**Descriptive analysis and data pre-processing on sales and discounts dataset**

**Introduction:**

To perform descriptive analysis, visualize data distributions and to pre-process the dataset for further analysis.

**1.Descriptive Analytics for Numerical Columns**

**Objective: -**

Compute and analyse basic statistical measures for numerical columns in the dataset

**Steps:**

#Load this dataset into the python through pandas

**INPUT:**

# Import necessary libraries

import pandas as pd

**INPUT:**

# Load the dataset (replace 'sales\_dataset.csv' with your dataset file)

df = pd.read\_csv('sales\_data\_with\_discounts.csv')

#Identify which columns are numerical

**INPUT:**

# Identify numerical columns

Numeric\_Sales= df.select\_dtypes(include=['int64', 'float64']).columns

Numeric\_Sales

**OUTPUT:**

Index(['Volume', 'Avg Price', 'Total Sales Value', 'Discount Rate (%)',

'Discount Amount', 'Net Sales Value'],

dtype='object')

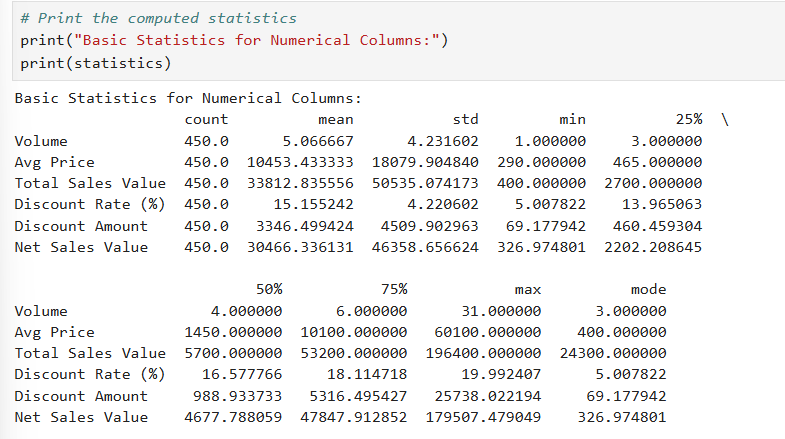
**Mean, median, mode and standard deviation of numerical columns have to be calculated**

**INPUT:**

# Calculate mean, median, mode, and standard deviation

statistics = df[Numeric\_Sales].describe().transpose()

statistics['mode'] = df[Numeric\_Sales].mode().iloc[0] # Mode as the most frequent value



**Brief:**

**Volume:**

The mean of the volume is 5, including that the average volume is 5. The median of the volume is 4, Showing that half of the people buy above 4 and below volume of products. The mode is 0,3, which is the most common number of products that people buy. The Standard deviation of the volume is 4.23 this indicates that 68% people buy products of volume within deviation 4.23 from the median of the volume. most of the people buy products of the volume within the range of 2 to 8.

**Avg Price:**

The mean of the avg price is 10453.4, including that the average price of the sales is 10453.4. The median of the avg price is 1450, Showing that half of the price of sales is 1450 above and below avg sales prize of products.The modes are 400,450,500,1300,8100, which are the most common average prize of the sale products.The Standard deviation of the avg prize is 18079.9 this indicates that 68% of avg sales price lies within deviation 18079 from the median of the avg price.

**Total Sales Value:**

The mean of the total sales value is 33812.835, including that the average of total sales value is 33812.8. The median of the total sales value is 5700, Showing that half of the totalsales value is below 5700 and another half is above 5700. The mode is 24300 , which is the most common value of the total sales value. The Standard deviation of the total sales value is 50535.07, this indicates that 68% of total sales value lies within deviation 50535.07 from the median of the total sales value.

**Discount Rate (%):**

The mean of the discount rate in percentage is15.155, including that the average of discount rate is 15.155 . The median of the total sales value is 16.577, Showing that half of the discount rate is below 16.577 and another half is above 16.577.The Standard deviation of the discount rate is 4.22,this indicates that 68% of discount rate lies within deviation 4.22 from the median of the discount rate.

**Discount Amount:**

The mean of the discount amount is 3346.499, including that the average of discount amount is 3346.499.the median of the discount amount is 988.933, Showing that half of the discount amount is 988.933below and another half is above 988.933.The Standard deviation of the discount rate is 4509.902,this indicates that 68% of discount amount lies within deviation 4509.902 from the median of the discount rate.

**Net Sales Value:**

The mean of the net sales value is 30466.336, including that the average of net sales value is 30466.336. The median of the net sales value is 4677.788, Showing that half of the net sales value is 4677.788 below and another half is above 4677.788. The Standard deviation of the net sale value is 46358.656, this indicates that 68% of discount amount lies within deviation 46358.656 from the median of the net sales value.

**2. Data Visualization**

**Objective:**

To visualize numerical and categorical variables distribution as well as relationships among them based on this dataset.

**Histograms:**

**INPUT:**

import matplotlib.pyplot as plt

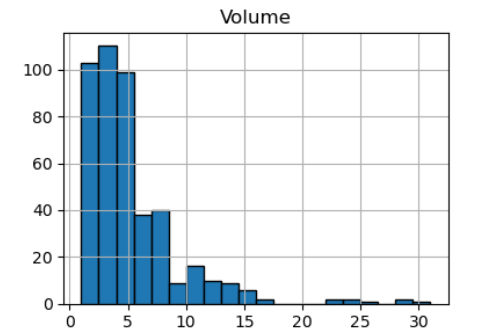
**# Plot histograms for numerical columns**

**INPUT:**

plt.hist(Numeric\_Sales["Volume"])

**OUTPUT:**

**Histogram for volume:**



Skewedness:2.731

Kurtosis:10.258

**Inference:**

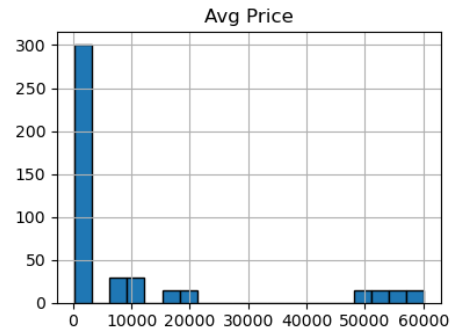
From the given values, the histogram looks to the right, which is positive skewness, and it has heavy tails, which means high kurtosis, therefore corresponding to non-symmetric distribution with outliers.

**INPUT:**

plt.hist(Numeric\_Sales["Avg Price"])

**OUTPUT:**

**Histogram for Avg price:**



Skewedness:1.9088

Kurtosis:2.07565

**Inference:**

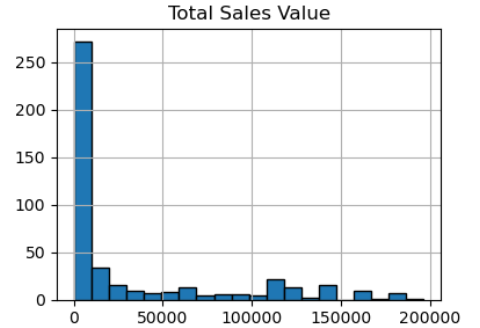
It is right-skewed with a medium tail, indicating that the data is skewed to the right, possibly with a long tail extending to the right side of the mean.

**INPUT**:

plt.hist(Numeric\_Sales["Total Sales Value"])

**OUTPUT**:

**Histogram for Total Sales Value:**



Skewedness:1.5347

Kurtosis:1.024916

**Inference:**

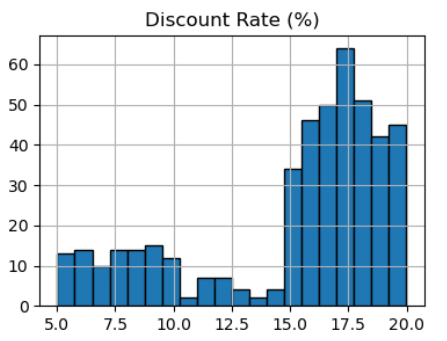
Using a skewness value of 1.5347 and a kurtosis measure of 1.024916, it is right skewed.

**INPUT:**

plt.hist(Numeric\_Sales["Discount Rate (%)"])

**OUTPUT:**

**Histogram for Discount Rate(%):**



Skewedness: -1.0622

Kurtosis: -0.1785

**Inference:**

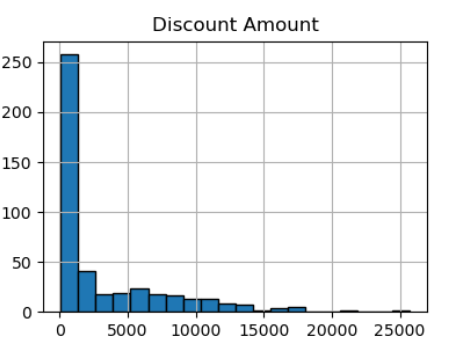
The histogram thus appears slightly skewed to the left, considering a skew value less than 0, and has a comparatively flat top and no outliers at all, considering kurtosis less than 0.

**INPUT:**

plt.hist(Numeric\_Sales["Discount Amount"])

**OUTPUT:**

**Histogram for Discount Amount:**



Skewedness:1.91303

Kurtosis:3.8311

**Inference:**

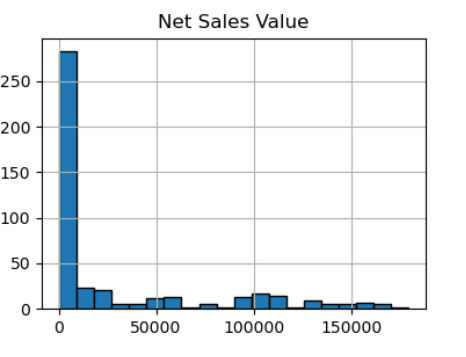
The data appears to be positively skewed—that is, the high values at the end stretch out the mean from the median. The high kurtosis value indicated that the data is heavily tailed, indicating extreme values far in excess of the mean and a concentration of the distribution around those extreme values.

**INPUT:**

plt.hist(Numeric\_Sales["Net Sales Value"])

**OUTPUT:**

**Histogram for Net Sales Value:**



Skewedness:1.5408

Kurtosis:1.01246

**Inference:**

The histogram reveals a fundamental shape, slightly skewed to the right, hence specifying that most of the information is concentrated toward the left side of the mean, with a few outliers toward the right. The fairly low kurtosis value indicates that the shape of the distribution is basically normal, with most data points clustered around the mean.

**BOXPLOTS:**

Identify outlying values and the range between quartiles using boxplots for numerical variables.

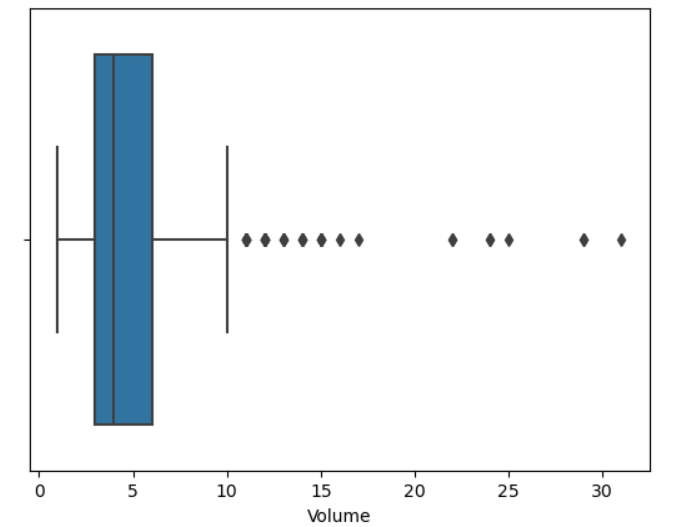
# Assuming 'df' is the DataFrame containing the data

**INPUT:**

sns.boxplot(Numeric\_Sales["Volume"])

**OUTPUT:**

**Boxplot for Volume:**



Quartile -1(Q1) =3 whisker length:

Quartile-2(Q2) =4 lower whisker length =1.5

Quartile-3(Q3) =6 upper whisker length =10.5

**Inference:**

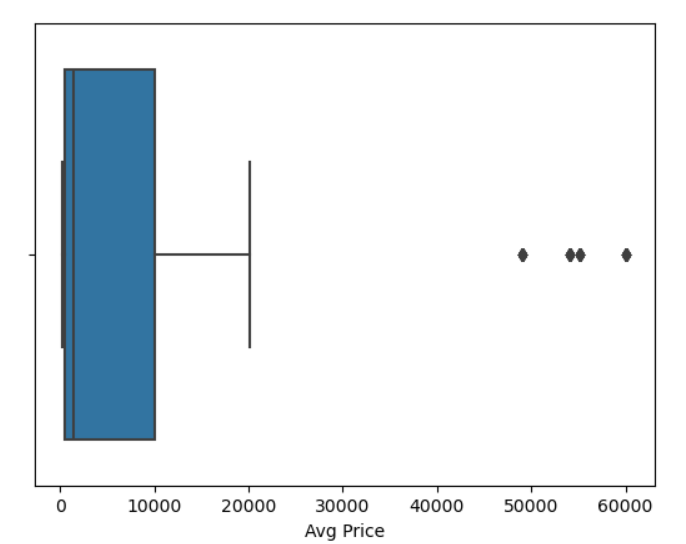
The middle 50 percent of the data is from 3 to 6. Most of the values cluster around 4-5. Outliers suggest Q1 and Q3, indicating that most of the data shifts toward the low end with a long tail up to 10.5.

**INPUT:**

sns.boxplot(Numeric\_Sales["Avg Price"])

**OUTPUT**:

**Boxplot for Avg Price:**



Quartile -1(Q1) =465 whisker length:

Quartile-2(Q2) =1450 lower whisker length=-1398.5

Quartile-3(Q3) =10100 upper whisker length=24552.5

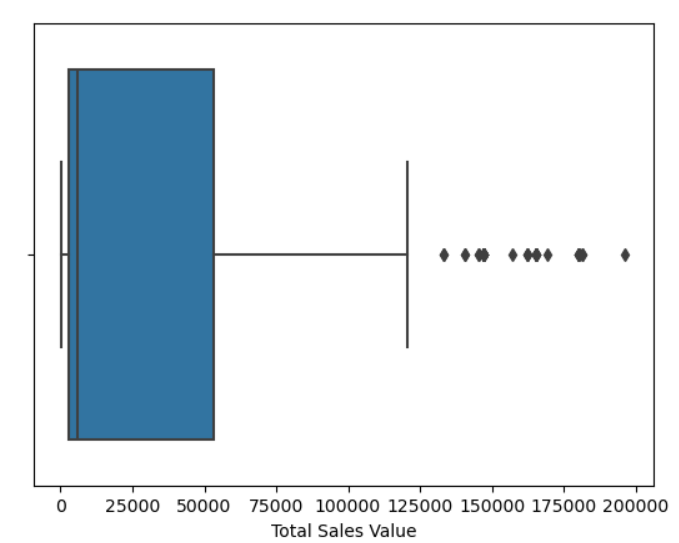
Inference: The length of the upper whisker is much shorter than that of the lower whisker, suggesting that extreme values may occur at large numbers at the lower end of data.

**INPUT:**

sns.boxplot(Numeric\_Sales["Total Sales Value"])

**OUTPUT:**

**Boxplot for Total Sales Value:**



Quartile -1(Q1) =2700 whisker length:

Quartile-2(Q2) =5700 lower whisker length=73050

Quartile-3(Q3) =53200 upper whisker length=128950

**Inference:**

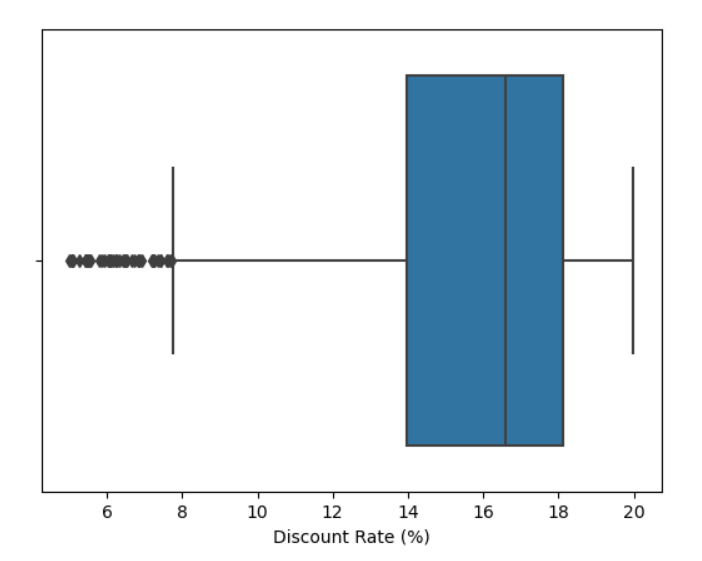
The data is skewed to the right, with a long tail reaching out toward higher values. Most of the data points Q1 and Q3 lie relatively low, but some extreme values travel quite high to form the upper whisker and lower whisker.

**INPUT:**

sns.boxplot(Numeric\_Sales["Discount Rate (%)"])

**OUTPUT:**

**Boxplot for Discount Rate(%):**



Quartile -1(Q1) =13.96 whisker length:

Quartile-2(Q2) =16.57 lower whisker length=7.735

Quartile-3(Q3) =18.11 upper whisker length=24.335

**Inference:**

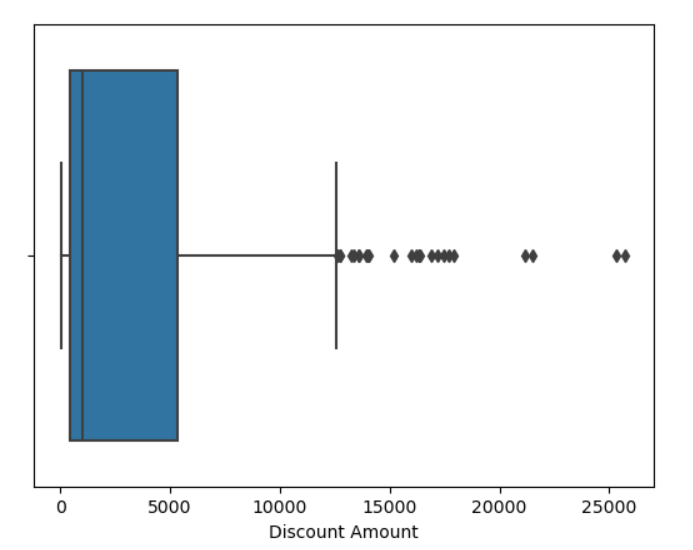
The data appeared skewed to the right, although most values are very centrally located. The upper whisker told that all information lied below 24.34, indicating a majority of the values below this threshold.

**INPUT:**

sns.boxplot(Numeric\_Sales["Discount Amount"])

**OUTPUT:**

**Boxplot for Discount Amount:**



Quartile -1(Q1) =460.45 whisker length:

Quartile-2(Q2) =988.93 lower whisker length=6823.61

Quartile-3(Q3) =5316.49 upper whisker length=12600.55

**Inference:**

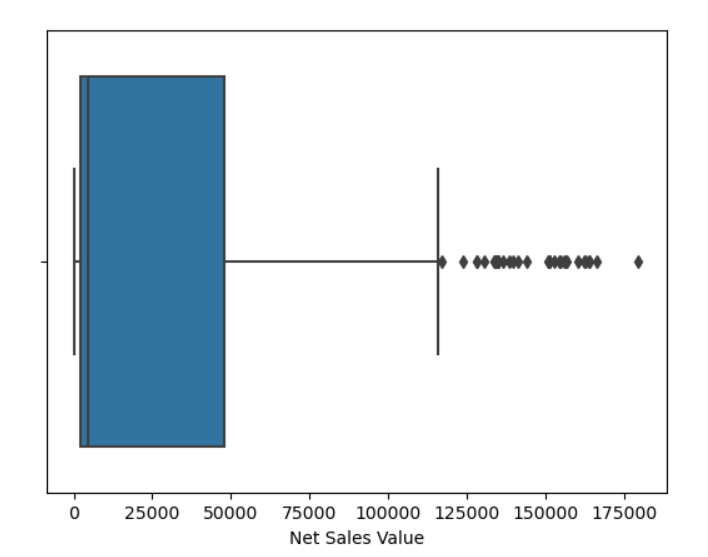
Most of the data focuses in on 460-600, and at the extreme end, there are a large number of outliers for 6,000 to 13,000.

**INPUT:**

sns.boxplot(Numeric\_Sales["Net Sales Value"])

**OUTPUT:**

**Boxplot for Net Sales Value:**



Quartile -1(Q1) =2202.20 whisker length:

Quartile-2(Q2) =4677.78 lower whisker length=66266.365

Quartile-3(Q3) =47847.91 upper whisker length=116316.465

**Inference:**

Q1, Q2, and Q3 all indicate that the data is skewed to the right, with a long tail of high values. The upper whisker's length way exceeds the maximum value of Q3, so there must be some extremely large values in the dataset.

**ANALYSIS OF BAR CHARTS FOR CATEGORICAL COLUMNS:**

**INPUT:**

# Plot bar charts for categorical columns

**INPUT:**

import seaborn as sns

import matplotlib.pyplot as plt

categorical\_columns = [ 'Day', 'SKU', 'City', 'BU', 'Brand', 'Model']

for col in categorical\_columns:

plt.figure() # Create a new figure for each plot

sns.countplot(x=col, data=df)

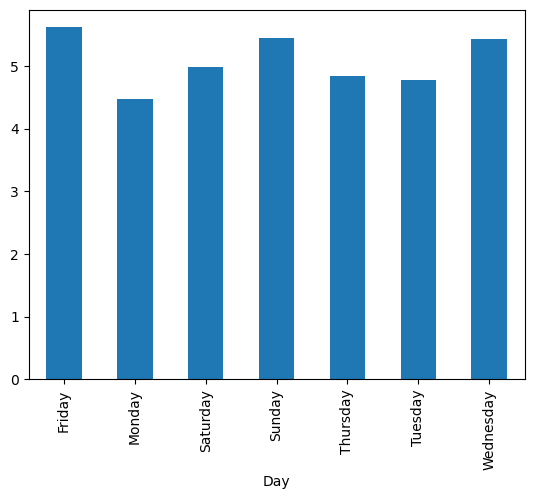
plt.title(f'Bar Plot for {col}')

plt.xlabel(col)

plt.ylabel('Count')

plt.xticks(rotation=90) # Rotate x-axis labels for better readability if needed

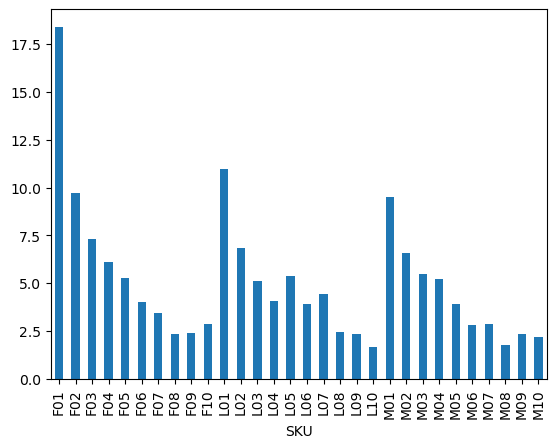
**Bar graph for Days:**



**Inference:**

The bar plot shows the sales of the days. The bar plot shows that the highest sales are on Friday and lowest on Monday. This suggest that our sales rate is more on weekdays only i.e on Friday,Saturday and Sunday. So, to hold more customers we may need to add more staff on weekends to improve the sales rate.

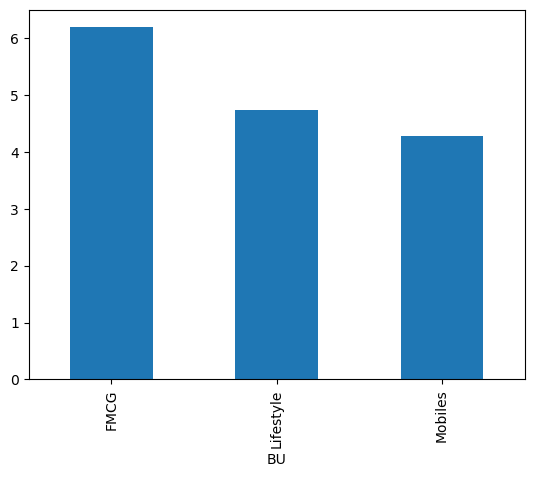
**Bar graph for SKU:**



**Inference:**

Here, Bar graph shows the sku of the sales data. The graph represents performance of sku that is stoke keeping unit on x-axis based on the scale on y-axis. we have 3 types of sku on the graph is FO,LO,MO all these three have 10 models each. The highest peak value is FO1 and it gradually decrease from 1st model to 10th model. Second highest peak is LO1 It also gradually decrease from LO1 model to LO10 model and finally third highest peak is MO1 similarly, These also gradually decrease from MO1 to MO10 model.

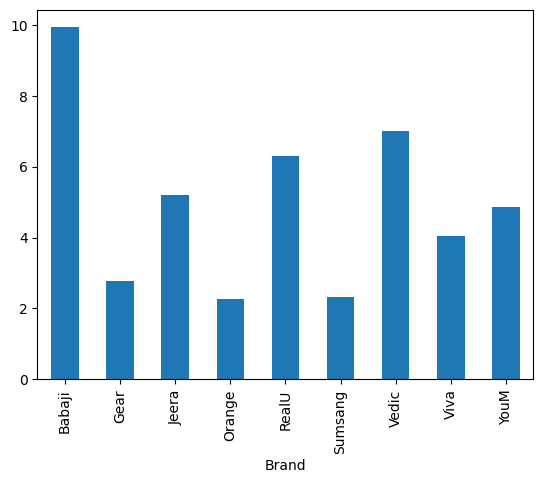
**Bar graph for BU:**



**Inference:**

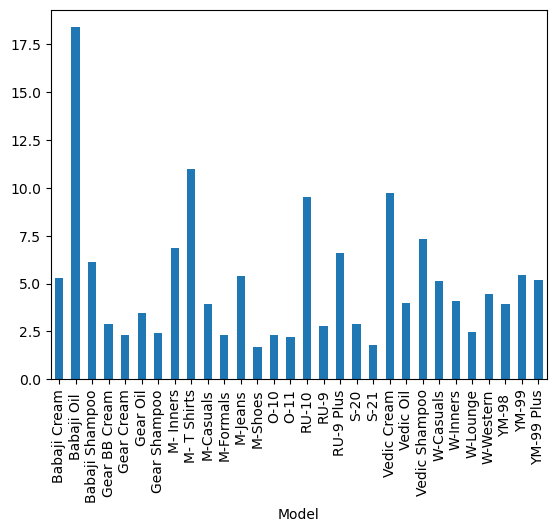
The Bar graph represents the BU and sales. on x-axis It take BU that business units of Products and it take unit of sales on y-axis. we have three different bu products on the graph one is FMCG products second is life Lifestyle products and third one is Mobile products. Sales of FMCG products is more when compared to other two. Mobiles are the lowest purchased products and lifestyle products are purchased moderately by the customers.

**Bar graph for Brand:**



**Inference:**

In this bar grand we have taken different brands of products on x-axis and sales rate o y-axis. Here, we have 9 different brands. The highest sailed brand is Babaji brand and the lowest sailed brand is orange. second highest brand is Vedic, third highest brand is RealU, fourth highest is jeera brand. hence, we have to be on more prioritize with the highest brand that is Babaji brand product stoke should be more in the sales shop.



**Inference:**

The bag graph shows the different models of products and its sale rate. On x-axis different Products models and on y-axis sales rate is given. we have total 30 models on the bar graph. from these the highest sales product is Babaji oil which is above 1.75 sales range, the second highest product is M- Tshirts, Third highest product is Vedic cream and fourth highest one is RU10.Mshoes are the lowest models that means it sales rate is less compared to others.so we need to invest less on RU10 product.

**3. Standardization of Numerical Variables**

**Objective**:

Numerical variables should be normalized into z-scores.

**Explanation for standardization:**

Standardization, also called z-score normalization, is a statistical technique for changing a dataset into a standard normal distribution with a mean of 0 and a standard deviation of 1. This type of transformation is used to compare data or analyse it with any other data set, as well as to improve the performance of machine learning algorithms.

This is how it works:

**Z-Score:**

z-score is a statistical expression for the number of standard deviations an observation is away from the population mean.

**INPUT:**

# Standardization using z-score normalization

from sklearn.preprocessing import StandardScaler

# Create a scaler object

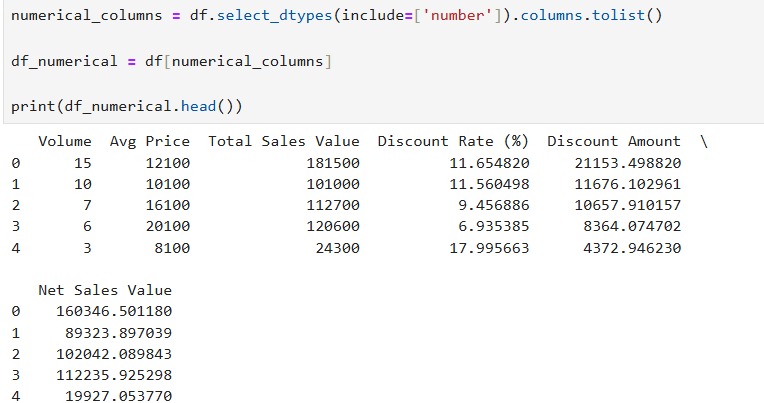
scaler = StandardScaler()

# Fit and transform the numerical columns

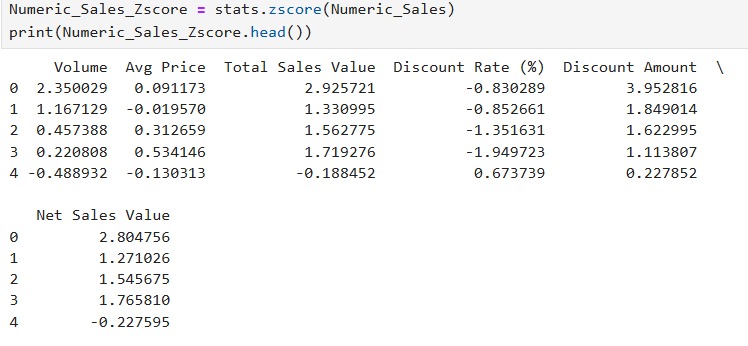
df[Numeric\_Sales] = scaler.fit\_transform(df[numerical\_columns])

# Show before and after comparisons of the data distributions (only for numerical columns)

**Numerical variables Before standardization:**



**Numerical variables After standardization:**



**4. Conversion of Categorical Data into Dummy Variables**

**Objective:**

Convert Categorical variables to dummies (one-hot encoding)

**Steps:**

One-hot encoding is one of the methods of changing categorical variables to be given to algorithms, in this fashion; the value of the categorical variable is given out in a binary vector representation, with the bit equal to 1 at a position of the respective category and otherwise filling with 0s.

So, we need to convert the categorical data into dummy variables

**INPUT:**

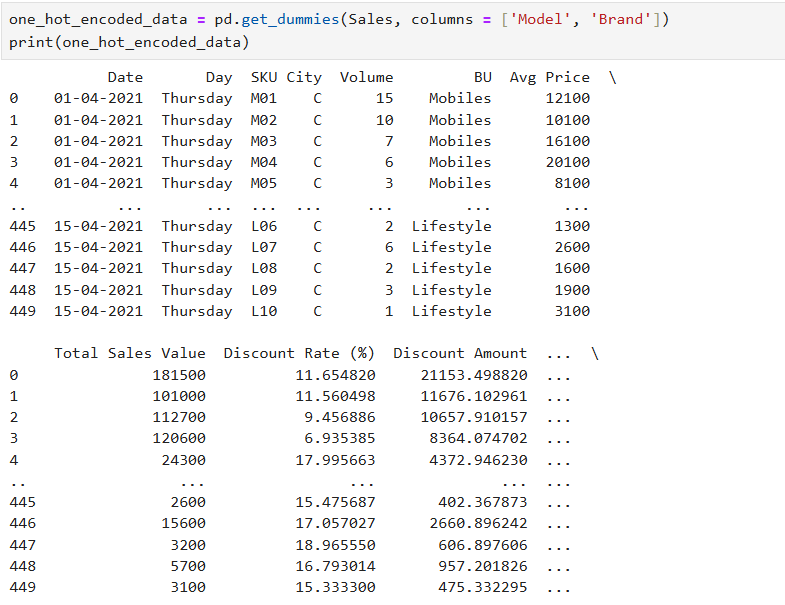
**# Apply one-hot encoding to categorical columns**

df = pd.get\_dummies(df, columns=categorical\_cols, drop\_first=True)

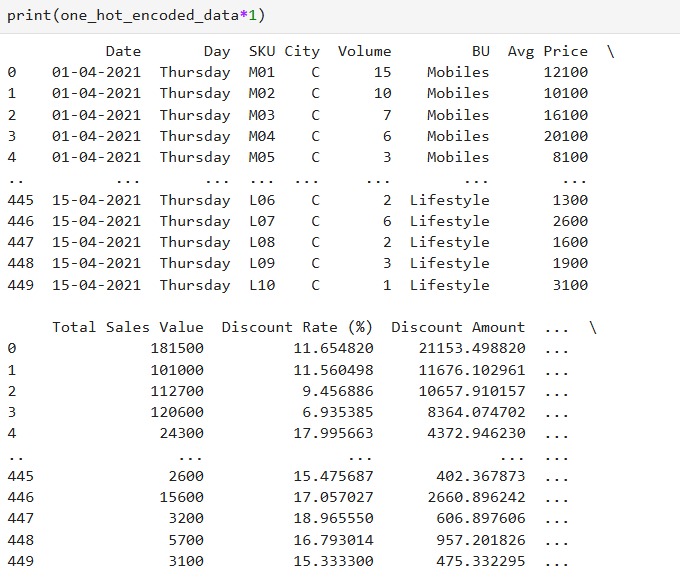
# Display a portion of the transformed dataset

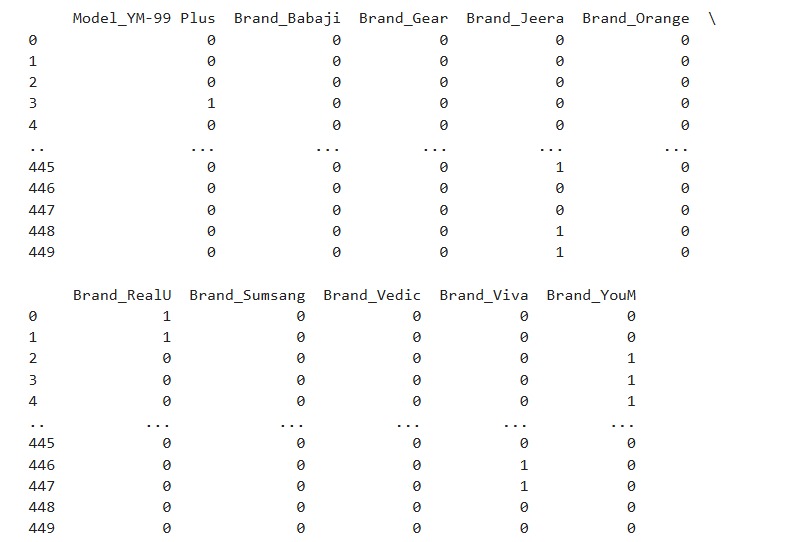
**OUTPUT:**

**Before:**



**After:**





**Conclusion:**

In this Assignment I have applied Python and pandas to conduct descriptive analytics by calculating statistical measures and visualizing data distributions. I also learned about pre-processing methods such as standardization using Z-score and one hot encodings that are necessary when preparing data before further analysis or modelling in machine learning. This sequence provided me a good base to my understanding.