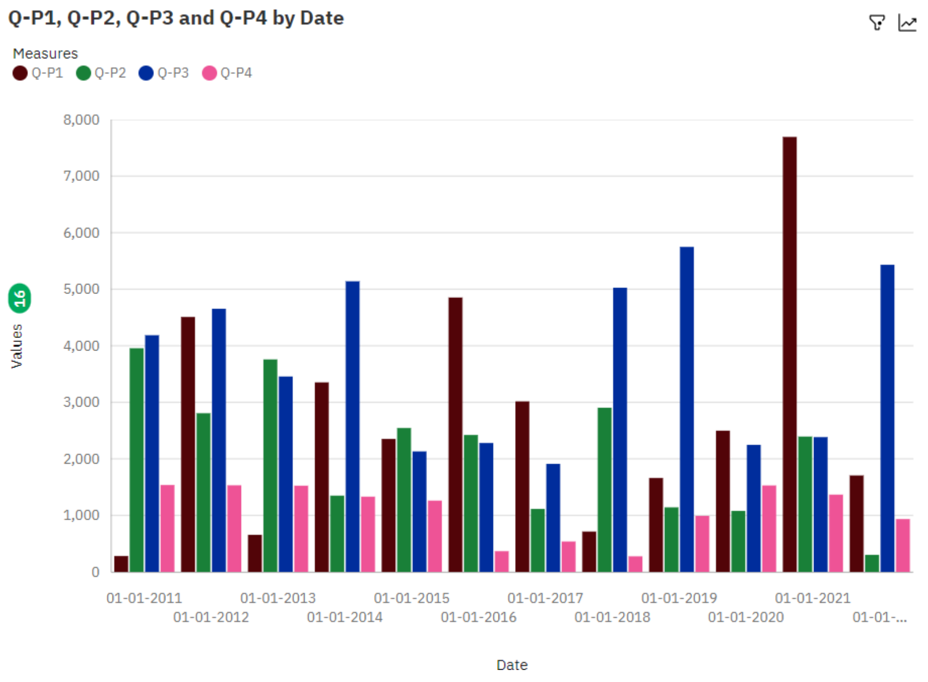
```python
plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Total Revenue', 'Year', 'sum')

plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Mean Revenue', 'Year', 'mean')
```



```python
# Sample data (replace with your actual sales data)
data = {
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04', '2023-01-05'],
    'S-P1': [100, 150, 200, 120, 170],
    'S-P2': [80, 120, 90, 110, 95],
    'S-P3': [130, 110, 180, 140, 160],
    'S-P4': [70, 60, 50, 80, 75],
    'Q-P1': [500, 600, 700, 550, 580],
    'Q-P2': [300, 350, 400, 320, 310],
    'Q-P3': [400, 450, 350, 420, 390],
    'Q-P4': [250, 280, 220, 270, 240]
}

# Create a DataFrame
df = pd.DataFrame(data)
df['Date'] = pd.to_datetime(df['Date'])

# Create separate histograms for each variable
variables = ['S-P1', 'S-P2', 'S-P3', 'S-P4', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']

plt.figure(figsize=(12, 10))
for var in variables:
    plt.subplot(3, 3, variables.index(var) + 1)
    plt.hist(df[var], bins=5, alpha=0.6)
    plt.xlabel('Sales Amount')
    plt.ylabel('Frequency')
    plt.title(f'Histogram for {var}')

plt.tight_layout()
plt.show()
```

Data Visualization with IBM Cognos:





Conclusion:

IBM Cognos was utilized for data analysis and visualization to improve inventory management and marketing strategies. The data was collected, preprocessed, and transformed into interactive dashboards. These dashboards revealed top-selling products, sales trends, and customer preferences, guiding inventory planning and marketing strategies. Understanding customer preferences through segmentation and visualizations was crucial for targeted marketing campaigns. The insights have tangible business impacts, including cost reduction and customer satisfaction improvement.