In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
#Loading the file to be cleaned based on its location in the computer.
file_path = "C:\\Users\\Owner\\Downloads\\retail_store_sales.csv"
df = pd.read_csv(file_path)
```

In [3]:

```
#see basic information of the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12575 entries, 0 to 12574
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Transaction ID    12575 non-null  object 
 1   Customer ID       12575 non-null  object 
 2   Category          12575 non-null  object 
 3   Item              11362 non-null  object 
 4   Price Per Unit    11966 non-null  float64
 5   Quantity          11971 non-null  float64
 6   Total Spent       11971 non-null  float64
 7   Payment Method    12575 non-null  object 
 8   Location          12575 non-null  object 
 9   Transaction Date  12575 non-null  object 
 10  Discount Applied  8376 non-null   object 
dtypes: float64(3), object(8)
memory usage: 1.1+ MB
```

In [4]:

```
df.head(10)
```

Out[4]:

	Transaction ID	Customer ID	Category	Item	Price Per Unit	Quantity	Total Spent	Payment Method	Location	Transaction Date	Discount Applied
0	TXN_6867343	CUST_09	Patisserie	Item_10_PAT	18.5	10.0	185.0	Digital Wallet	Online	2024-04-08	True
1	TXN_3731986	CUST_22	Milk Products	Item_17_MILK	29.0	9.0	261.0	Digital Wallet	Online	2023-07-23	True
2	TXN_9303719	CUST_02	Butchers	Item_12_BUT	21.5	2.0	43.0	Credit Card	Online	2022-10-05	False
3	TXN_9458126	CUST_06	Beverages	Item_16_BEV	27.5	9.0	247.5	Credit Card	Online	2022-05-07	NaN
4	TXN_4575373	CUST_05	Food	Item_6_FOOD	12.5	7.0	87.5	Digital Wallet	Online	2022-10-02	False
5	TXN_7482416	CUST_09	Patisserie	NaN	NaN	10.0	200.0	Credit Card	Online	2023-11-30	NaN
6	TXN_3652209	CUST_07	Food	Item_1_FOOD	5.0	8.0	40.0	Credit Card	In-store	2023-06-10	True
7	TXN_1372952	CUST_21	Furniture	NaN	33.5	NaN	NaN	Digital Wallet	In-store	2024-04-02	True
8	TXN_9728486	CUST_23	Furniture	Item_16_FUR	27.5	1.0	27.5	Credit Card	In-store	2023-04-26	False
9	TXN_2722661	CUST_25	Butchers	Item_22_BUT	36.5	3.0	109.5	Cash	Online	2024-03-14	False

In [5]:

```
df.describe(include='all')
```

Out[5]:

	Transaction ID	Customer ID	Category	Item	Price Per Unit	Quantity	Total Spent	Payment Method	Location	Transaction Date	Discount Applied
count	12575	12575	12575	11362	11966.000000	11971.000000	11971.000000	12575	12575	12575	8376
unique	12575	25	8	200	NaN	NaN	NaN	3	2	1114	2
top	TXN_6867343	CUST_05	Furniture	Item_2_BEV	NaN	NaN	NaN	Cash	Online	2022-05-30	True
freq	1	544	1591	126	NaN	NaN	NaN	4310	6354	26	4219
mean	NaN	NaN	NaN	NaN	23.365912	5.536380	129.652577	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	10.743519	2.857883	94.750697	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	5.000000	1.000000	5.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	14.000000	3.000000	51.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	23.000000	6.000000	108.500000	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	33.500000	8.000000	192.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	41.000000	10.000000	410.000000	NaN	NaN	NaN	NaN

In [6]:

```
# count missing vallues per column and determine the percentage of each column.
# Then Create a dataframe to show the statistics
mis_value = df.isnull().sum()
mis_value_percentage = (mis_value / len(df)) * 100
mis_df = pd.DataFrame({ 'Missing Value': mis_value, 'Percentage (%)': mis_value_percentage})
mis_df
```

Out[6]:

	Missing Value	Percentage (%)
Transaction ID	0	0.000000
Customer ID	0	0.000000
Category	0	0.000000
Item	1213	9.646123
Price Per Unit	609	4.842942
Quantity	604	4.803181
Total Spent	604	4.803181
Payment Method	0	0.000000
Location	0	0.000000
Transaction Date	0	0.000000
Discount Applied	4199	33.391650

In [7]:
```
#Show only columns with missing values
print(mis_df[mis_df["Missing Value"] > 0])
```

```
                  Missing Value  Percentage (%)
Item                       1213        9.646123
Price Per Unit              609        4.842942
Quantity                    604        4.803181
Total Spent                 604        4.803181
Discount Applied           4199       33.391650
```

In [8]:
```
#Standardizing column names
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
```

In [9]:
```
#Convert the Trasaction Date to date time format
df['transaction_date']= pd.to_datetime(df["transaction_date"], errors = "coerce")
```

In [10]:
```
#Extract specific values from datetime variable to create new columns
df['day_of_week'] = df['transaction_date'].dt.strftime('%A')
df['month_name'] = df['transaction_date'].dt.strftime('%B')
```

In [11]:
```
#Calculating price per unit
temp_price = df['quantity'].notna() & df['total_spent'].notna()
df.loc[temp_price, 'price_per_unit'] = df.loc[temp_price, 'total_spent']/df.loc[temp_price, 'quantity']
```

In [12]:
```
#fill item column by grouping the category and price per unit to get the unique item
df['item'] = df.groupby(['category','price_per_unit'])['item'].transform(lambda x: x.ffill().bfill())
```

In [13]:
```
#check the quantity column to see if we can the missing data
df[df['quantity'].isnull()]
```

Out[13]:

	transaction_id	customer_id	category	item	price_per_unit	quantity	total_spent	payment_method	location	transaction_date	discount_applied	day_of_week
7	TXN_1372952	CUST_21	Furniture	Item_20_FUR	33.5	NaN	NaN	Digital Wallet	In-store	2024-04-02	True	Tuesday
15	TXN_1809665	CUST_14	Beverages	Item_14_BEV	24.5	NaN	NaN	Credit Card	In-store	2022-05-11	NaN	Wednesday
19	TXN_4206593	CUST_01	Furniture	Item_21_FUR	35.0	NaN	NaN	Digital Wallet	Online	2025-01-13	False	Monday
25	TXN_3481599	CUST_05	Furniture	Item_24_FUR	39.5	NaN	NaN	Cash	Online	2022-09-08	False	Thursday
34	TXN_1621497	CUST_06	Patisserie	Item_13_PAT	23.0	NaN	NaN	Cash	In-store	2023-02-18	NaN	Saturday
...	...	...	...	...	...	...	...	...	...	...	...	...
12527	TXN_1069238	CUST_23	Food	Item_1_FOOD	5.0	NaN	NaN	Digital Wallet	In-store	2022-08-13	False	Saturday
12552	TXN_4823896	CUST_05	Milk Products	Item_3_MILK	8.0	NaN	NaN	Cash	In-store	2022-07-21	False	Thursday
12556	TXN_4397672	CUST_04	Beverages	Item_25_BEV	41.0	NaN	NaN	Credit Card	Online	2024-11-28	True	Thursday
12562	TXN_7422454	CUST_07	Butchers	Item_20_BUT	33.5	NaN	NaN	Cash	Online	2023-04-15	NaN	Saturday
12564	TXN_2153066	CUST_17	Electric household essentials	Item_17_EHE	29.0	NaN	NaN	Digital Wallet	In-store	2024-03-28	False	Thursday

604 rows × 13 columns

In [14]:
```
#Drop missing quantity because where quantity is missing, total spent is also missing. And the perentage of such missing data is 4%, so it w
df = df.dropna(subset=['quantity', 'total_spent'])
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11971 entries, 0 to 12574
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   transaction_id    11971 non-null  object
 1   customer_id       11971 non-null  object
 2   category          11971 non-null  object
 3   item              11971 non-null  object
 4   price_per_unit    11971 non-null  float64
 5   quantity          11971 non-null  float64
 6   total_spent       11971 non-null  float64
 7   payment_method    11971 non-null  object
 8   location          11971 non-null  object
 9   transaction_date  11971 non-null  datetime64[ns]
 10  discount_applied  7983 non-null   object
 11  day_of_week       11971 non-null  object
 12  month_name        11971 non-null  object
dtypes: datetime64[ns](1), float64(3), object(9)
memory usage: 1.3+ MB
```

In [16]:
```
df['discount_applied'].unique()
```

Out[16]:
```
array([True, False, nan], dtype=object)
```

In [18]:
```
#Drop the discount column
df = df.drop('discount_applied', axis=1)
```

In [19]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11971 entries, 0 to 12574
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   transaction_id    11971 non-null  object
 1   customer_id       11971 non-null  object
 2   category          11971 non-null  object
 3   item              11971 non-null  object
 4   price_per_unit    11971 non-null  float64
 5   quantity          11971 non-null  float64
 6   total_spent       11971 non-null  float64
 7   payment_method    11971 non-null  object
 8   location          11971 non-null  object
 9   transaction_date  11971 non-null  datetime64[ns]
 10  day_of_week       11971 non-null  object
 11  month_name        11971 non-null  object
dtypes: datetime64[ns](1), float64(3), object(8)
memory usage: 1.2+ MB
```

In [21]:

```
#Check to see sample of the cleaned data
df.head(10)
```

Out[21]:

	transaction_id	customer_id	category	item	price_per_unit	quantity	total_spent	payment_method	location	transaction_date	day_of_week	month_name
0	TXN_6867343	CUST_09	Patisserie	Item_10_PAT	18.5	10.0	185.0	Digital Wallet	Online	2024-04-08	Monday	April
1	TXN_3731986	CUST_22	Milk Products	Item_17_MILK	29.0	9.0	261.0	Digital Wallet	Online	2023-07-23	Sunday	July
2	TXN_9303719	CUST_02	Butchers	Item_12_BUT	21.5	2.0	43.0	Credit Card	Online	2022-10-05	Wednesday	October
3	TXN_9458126	CUST_06	Beverages	Item_16_BEV	27.5	9.0	247.5	Credit Card	Online	2022-05-07	Saturday	May
4	TXN_4575373	CUST_05	Food	Item_6_FOOD	12.5	7.0	87.5	Digital Wallet	Online	2022-10-02	Sunday	October
5	TXN_7482416	CUST_09	Patisserie	Item_11_PAT	20.0	10.0	200.0	Credit Card	Online	2023-11-30	Thursday	November
6	TXN_3652209	CUST_07	Food	Item_1_FOOD	5.0	8.0	40.0	Credit Card	In-store	2023-06-10	Saturday	June
8	TXN_9728486	CUST_23	Furniture	Item_16_FUR	27.5	1.0	27.5	Credit Card	In-store	2023-04-26	Wednesday	April
9	TXN_2722661	CUST_25	Butchers	Item_22_BUT	36.5	3.0	109.5	Cash	Online	2024-03-14	Thursday	March
10	TXN_8776416	CUST_22	Butchers	Item_3_BUT	8.0	9.0	72.0	Cash	In-store	2024-12-14	Saturday	December

In [26]:

```
df.to_csv('cleaned_retail_store_sales.csv', index=False)
```

In [ ]:

In [127]:

In [ ]: