```
# Importing all necessary libraries

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Loading the dataset
df = pd.read_excel('..\\files\\e-comdata-set1.xlsx')


fact_table = pd.read_excel("..\\files\\e-comdata-set1.xlsx",
sheet_name = "Fact_table", engine='openpyxl')
trans_dim = pd.read_excel("..\\files\\e-comdata-set1.xlsx", sheet_name
= "Trans_dim", engine='openpyxl')
item_dim = pd.read_excel("..\\files\\e-comdata-set1.xlsx", sheet_name
= "Item_dim", engine='openpyxl')
customer_dim = pd.read_excel("..\\files\\e-comdata-set1.xlsx",
sheet_name = "Customer_dim", engine='openpyxl')
time_dim = pd.read_excel("..\\files\\e-comdata-set1.xlsx", sheet_name
= "Time_dim", engine='openpyxl')
store_dim =  pd.read_excel("..\\files\\e-comdata-set1.xlsx",
sheet_name = "Store_dim", engine='openpyxl')

print("data has been loaded successfully!!")
```

```
data has been loaded successfully!!
```

Load all of the dimension tables from e-comdata-set1.xlsx into the pandas dataframe.

a. Calculate the mean value of the total price of the fact table.

b. Calculate the standard deviation of all columns of the fact table.

c. Find the most common unit in the fact table.

```
# Q-a: Calculate the mean value of the total price of the fact table.

fact_table_mean = fact_table['total_price'].mean()
print(f'The mean value of the total price of the fact table is:
{fact_table_mean}')
```

```
The mean value of the total price of the fact table is: 293.590325
```

```
# Q-b: Calculate the standard deviation of all columns of the fact
table.

fact_table_std = fact_table.select_dtypes(include=['float',
'int']).std()
print(f'The standard deviation of all columns of the fact table is: \
n{fact_table_std}')
```

```
The standard deviation of all columns of the fact table is:
quantity         9.833300
unit_price      10.847940
total_price    272.961678
dtype: float64
```

```
# Q-c: Find the most common unit in the fact table.
common_unit = fact_table['unit'].mode().iloc[0]
unit_counts = df['unit'].value_counts().loc['cans']
print(f'The most common unit is: {common_unit}. Cause it\'s total row
count is {unit_counts} rows.')
```

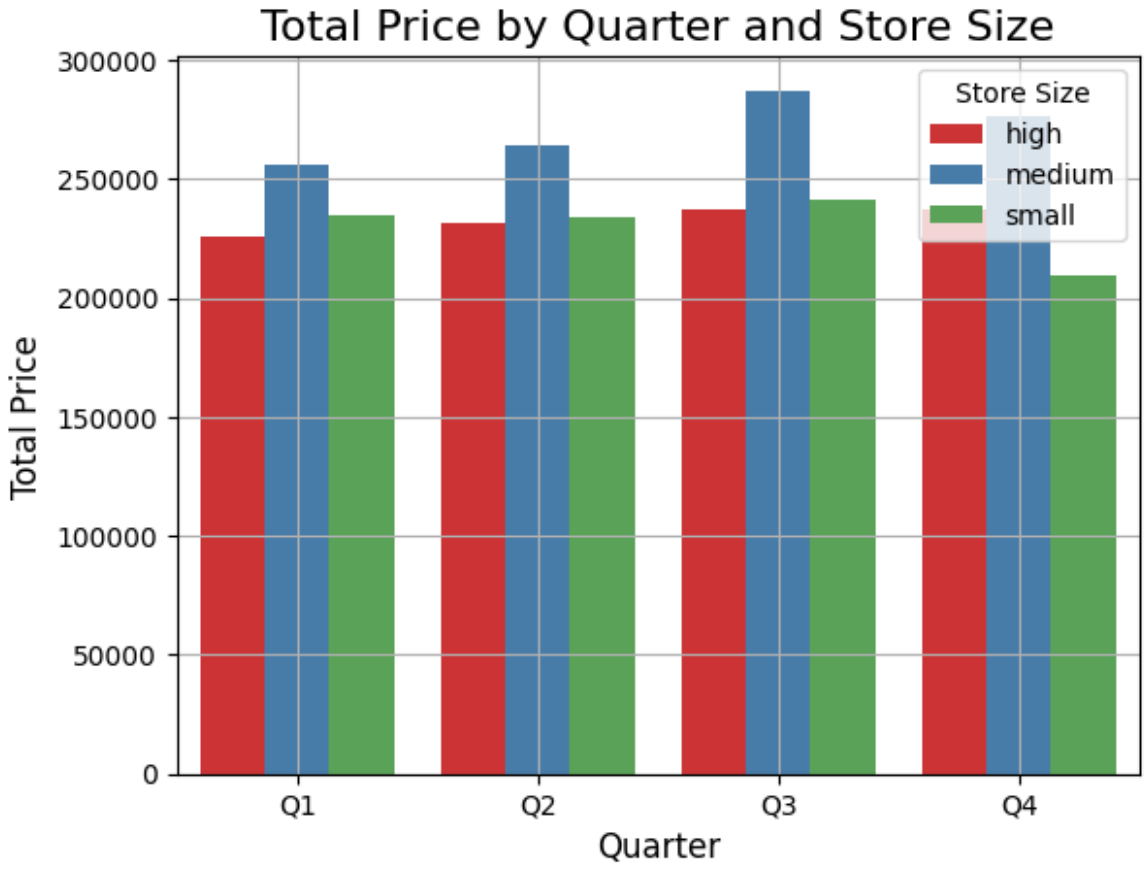
```
The most common unit is: cans. Cause it's total row count is 7664
rows.
```

Q-2: Find the store_size-wise quarterly total sales price of all stores joining the fact table and respective dimension tables and visualize it to a bar chart.

```
fact_store_table = pd.merge(fact_table, store_dim, on='store_key')
fact_store_time_table = pd.merge(fact_store_table, time_dim, on=
'time_key')

new_df1 = fact_store_time_table[['store_size',
'quarter','total_price']]
mask_1 = new_df1.groupby(['store_size', 'quarter'])
['total_price'].sum().reset_index()
print(f'this is avg total price data: \n {mask_1}')

sns.barplot(
    x='quarter',
      y='total_price',
        hue='store_size',
         palette='Set1',
          data=mask_1)

# Customize the plot
plt.title('Total Price by Quarter and Store Size', fontsize=16)
plt.xlabel('Quarter', fontsize=12)
plt.ylabel('Total Price', fontsize=12)
plt.legend(title='Store Size')
plt.grid()
plt.show()
```

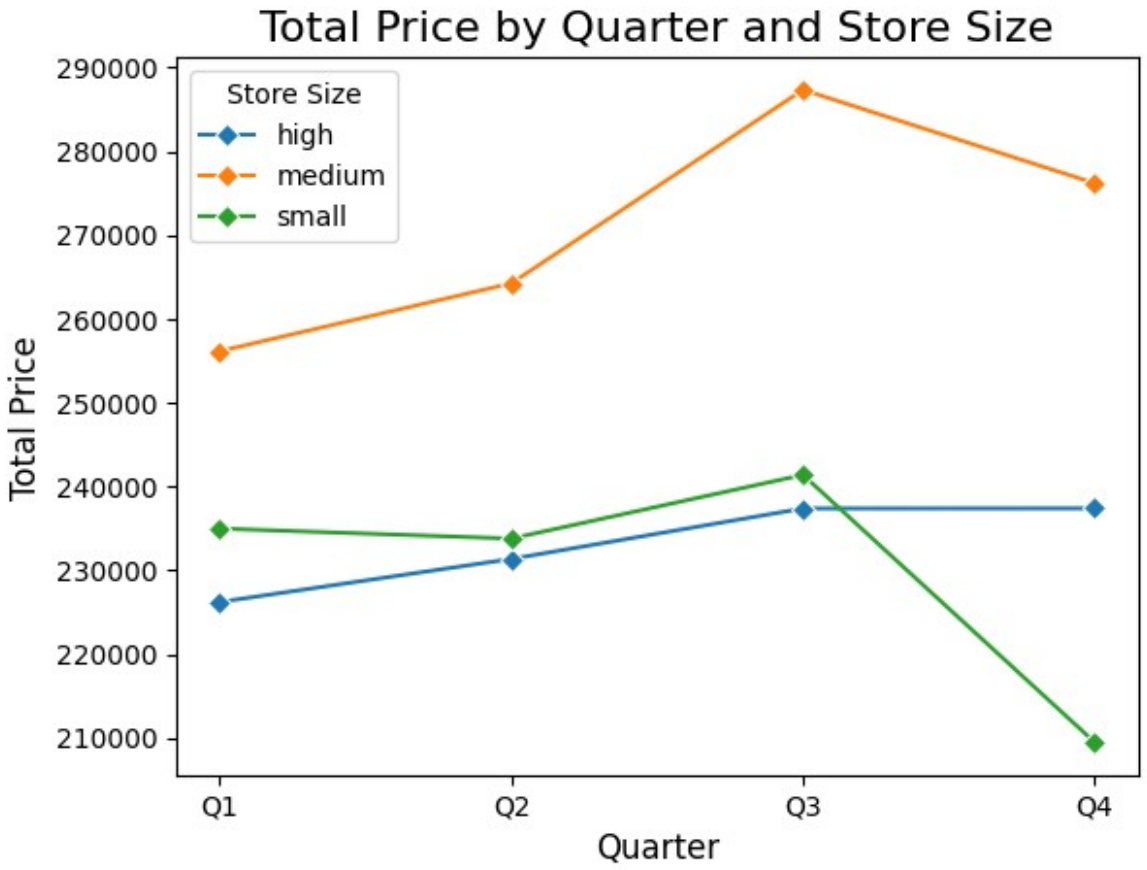
```
this is avg total price data:
    store_size quarter  total_price
0         high      Q1    226211.75
1         high      Q2    231349.50
```

```
2         high      Q3    237378.25
3         high      Q4    237388.75
4       medium      Q1    256150.00
5       medium      Q2    264251.25
6       medium      Q3    287346.50
7       medium      Q4    276202.00
8        small      Q1    234977.75
9        small      Q2    233783.50
10       small      Q3    241386.00
11       small      Q4    209478.00
```




```
# alternate plot
sns.lineplot(
    data=mask_1,
    x='quarter',
    y='total_price',
    hue='store_size',
    marker='D'  # Add markers to highlight data points
)
plt.title('Total Price by Quarter and Store Size', fontsize=16)
plt.xlabel('Quarter', fontsize=12)
plt.ylabel('Total Price', fontsize=12)
```

```
plt.legend(title='Store Size')
plt.show()
```

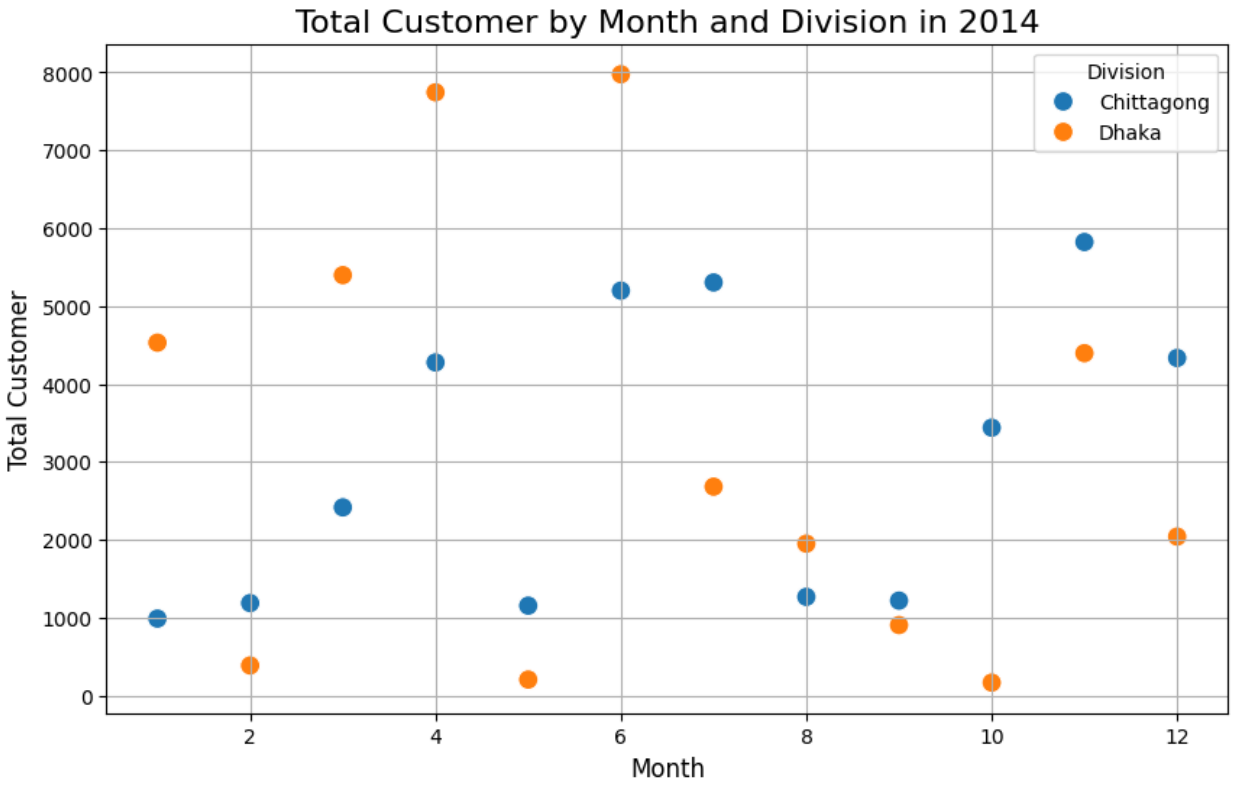


Q-3: Compare the year-wise monthly total quantity in Dhaka and Chittagong divisions of all customers joining the fact table and respective dimension tables and visualize it to a scatter chart.

```
# year, month, city, total customer
fact_store_time_custo_tab = pd.merge(fact_store_time_table,
customer_dim, on='customer_key')

df2 = fact_store_time_custo_tab[['customer_key', 'month', 'year',
'division_y']].reset_index()
filtered_data = df2[df2['division_y'].isin(['Dhaka', 'Chittagong'])]
yearly_monthly_totals = filtered_data.groupby(['year', 'month',
'division_y'], as_index=False)['customer_key'].idxmax()
yearly_monthly_totals
data_2014 = yearly_monthly_totals[yearly_monthly_totals['year'] ==
2014]
```

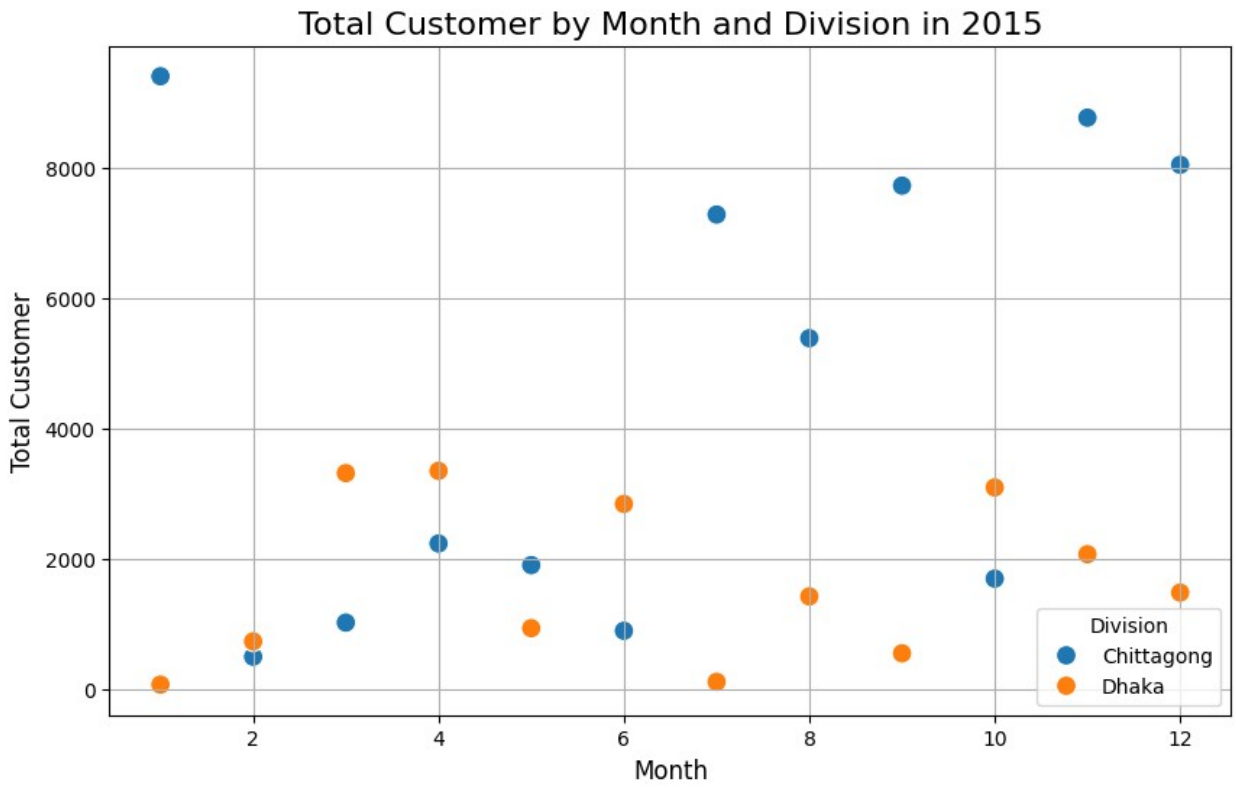
```
print(f'this is the year data: \n{data_2014}')

plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=data_2014,
    x='month',
    y='customer_key',
    hue='division_y',
    palette='tab10',
    s=100
)

plt.title('Total Customer by Month and Division in 2014', fontsize=16)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Total Customer', fontsize=12)
plt.legend(title='Division')
plt.grid()
plt.show()
```

```
this is the year data:
    year  month  division_y  customer_key
0   2014      1  Chittagong           987
1   2014      1       Dhaka          4530
2   2014      2  Chittagong          1186
3   2014      2       Dhaka           385
4   2014      3  Chittagong          2415
5   2014      3       Dhaka          5397
6   2014      4  Chittagong          4276
7   2014      4       Dhaka          7744
8   2014      5  Chittagong          1153
9   2014      5       Dhaka           204
10  2014      6  Chittagong          5198
11  2014      6       Dhaka          7976
12  2014      7  Chittagong          5303
13  2014      7       Dhaka          2681
14  2014      8  Chittagong          1266
15  2014      8       Dhaka          1949
16  2014      9  Chittagong          1218
17  2014      9       Dhaka           904
18  2014     10  Chittagong          3438
19  2014     10       Dhaka           165
20  2014     11  Chittagong          5822
21  2014     11       Dhaka          4395
22  2014     12  Chittagong          4333
23  2014     12       Dhaka          2040
```




```
def total_customer_by_month_division(year):
    df2 = fact_store_time_custo_tab[['customer_key', 'month', 'year',
'division_y']].reset_index()
    filtered_data = df2[df2['division_y'].isin(['Dhaka',
'Chittagong'])]
    yearly_monthly_totals = filtered_data.groupby(['year', 'month',
'division_y'], as_index=False)['customer_key'].idxmax()
    yearly_monthly_totals
    data_2014 = yearly_monthly_totals[yearly_monthly_totals['year'] ==
year]
    plt.figure(figsize=(10, 6))
    sns.scatterplot(
        data=data_2014,
        x='month',
        y='customer_key',
        hue='division_y',
        palette='tab10',
        s=100)

    plt.title(f'Total Customer by Month and Division in {year}',
fontsize=16)
    plt.xlabel('Month', fontsize=12)
    plt.ylabel('Total Customer', fontsize=12)
    plt.legend(title='Division')
    plt.grid()
```

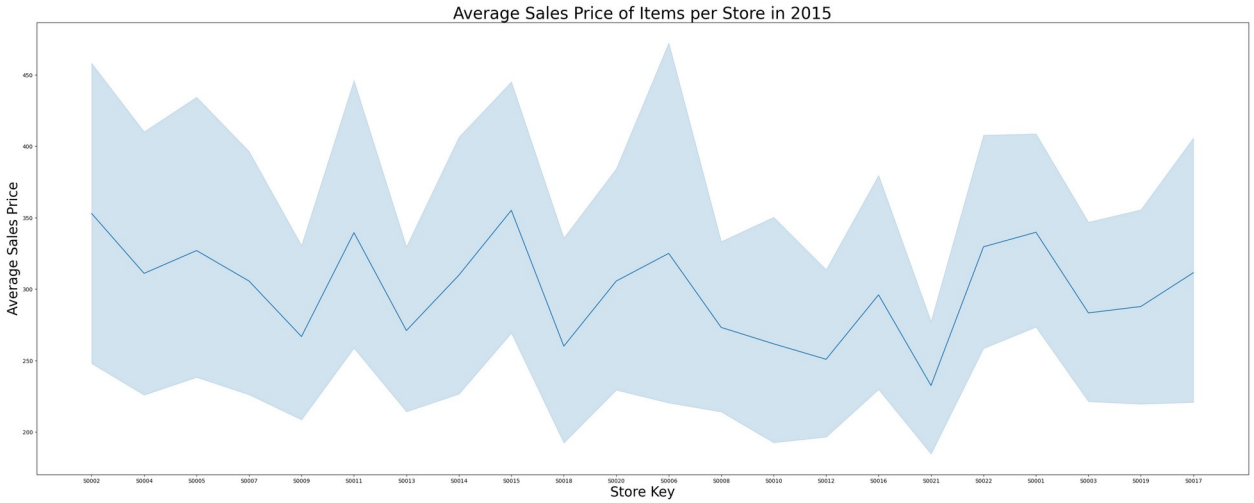
```
    return plt.show()

total_customer_by_month_division(2015)
```



Q-4: What are the average sales price of items per store yearly? Show the data in a line chart.

```
# total price, item, store, year
fact_store_time_custo_tab.columns
df3 = fact_store_time_custo_tab[['item_key', 'store_key',
'total_price', 'year']]
fil_data = df3.groupby(['year', 'item_key', 'store_key'])
['total_price'].mean().reset_index()
avg_14 = fil_data[fil_data['year'] == 2015]
data_2015 = avg_14[avg_14['year'] == 2015]
data_2015
# Create a line chart
plt.figure(figsize=(40, 15))
sns.lineplot(x='store_key', y='total_price', data=data_2015)

# Set the title and labels
plt.title('Average Sales Price of Items per Store in 2015', fontsize =
30)
plt.xlabel('Store Key', fontsize = 25)
```

```
plt.ylabel('Average Sales Price', fontsize = 25)
plt.show()
```




```
def avg_sales_per_str_by_year(year):
    fact_store_time_custo_tab.columns
    df3 = fact_store_time_custo_tab[['item_key', 'store_key',
'total_price', 'year']]
    fil_data = df3.groupby(['year', 'item_key', 'store_key'])
['total_price'].mean().reset_index()
    avg_14 = fil_data[fil_data['year'] == year]
    year_data = avg_14[avg_14['year'] == year]
    print(f'this is the data: \n{year_data}')

    # Create a line chart

    plt.figure(figsize=(40, 15))
    sns.lineplot(x='store_key',
                 y='total_price',
                 data=year_data,)
    plt.title(f'Average Sales Price of Items per Store in {year}',
fontsize = 30)
    plt.xlabel('Store Key', fontsize = 25)
    plt.ylabel('Average Sales Price', fontsize = 25)
    return plt.show()
avg_sales_per_str_by_year(2020)
```

```
this is the data:
      year item_key store_key  total_price
5075  2020   I00001     S0001       161.00
5076  2020   I00001     S0002       310.50
5077  2020   I00001     S0004       258.75
5078  2020   I00001     S0006       172.50
5079  2020   I00001     S0009       218.50
...    ...      ...       ...          ...
```

```
5900  2020   I00069     S0016       455.00
5901  2020   I00069     S0017       192.50
5902  2020   I00069     S0019       367.50
5903  2020   I00069     S0020       560.00
5904  2020   I00069     S0022       358.75

[830 rows x 4 columns]
```