
CENTURYPLY®

TASK – 2

[1]:
```
#  Import Required Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
# Replace file paths with your actual paths or filenames
file_2021 = "Sales Data-Jan'21 to Dec'21.xlsx"
```

[3]:
```
# Replace file paths with your actual paths or filenames
file_2022_2023 = "Sales Data - Jan'22 to Mar'23.xlsx"
```

[4]:
```
# Replace file paths with your actual paths or filenames
file_2023_2025 = "Sales Data-Apr'23 to Jan'25.xlsx"
```

[5]:
```
# Define a function to load and clean each file
def load_sales_data(path):
    df = pd.read_excel(path)
    df = df.loc[:, ~df.columns.str.contains('^Unnamed')]
    return df
```

[6]:
```
# Load all three datasets
df1 = load_sales_data(file_2021)
```

[7]:
```
# Load all three datasets
df2 = load_sales_data(file_2022_2023)
```

[8]:
```
# Load all three datasets
df3 = load_sales_data(file_2023_2025)
```

---

[9]:
```
#  Combine the Datasets
combined_df = pd.concat([df1, df2, df3], ignore_index=True)
```

---

[10]:
```
# Rename Billed Qty (CBM) for easier access
combined_df.rename(columns={"Billed Qty (CBM)": "Billed_Qty(CBM)"}, inplace=True)
```

---

[11]:
```
# Convert Date column to datetime
combined_df["Date"] = pd.to_datetime(combined_df["Date"], errors="coerce")
```

---

[12]:
```
# Extract year
combined_df["Year"] = combined_df["Date"].dt.year
```

---

1. DETAILS OF THE DATASET :

[13]:
```
# 1. Provide Details of the Combined Dataset

print("Dataset Shape:", combined_df.shape)
```

```
Dataset Shape: (488231, 39)
```

[14]:
```
print("\n Columns:\n", combined_df.columns.tolist())
```

```
 Columns:
 ['Customer Name', 'Customer Code', 'Ship To Cust Code', 'Ship To Cust Name', 'Bill Type', 'C', 'Inv.No', 'ODN', 'BUS. PLACE', 'SO.No', 'Date', 'Week',
'Incoterms', 'PO.No', 'PI.No', 'Cust.Nature', 'Territory Code', 'Territory', 'Inv.Date', 'Brand', 'Sub-Brand', 'Prime Brand', 'Folder/Non-Folder', 'Mat
erial', 'Material Desc.', 'Division', 'Division Text', 'Thickness', 'Vol.CBM', 'Design(LAM)', 'Finish(LAM)', 'Billed Qty', 'Sales unit', 'Billed Qty(SQ
M)', 'Billed Qty(PCS)', 'Billed Qty(NA)', 'Billed_Qty(CBM)', 'Billed Qty (MT)', 'Year']
```

[15]:
```
print("\n Sample Records:\n", combined_df.head())
```

```
 Sample Records:
   Customer Name  Customer Code  Ship To Cust Code Ship To Cust Name Bill Type  \
0  N.K. Traders     10016196.0         10016196.0      N.K. Traders      ZFAC
1  N.K. Traders     10016196.0         10016196.0      N.K. Traders      ZFAC
2  N.K. Traders     10016196.0         10016196.0      N.K. Traders      ZFAC
3  N.K. Traders     10016196.0         10016196.0      N.K. Traders      ZFAC
4  N.K. Traders     10016196.0         10016196.0      N.K. Traders      ZFAC

     C        Inv.No           ODN BUS. PLACE         SO.No  ... Design(LAM)  \
0  NaN  2.027009e+09  F22027008891       PH01  1.129146e+09  ...         NaN
1  NaN  2.027009e+09  F22027008891       PH01  1.129146e+09  ...         NaN
2  NaN  2.027009e+09  F22027008891       PH01  1.129146e+09  ...         NaN
3  NaN  2.027009e+09  F22027008891       PH01  1.129146e+09  ...         NaN
4  NaN  2.027009e+09  F22027008891       PH01  1.129146e+09  ...         NaN

   Finish(LAM) Billed Qty  Sales unit  Billed Qty(SQM) Billed Qty(PCS)  \
0          NaN      825.0          PC          460.515           825.0
1          NaN     1131.0          PC          631.324          1131.0
2          NaN      610.0          PC          737.856           610.0
3          NaN      357.0          PC          398.519           357.0
4          NaN       70.0          PC           39.074            70.0

   Billed Qty(NA) Billed_Qty(CBM) Billed Qty (MT)    Year
0            0.0           2.533           2.003  2021.0
1            0.0           4.419           3.491  2021.0
2            0.0           8.116           0.000  2021.0
3            0.0           6.675           5.272  2021.0
4            0.0           0.654           0.517  2021.0

[5 rows x 39 columns]
```

[16]:
```
print("\n Null Values:\n", combined_df.isnull().sum())
```

```
 Null Values:
 Customer Name         293296
Customer Code          293296
Ship To Cust Code      293296
Ship To Cust Name      293296
Bill Type              293296
C                      485706
Inv.No                 293296
ODN                    293296
BUS. PLACE             293296
SO.No                  293296
Date                   293296
Week                   293296
Incoterms              293296
PO.No                  488231
PI.No                  488231
Cust.Nature            293296
Territory Code         293296
Territory              293296
Inv.Date               293296
Brand                  488231
Sub-Brand              488231
Prime Brand            488231
Folder/Non-Folder      488231
Material               293296
Material Desc.         293296
Division               293296
Division Text          293296
Thickness              293296
Vol.CBM                293296
Design(LAM)            488231
Finish(LAM)            488231
Billed Qty             293296
```

```
Sales unit            293296
Billed Qty(SQM)       293296
Billed Qty(PCS)       293296
Billed Qty(NA)        293296
Billed_Qty(CBM)       293296
Billed Qty (MT)       293296
Year                  293296
dtype: int64
```

[17]:
```
print("\n Year Range in Data:", combined_df["Year"].min(), "to", combined_df["Year"].max())
```

```
 Year Range in Data: 2020.0 to 2023.0
```

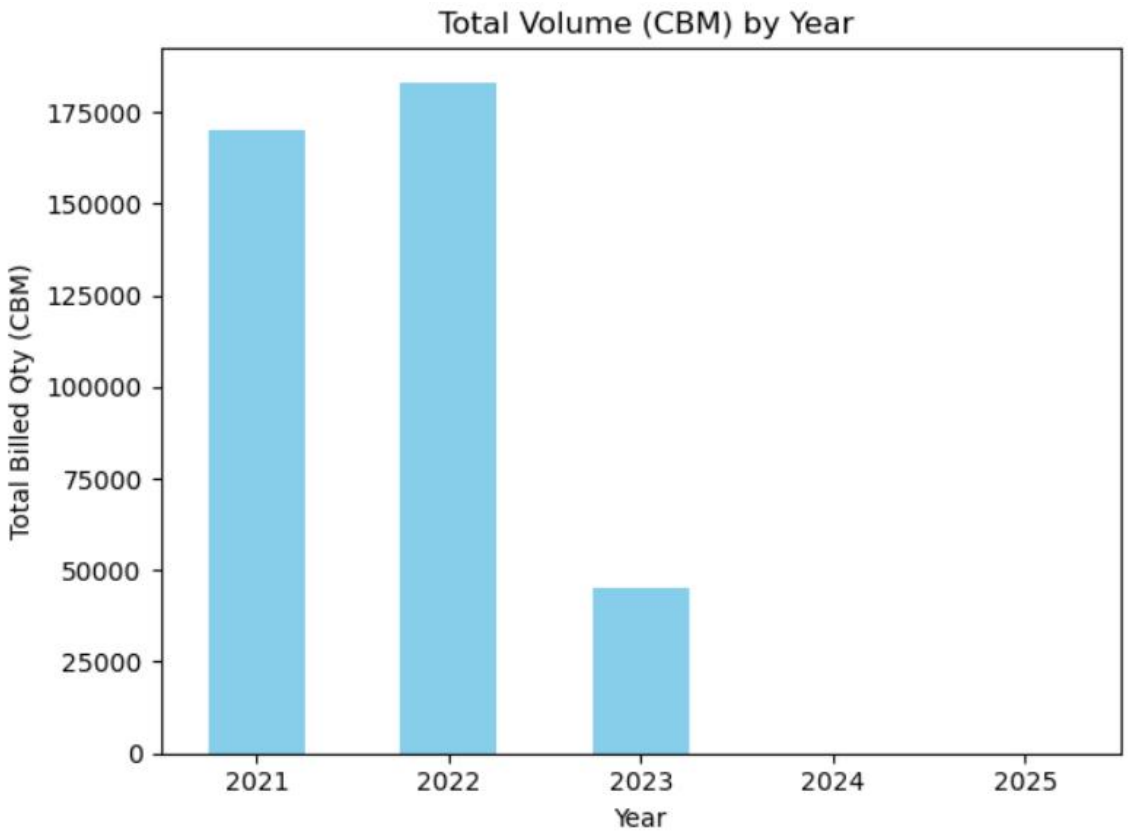
[18]:
```
print(combined_df[combined_df["Year"].isin([2024, 2025])][["Year", "Billed_Qty(CBM)"]].describe())

# Since the count is 0 , that's why they don't show in the plot.
```

```
       Year  Billed_Qty(CBM)
count   0.0              0.0
mean    NaN              NaN
std     NaN              NaN
min     NaN              NaN
25%     NaN              NaN
50%     NaN              NaN
75%     NaN              NaN
max     NaN              NaN
```

2. Volume-wise Analysis using "Billed_Qty(CBM)" :

[19]:
```
# 2. Volume-wise Analysis using "Billed_Qty(CBM)"

# Total volume per year

volume_by_year = combined_df.groupby("Year")["Billed_Qty(CBM)"].sum()
volume_by_year = volume_by_year.reindex([2021, 2022, 2023, 2024, 2025], fill_value=0)
```

[20]:
```
# Plot

volume_by_year.plot(kind='bar', title=" Total Volume (CBM) by Year", color='skyblue')
plt.xlabel("Year")
plt.ylabel("Total Billed Qty (CBM)")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```




```
[21]: print(volume_by_year)
```

```
Year
2021    170307.668
2022    183153.078
2023     45220.005
2024         0.000
2025         0.000
Name: Billed_Qty(CBM), dtype: float64
```

3. Extract Primary Columns for Focused Analysis :

[22]:
```
# 3. Extract Primary Columns for Focused Analysis

# Create filtered dataframe with primary columns

primary_df = combined_df[["Date", "Material", "Thickness", "Bill Type", "Billed_Qty(CBM)", "Year", "Territory"]].copy()
```

[23]:
```
# Preview

print(primary_df.head())
```

```
        Date            Material  Thickness Bill Type  Billed_Qty(CBM)  \
0 2021-01-01  HMCI055PC183003050       5.50      ZFAC            2.533
1 2021-01-01  HMCI070PC183003050       7.00      ZFAC            4.419
2 2021-01-01  HMCI110PC198306100      11.00      ZFAC            8.116
3 2021-01-01  HMCI168PC183006100      16.75      ZFAC            6.675
4 2021-01-01  HMCI168PC183003050      16.75      ZFAC            0.654

     Year       Territory
0  2021.0  Prelam -Jaipur
1  2021.0  Prelam -Jaipur
2  2021.0  Prelam -Jaipur
3  2021.0  Prelam -Jaipur
4  2021.0  Prelam -Jaipur
```

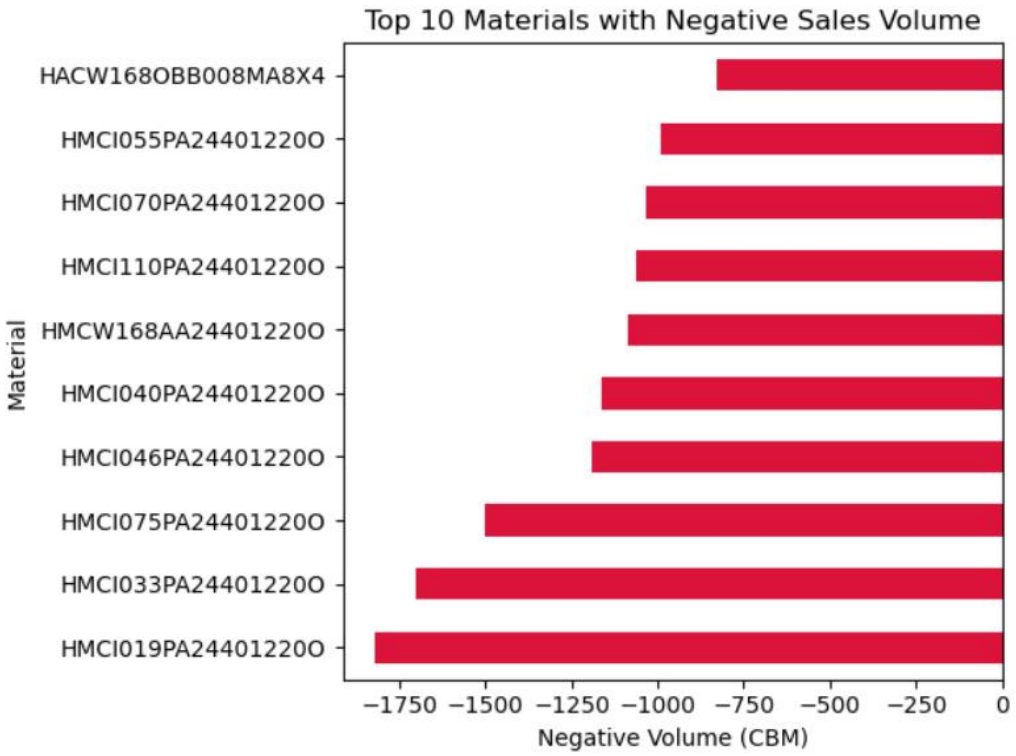
4. Analysis of Negative Sales (Billed Volume < 0) :

```
# 4. Analysis of Negative Sales (Billed Volume < 0)

# Filter negative sales

negative_sales = primary_df[primary_df["Billed_Qty(CBM)"] < 0]
```

```
# Summary by Year and Material

neg_summary = negative_sales.groupby(["Year", "Material"])["Billed_Qty(CBM)"].sum().reset_index()
```

```
# Top 10 materials with highest negative volume

top_negative = negative_sales.groupby("Material")["Billed_Qty(CBM)"].sum().sort_values().head(10)
```

```
# Plot

top_negative.plot(kind='barh', color='crimson', title='Top 10 Materials with Negative Sales Volume')
plt.xlabel("Negative Volume (CBM)")
plt.tight_layout()
plt.show()
```



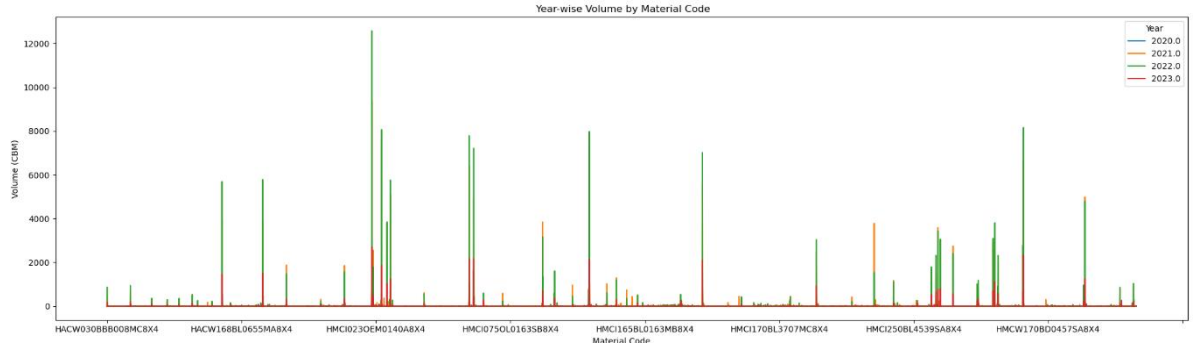

```
print("\n Negative Sales Summary (Year + Material):\n", neg_summary.head(10))
```

```
 Negative Sales Summary (Year + Material):
      Year            Material  Billed_Qty(CBM)
0   2020.0  HACW055OBB008MB8X4           -0.491
1   2020.0  HACW080BBB008MA8X4           -0.476
2   2020.0  HACW080BBB008MB8X4           -0.643
3   2020.0  HACW080BBB008SB8X4           -0.119
4   2020.0  HACW080BL0111MB8X4           -0.095
5   2020.0  HACW080BL0238MB8X4           -0.024
6   2020.0  HACW080BL4535MB8X4           -0.024
7   2020.0  HACW080BL4854MB8X4           -0.024
8   2020.0  HACW080OBB008MA8X4           -0.119
9   2020.0  HACW080OBB008SB8X4           -0.071
```

5. Year-wise Patterns: Volume, Material Codes, and Territories :

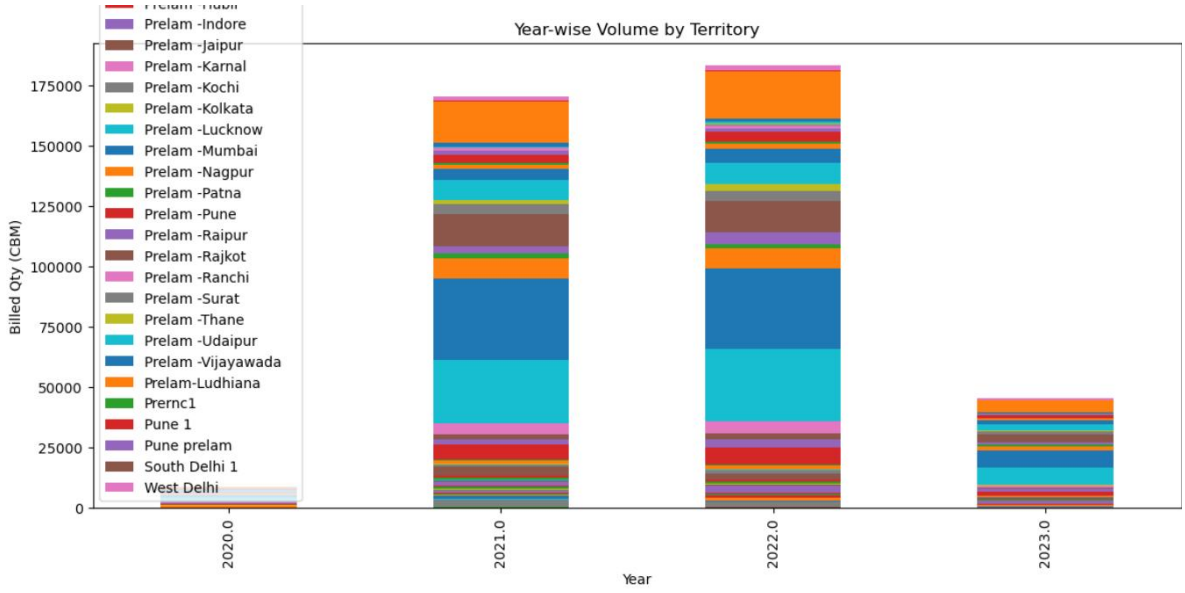
```
# 5. Year-wise Patterns: Volume, Material Codes, and Territories

#  Volume by Material Code by Year

material_yearly = combined_df.groupby(["Year", "Material"])["Billed_Qty(CBM)"].sum().unstack().fillna(0)
```

```
# Plot trends

material_yearly.T.plot(figsize=(20, 6), title=" Year-wise Volume by Material Code")
plt.ylabel("Volume (CBM)")
plt.xlabel("Material Code")
plt.tight_layout()
plt.show()
```



```python
#  Sales by Territory per Year

# Make sure 'Territory' column exists

if "Territory" in combined_df.columns:
    territory_yearly = combined_df.groupby(["Year", "Territory"])["Billed_Qty(CBM)"].sum().unstack().fillna(0)

# Plot stacked bar chart

    territory_yearly.plot(kind="bar", stacked=True, figsize=(14, 6), title="Year-wise Volume by Territory")
    plt.ylabel("Billed Qty (CBM)")
    plt.xlabel("Year")
    plt.tight_layout()
    plt.show()
else:
    print("'Territory' column not found in dataset.")
```


Prechn1
Prechn2
Precmb1
Precmb2
Prehyd1
Prejap1
Prekoch1
Prekoch2
Prekoch3
Prekol1
Prekol2
Prekol3
Prelam
Prelam -Ahmedabad
Prelam -Bangalore
Prelam -Bhuvneshwar
Prelam -Chennai
Prelam -Coimbatore
Prelam -Dehradun
Prelam -Delhi
Prelam -Ghaziabad
Prelam -Gurgaon
Prelam -Guwahati
Prelam -Hubli


Year-wise Volume by Territory
Billed Qty (CBM)
Year
0
25000
50000
75000
100000
125000
150000
175000
2020.0
2021.0
2022.0
2023.0
Prelam -Indore
Prelam -Jaipur
Prelam -Karnal
Prelam -Kochi
Prelam -Kolkata
Prelam -Lucknow
Prelam -Mumbai
Prelam -Nagpur
Prelam -Patna
Prelam -Pune
Prelam -Raipur
Prelam -Rajkot
Prelam -Ranchi
Prelam -Surat
Prelam -Thane
Prelam -Udaipur
Prelam -Vijayawada
Prelam-Ludhiana
Prernc1
Pune 1
Pune prelam
South Delhi 1
West Delhi

6. Material-wise Analysis :

```
# 6. Material-wise Analysis

# number of transactions

material_analysis = combined_df.groupby("Material").agg({"Billed_Qty(CBM)": "sum",
"Thickness": "mean","Date": "count"  }).rename(columns={"Billed_Qty(CBM)": "Total_Volume(CBM)",
"Date": "Transaction_Count", "Thickness": "Avg_Thickness"})
```

```
# Sort and display top 10

top_materials = material_analysis.sort_values("Total_Volume(CBM)", ascending=False).head(10)
print("Top 10 Materials by Total Volume:\n", top_materials)
```

```
Top 10 Materials by Total Volume:
                    Total_Volume(CBM)  Avg_Thickness  Transaction_Count
Material
HMCI019PA24401220O           25175.034           1.90               8144
HMCI110PA24401220O           17768.908          11.00               6479
HMCW168AA24401220O           17509.452          16.75               6008
HMCI033PA24401220O           17456.305           3.30               7211
HMCI055PA24401220O           16587.466           5.50               6438
HMCI165PA24401220O           14675.541          16.50               4908
HMCI070PA24401220O           14521.219           7.00               4252
HMCI046PA24401220O           12246.225           4.60               4728
HMCW180AA24401220O           11262.908          18.00               4233
HACW168OBB008MA8X4           11216.429          16.75               4405
```