Basic Data Cleaning on a Retail Dataset

This notebook demonstrates basic cleaning techniques I will implement, such as...

- Handling missing values
- Removing duplicates
- Converting data types when necessary

For this project, I will be using a retail store sales dataset on Kaggle

In [15]:
```
import pandas as pd
```

In [16]:
```
df = pd.read_csv("retail_store_sales.csv")
df.head()
```

Out[16]:

	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

First, I will check the info and stats of this dataset...

In [17]:
```
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
```

From this info() command, I can infer that there are 12575 rows, but some columns don't have that many rows. For example, the columns Item, Price Per unit, Quantity, Total Spent, and Discount Applied have missing rows and therefore require cleaning. Another thing I noticed is that the Transaction Date is an object Dtype, but it should be a datetime for any time analysis. Additionally, Discount Applied is an object, but it should be a boolean, true or false.

First, I will look for any duplicate values and get rid of them

In [18]:
```
#Here I am checking if there are any duplicates. It turns out that there aren't any duplicates, so I don't have to drop anything here.
df.duplicated().sum()
```

Out[18]:
```
np.int64(0)
```

Next, I will fill in some missing values and get rid of others

In [ ]:
```
# I am going to assume that if there is no value for discount, then no discount was applied, and it's false.
# I am also going to change the data type from object to boolean
df["Discount Applied"].fillna(False, inplace=True)
df["Discount Applied"] = df["Discount Applied"].astype('bool')
```

In [19]:
```
#Now I am going drop all rows that are missing values for Price Per unit, Quantity, and Total Spent, because I don't know that data.
df.dropna(subset=["Price Per Unit", "Quantity", "Total Spent"], inplace = True)
```

In [20]:
```
#Finally, I am going to change Transaction Data from an object Dtype to a datetime Dtype
df["Transaction Date"] = pd.to_datetime(df["Transaction Date"])
```

In [21]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11362 entries, 0 to 12574
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype         
---  ------            --------------  -----         
 0   Transaction ID    11362 non-null  object        
 1   Customer ID       11362 non-null  object        
 2   Category          11362 non-null  object        
 3   Item              11362 non-null  object        
 4   Price Per Unit    11362 non-null  float64       
 5   Quantity          11362 non-null  float64       
 6   Total Spent       11362 non-null  float64       
 7   Payment Method    11362 non-null  object        
 8   Location          11362 non-null  object        
 9   Transaction Date  11362 non-null  datetime64[ns]
 10  Discount Applied  7579 non-null   object        
dtypes: datetime64[ns](1), float64(3), object(7)
memory usage: 1.0+ MB
```

Now with all the cleaning done, we can see that there are no longer any missing values, and the Transaction Date/Discount Applied columns have been given appropriate data types. Around 1000 rows of missing information were deleted, and many more were updated based on the assumption that if a discount wasn't recorded, then there was no discount.

In [ ]: