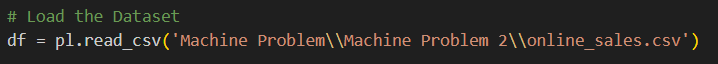
**Submission**

Document and report the steps and show the results following the tasks in order with reference to the expected output. In reporting the cleaned dataset, paste a screenshot of the table taken from the sheet's or MS Excels's interface. Upload the report in a PDF using the link provided.

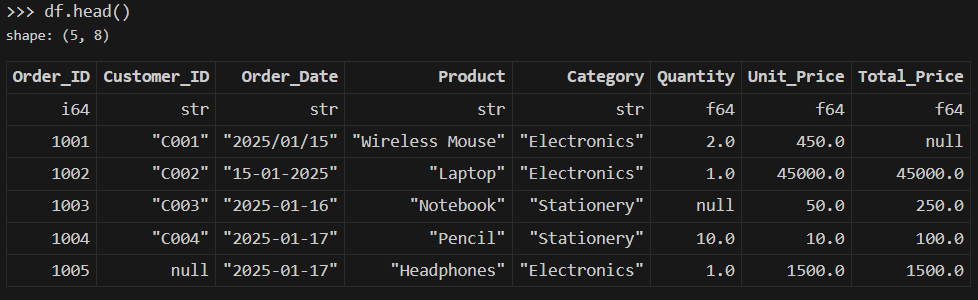
*Note: You can find the code used for this Machine Problem in this link (*[*MachineProblem2.py*](https://github.com/WakenMac/Waks-CSDS312-Stuff/blob/main/Machine%20Problem/Machine%20Problem%202/MachineProblem2.py)*)*

**Part 1: Collecting and Exploring Data**

1. Load the dataset using polars.



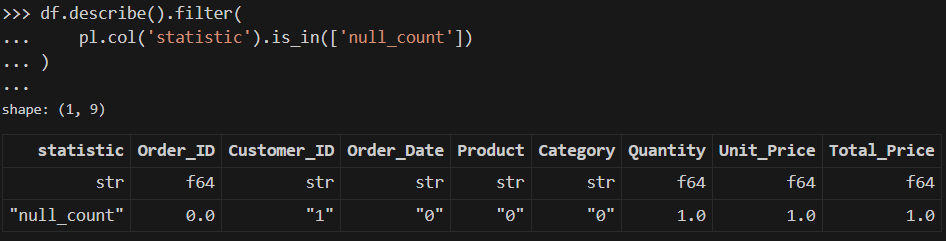
1. Display the first few rows and basic info about the dataset.



1. Compute summary statistics for numeric columns (Quantity, Unit\_Price, Total\_Price).



1. Find the number of missing values per column.



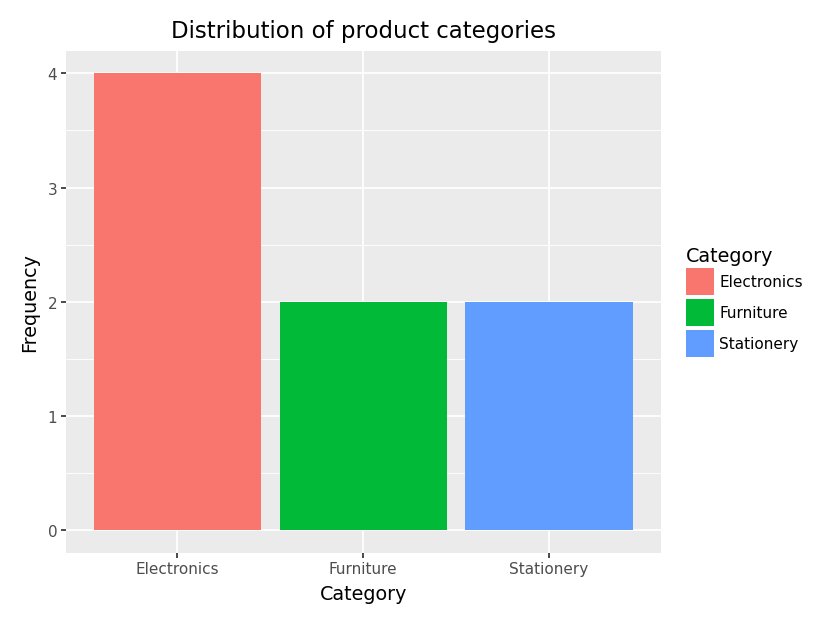
We can see that there are null values in the Customer\_ID, Quantity, Unit\_Price, and Total\_Price table, thus, we will need to drop null values or use imputation to clean our dataset.

1. Create visualizations using plotnine:

* Distribution of product categories



*Getting the aggregate number of products per category, then plotting them in a bar plot*

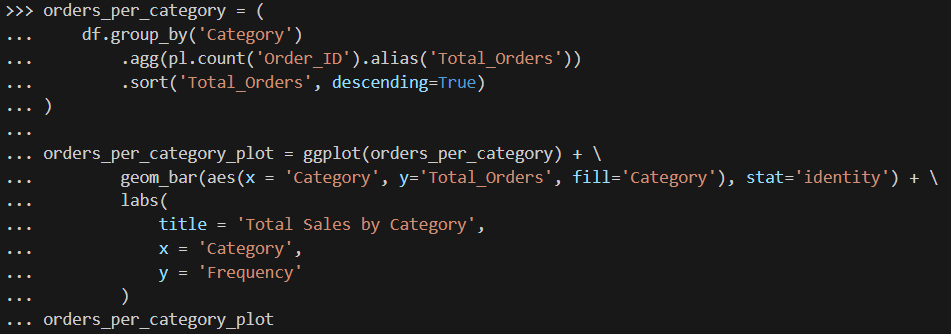


*Distribution of product categories*

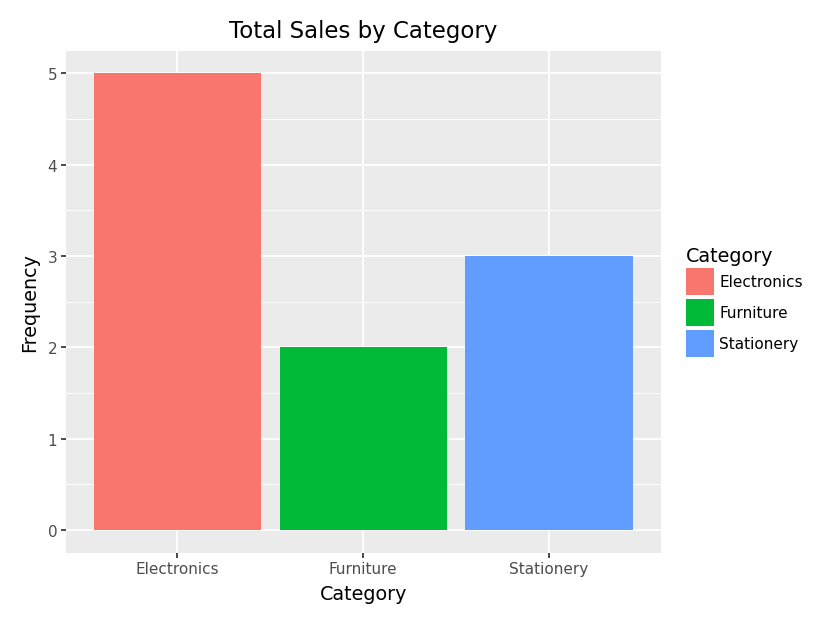
We can see from the bar chart that out of all of the orders in our dataset there are twice as much Electronic products than Furniture and Stationary products. Though this is possibly biased as this is generated from the list of orders rather than a literal product list or inventory.

Aside from that, this can be an insight to how electronics may be a more salable product category than Furniture and Stationary products.

* Total sales by category

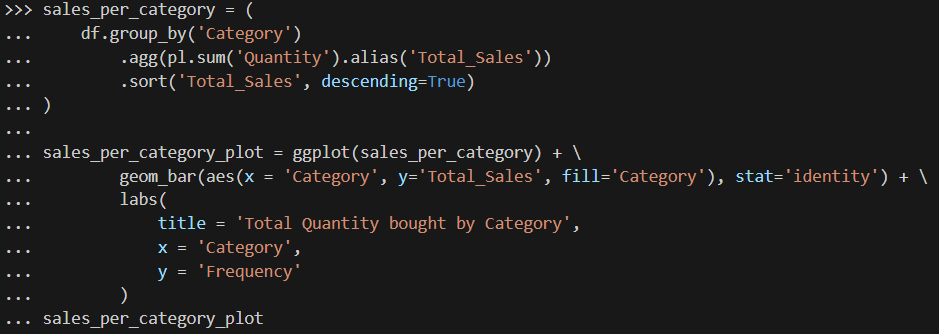


*Getting the number of orders per category then plotting it in a bar plot*

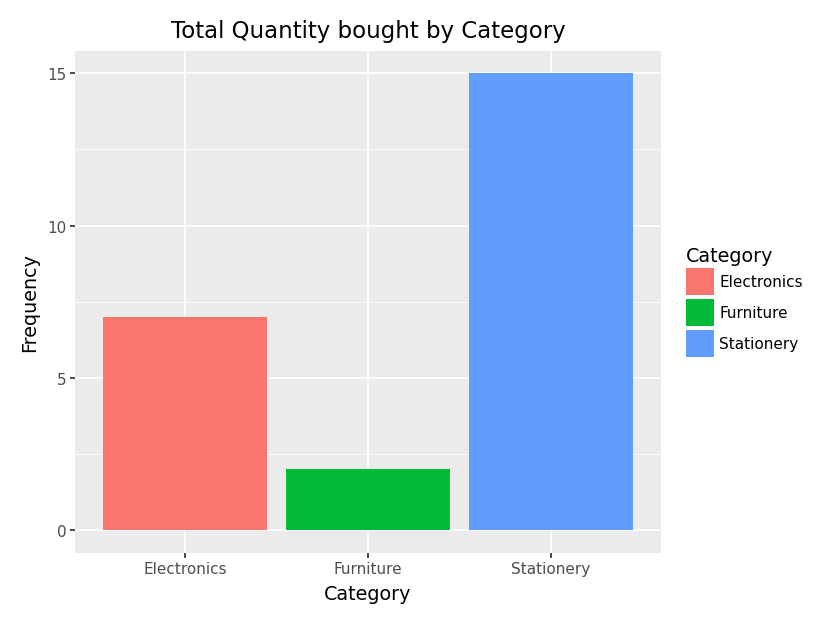


*Total number of sales by category (Electronics has the most amount)*

If we compare this to our previous plot, we can derive that Electronics and Stationary products are salable, each having at least one product bought twice (i.e., this is because there are 5 and 3 items bought for Electronics and Stationary categories even though there are 4 and 2 products under them), whereas both sales under the Furniture category are different products. However, this cannot be the true number of sales as this is counted based on the number of orders, and doesn’t account for the number of items bought per order.



*Code to aggregate the quantity of items bought per category, and plot them.*

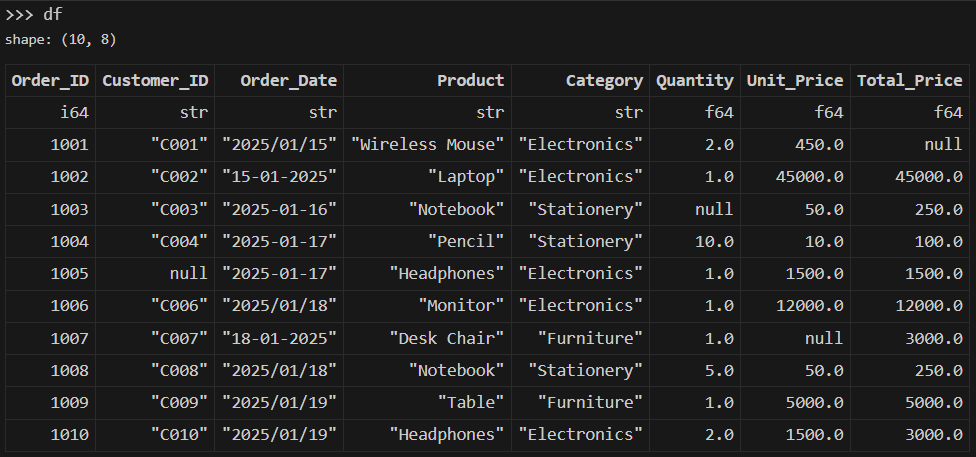


*Bar plot of total quantity of items bought*

Now, we finally get a better interpretation to which products are more salable, though it is understandable that furniture will have the least amount of sales as compared to stationary products (e.g., Notebooks and Pencils) and electronic products (e.g., Laptop, Monitor, Wireless Mouse, and Headphones).

Through this, we can focus our advertising more on Furniture and Electronic products to increase sales, especially due to the high price of Furniture and Electronic products (i.e., Desk Chair costing 5,000 and Laptop costing 45,000). Similarly, we now know that Stationary products should be checked regularly as they may be out of stock more often than other product categories.

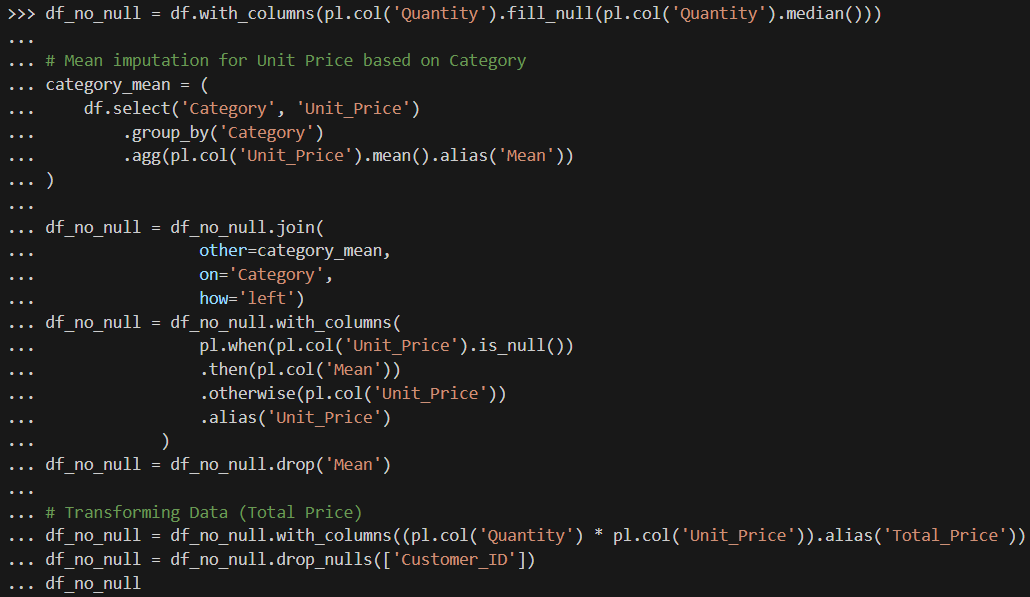
**Part 2: Cleaning and Preparing Data**

****

*DataFrame before removing null values*

1. Handle missing values:

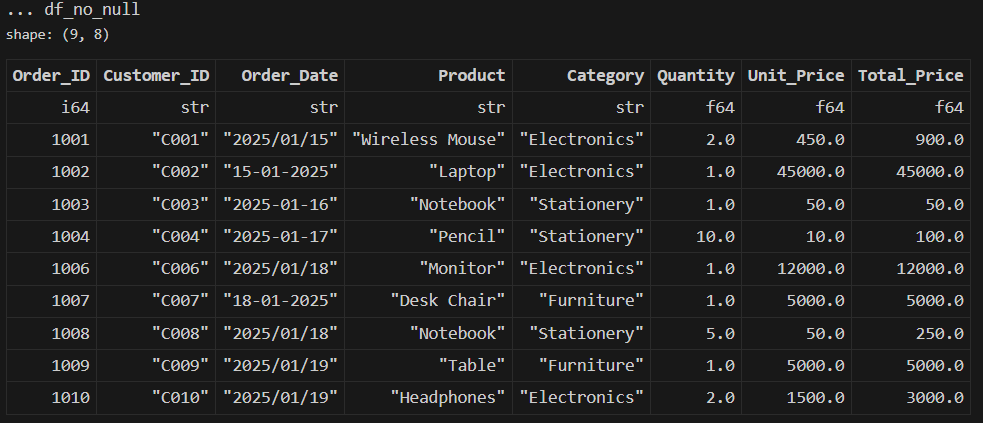
* Fill missing Quantity with the median quantity.
* Fill missing Unit\_Price with mean price per category.
* Compute missing Total\_Price as Quantity x Unit\_Price.
* Drop rows with missing Customer\_ID.



*Code to handle missing values and missing columns*

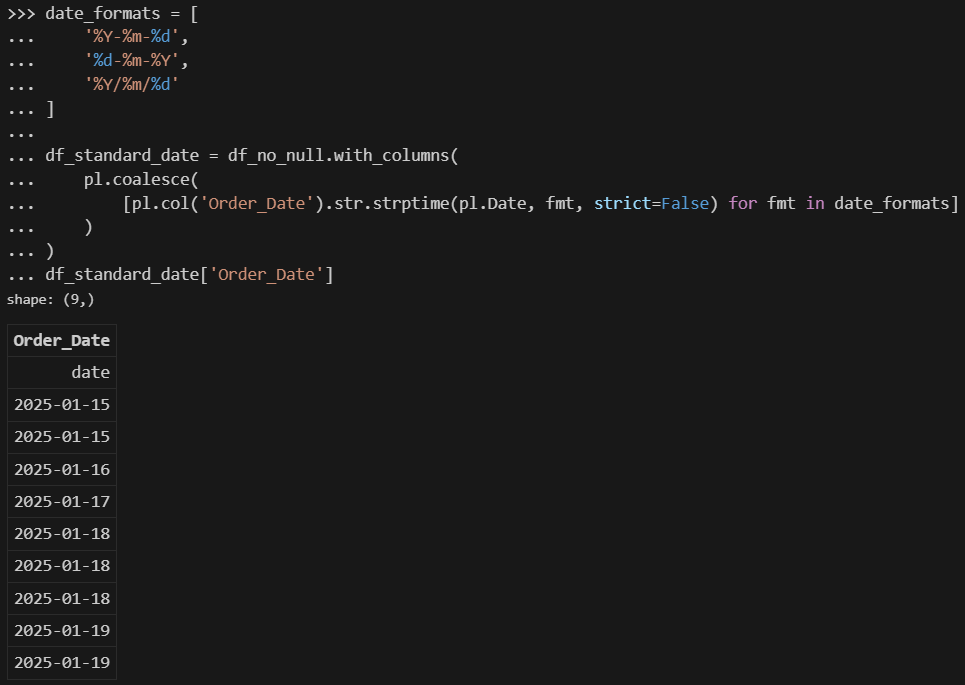
The block of code is made to remove null values through the following steps:

* + 1. Use median imputation for our Quantity column
    2. Aggregate the DataFrame grouped by Category and get the mean Unit\_Price, have the column be named “Mean”
    3. Join the aggregated DataFrame to our original DataFrame, using a one-to-many left join on the “Categories” column
    4. Use an if-else statement to use mean impute on null values in Unit\_Price based on the its category.
    5. Drop the “Mean” column
    6. Replace the null value for Total\_Price using feature engineering
    7. Drop rows with null values on the “Customer\_ID” column



*Resulting DataFrame after running the code*

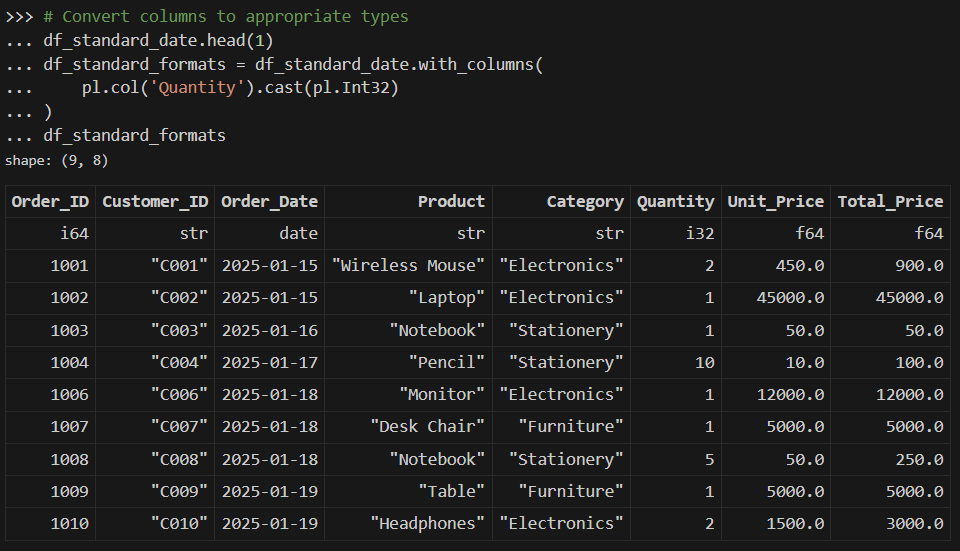
1. Standardize the Order\_date format to YYYY-MM-DD.



*Using coalesce() to standardize dates with multiple formats*

I used a list comprehension to apply different formats to our Order\_Date column, and wrap it all with the pl.coalesce() method to combine the different non-null results. This results in a code that can standardize the Order\_Date column regardless of the different date formats in our original csv.

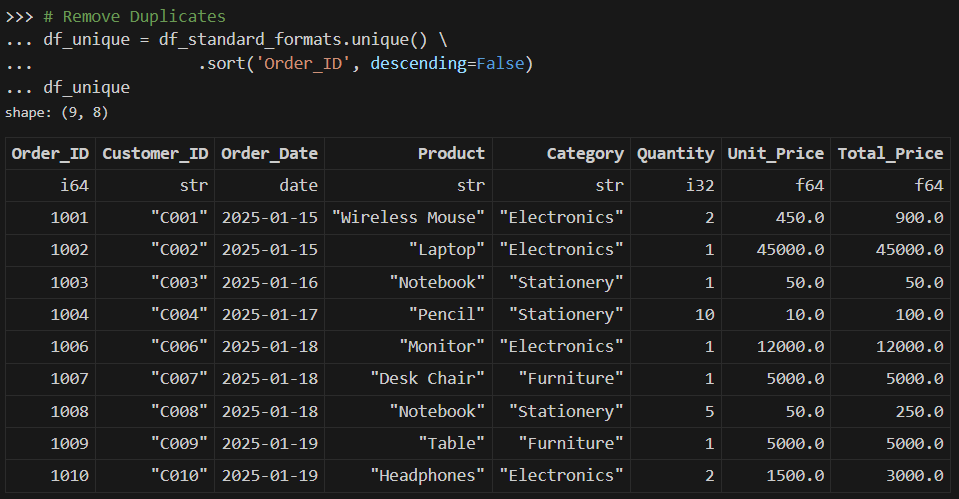
1. Convert columns to proper data types.



*Converting quantity to an int.*

We will then convert the Quantity column to an integer, as there are no floats for quantity.

1. Remove duplicates if any.



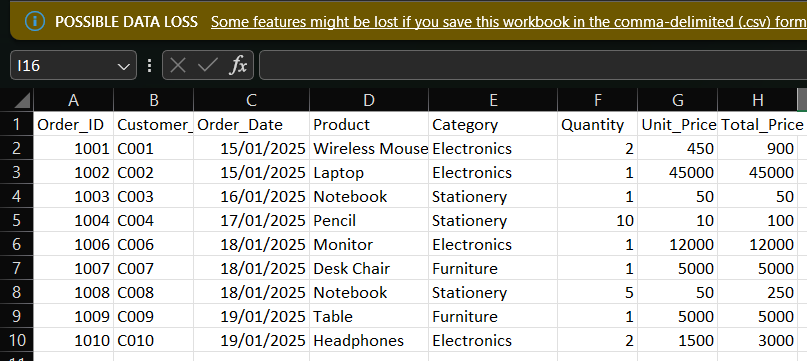
*DataFrame after handling null values, standardizing dates, formatting data types, and removing duplicates.*

We then remove duplicate entries in our DataFrame.

1. Save the cleaned dataset as cleaned\_online\_sales.csv.



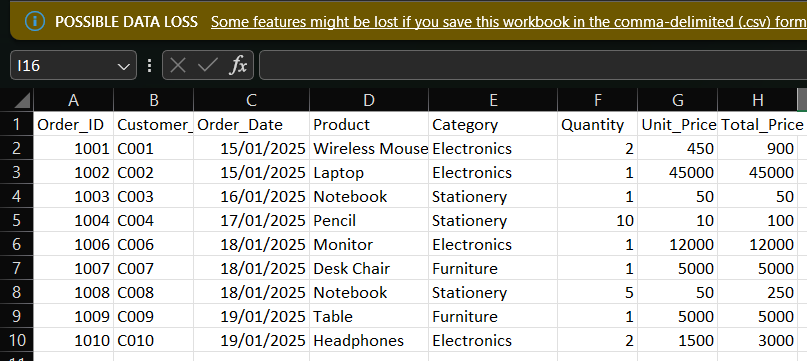
*Saving the Polars DataFrame into a csv.*



*Viewing the saved csv in Microsoft Excel*

**Expected Output**

* Bar chart of total sales per category.
* Distribution plot of product categories.
* Explanation for each cleaning step.
* Interpretation of the charts.
* Cleaned dataset file cleaned\_online\_sales.csv.



*Note: You can find the dataset file in the github link* [*(cleaned\_online\_sales.csv)*](https://github.com/WakenMac/Waks-CSDS312-Stuff/blob/main/Machine%20Problem/Machine%20Problem%202/cleaned_online_sales.csv)