app

# Exploratory Data Analysis[¶](#Exploratory-Data-Analysis)

1. Source the data.
2. Data Cleaning: Choose a Data Set of your choice and perform Data Cleaning process such as

* Remove the NULL Values(Regression)
* Handle the missing values(Either drop the missing values/replace with mean/replace with median/replace with mode)
* Handle the Outliers( Use Box Plot, Drop or replace the outliers using IQR method.)

Justify the methods chosen at each step of cleaning

1. Visualize the data using univariate and bi-variate analysis(Use Histogram)

Interpret the values of the graph and justify your analysis for the same.

In [1]:

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

In [2]:

df = pd.read\_excel("../dataset/Coffee\_Shop\_Sales.xlsx")

In [3]:

df.shape

Out[3]:

(149116, 11)

In [4]:

df.head()

Out[4]:

|  | transaction\_id | transaction\_date | transaction\_time | transaction\_qty | store\_id | store\_location | product\_id | unit\_price | product\_category | product\_type | product\_detail |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2023-01-01 | 07:06:11 | 2 | 5 | Lower Manhattan | 32 | 3.0 | Coffee | Gourmet brewed coffee | Ethiopia Rg |
| 1 | 2 | 2023-01-01 | 07:08:56 | 2 | 5 | Lower Manhattan | 57 | 3.1 | Tea | Brewed Chai tea | Spicy Eye Opener Chai Lg |
| 2 | 3 | 2023-01-01 | 07:14:04 | 2 | 5 | Lower Manhattan | 59 | 4.5 | Drinking Chocolate | Hot chocolate | Dark chocolate Lg |
| 3 | 4 | 2023-01-01 | 07:20:24 | 1 | 5 | Lower Manhattan | 22 | 2.0 | Coffee | Drip coffee | Our Old Time Diner Blend Sm |
| 4 | 5 | 2023-01-01 | 07:22:41 | 2 | 5 | Lower Manhattan | 57 | 3.1 | Tea | Brewed Chai tea | Spicy Eye Opener Chai Lg |

In [5]:

df.isnull().sum()

Out[5]:

transaction\_id 0  
transaction\_date 0  
transaction\_time 0  
transaction\_qty 0  
store\_id 0  
store\_location 0  
product\_id 0  
unit\_price 117  
product\_category 0  
product\_type 0  
product\_detail 17  
dtype: int64

* We will remove the rows that don't contain product detail as we can't fill with random info.
* Fill the unit\_price with the median value so we don't have any bias in price of the product.

In [6]:

df = df.dropna(subset=["product\_detail"])  
df["unit\_price"] = df["unit\_price"].fillna(df["unit\_price"].median())  
  
df.isnull().sum()

Out[6]:

transaction\_id 0  
transaction\_date 0  
transaction\_time 0  
transaction\_qty 0  
store\_id 0  
store\_location 0  
product\_id 0  
unit\_price 0  
product\_category 0  
product\_type 0  
product\_detail 0  
dtype: int64

In [7]:

plt.figure(figsize=(9, 6))  
sns.boxplot(data=df[["transaction\_qty", "unit\_price"]])  
plt.show()  
  
# function to handle outliers using IQR  
def handle\_outliers(df, column):  
 Q1 = df[column].quantile(0.25)  
 Q3 = df[column].quantile(0.75)  
 IQR = Q3 - Q1  
 lower\_bound = Q1 - 1.5 \* IQR  
 upper\_bound = Q3 + 1.5 \* IQR  
 df.loc[(df[column] < lower\_bound) | (df[column] > upper\_bound), column] = df[column].median()  
  
# applying the function to the 'transaction\_qty' and 'unit\_price' columns  
handle\_outliers(df, 'transaction\_qty')  
handle\_outliers(df, 'unit\_price')  
  
# box plot after handling outliers  
plt.figure(figsize=(10, 6))  
sns.boxplot(data=df[['transaction\_qty', 'unit\_price']])  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

![No description has been provided for this image](data:image/png;base64;base64,)

### Univariate Analysis[¶](#Univariate-Analysis)

In [8]:

df['unit\_price'].hist(bins=20, figsize=(10, 6))  
plt.xlabel('Price')  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### Bi-variate Analysis[¶](#Bi-variate-Analysis)

In [13]:

plt.figure(figsize=(10, 6))  
sns.scatterplot(x='unit\_price', y='transaction\_qty', data=df)  
plt.xlabel('Price')  
plt.ylabel('Quantity')  
plt.title('Price vs Quantity')  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)