**Descriptive Analytics for Numerical Columns**

**Objective:** To compute and analyse basic statistical measures for numerical columns in the dataset.

**Volume Feature:**

* **Mean (5.07)**: The average volume across the dataset is 5.07 units.
* **Median (4.0)**: Half of the volume values are below 4.0 and the other half are above. The fact that the median is lower than the mean suggests that the data might be right-skewed, meaning there are some higher volume values that pull the mean up.
* **Mode (3)**: The most frequent volume value is 3, indicating that this value occurs more often than any other.
* **Standard Deviation (4.23)**: On average, the volume values deviate from the mean by approximately 4.23 units.
* **Variance (17.91)**: The spread or dispersion of the volume values is 17.91, which aligns with the relatively high standard deviation, showing variability.

**Avg Price Feature:**

* **Mean (10,453.43)**: The average price across the dataset is 10,453.43. The relatively high mean indicates that the average prices are quite large.
* **Median (1,450)**: Half of the prices are below 1,450, which is much lower than the mean. This suggests that the price data is heavily right-skewed, with some extremely high values inflating the mean.
* **Mode ([400, 450, ...])**: Several price values are equally frequent, suggesting multiple clusters or common price points.
* **Standard Deviation (18,079.9)**: The price values fluctuate widely, with an average deviation of about 18,079.9 from the mean, indicating high price variability.
* **Variance (326,882,959.02)**: The extremely high variance further confirms the significant spread in the price data.

**Total Sales Value Feature:**

* **Mean (33,812.84)**: The average total sales value is 33,812.84, indicating a high sales volume.
* **Median (5,700)**: Half of the sales values are below 5,700, which is much lower than the mean. Like the Avg Price feature, this suggests right-skewed data with a few high sales values pulling up the mean.
* **Mode (24,300)**: The most frequent total sales value is 24,300.
* **Standard Deviation (50,535.07)**: There is significant variability in total sales values, with deviations averaging 50,535.07 units from the mean.
* **Variance (2,553,793,721.63)**: This large variance indicates a wide spread in total sales values, consistent with the high standard deviation.

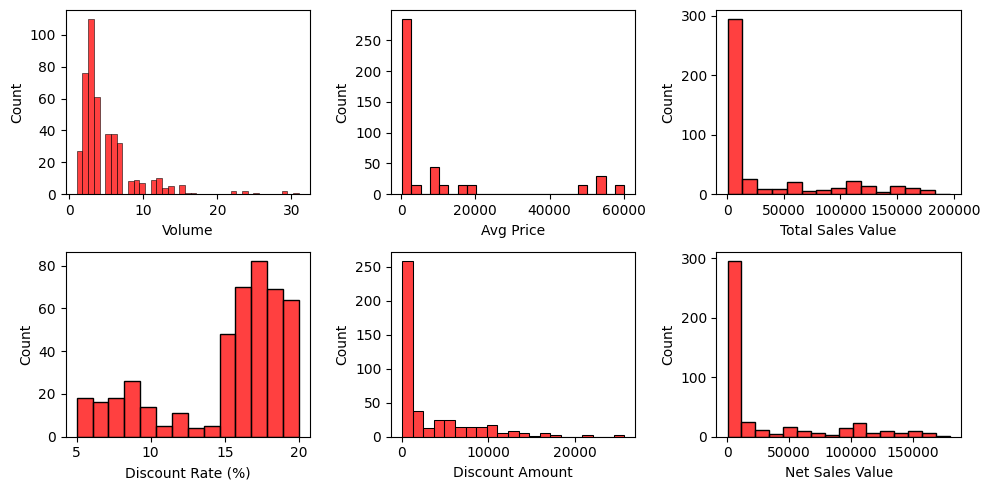
**Discount Rate (%) Feature:**

* **Mean (15.16%)**: The average discount rate is around 15.16%.
* **Median (16.58%)**: Half of the discount rates are below 16.58%, slightly higher than the mean, suggesting a left-skewed distribution (possibly lower discounts being more frequent).
* **Mode (multiple values)**: A large number of values are equally frequent, indicating there may not be a single dominant discount rate.
* **Standard Deviation & Variance**: With a detailed mode and a fairly small range, the discount rate seems to vary slightly, though specific numbers were not provided. However, the feature seems tightly clustered with many similar values.

**Data Visualization**

**Objective:** To visualize the distribution and relationship of numerical and categorical variables in the dataset.

**Histograms**



**Observations:**

**1. Discount Amount:**

* **Observation**: The distribution is highly skewed to the right. The majority of discount amounts are clustered around 0, with very few instances of higher discount amounts.
* **Interpretation**: Most of the transactions have either no discount or a very low discount. This suggests that high discount offers are rare.

**2. Total Sales Value:**

* **Observation**: This histogram is also right-skewed. A significant number of transactions have total sales values near 0, with a long tail extending towards higher values.
* **Interpretation**: Many transactions are for low-value sales, while fewer transactions involve larger sales amounts. This could indicate that smaller sales are more common in your dataset, or that there is a wide range in transaction values.

**3. Average Price:**

* **Observation**: There are two distinct peaks in this histogram. The first peak is at 0 or close to 0, and the second one is in the range of 50,000-60,000. The distribution has gaps between these peaks.
* **Interpretation**: This bimodal distribution suggests two clusters in the data. There may be a significant number of low-priced products alongside a segment of high-priced products. These could represent different categories of items or services.

**4. Volume:**

* **Observation**: The volume distribution is right-skewed, with most transactions involving a volume between 0 and 5, and a few reaching values higher than 20.
* **Interpretation**: The majority of transactions involve small quantities or volumes. Higher volumes are rare, which may indicate that most of the transactions are for small or moderate amounts of product.

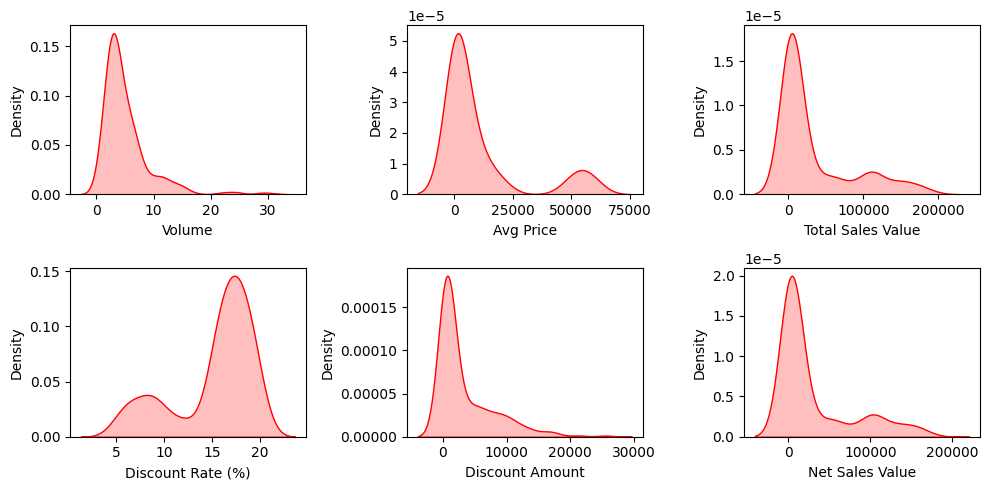
**5. Net Sales Value Histogram:**

* **Observation**: Most sales are below ₹25,000, with a sharp drop in frequency as values increase. Very few sales exceed ₹100,000.
* **Interpretation**: The business likely handles many small transactions, while high-value sales are rare. Efforts to increase premium sales could target higher-paying customers.

**6. Discount Rate Histogram:**

* **Observation**: Two main discount clusters appear: one around 5-10% and another at 15-20%, with the highest frequency between 16-18%.
* **Interpretation**: This suggests distinct pricing strategies. The frequent use of 16-18% discounts might indicate its effectiveness in driving sales.

**KDE Plots**

****

**Observations:**

1. **Volume**:

* Most transactions occur with low volumes, peaking around 0-5 units, with very few instances beyond 10 units.

2. **Average Price**:

* The majority of products have an average price clustered below ₹25,000, with a sharp drop thereafter. Some higher-priced products exist, but they're rare.

3. **Total Sales Value**:

* Sales are concentrated at lower values, with a peak below ₹50,000. High total sales values are infrequent.

4. **Discount Rate**:

* Two clusters are visible: one around 5-10% and a more dominant peak at 15-20%, with the majority of discounts falling in the higher range.

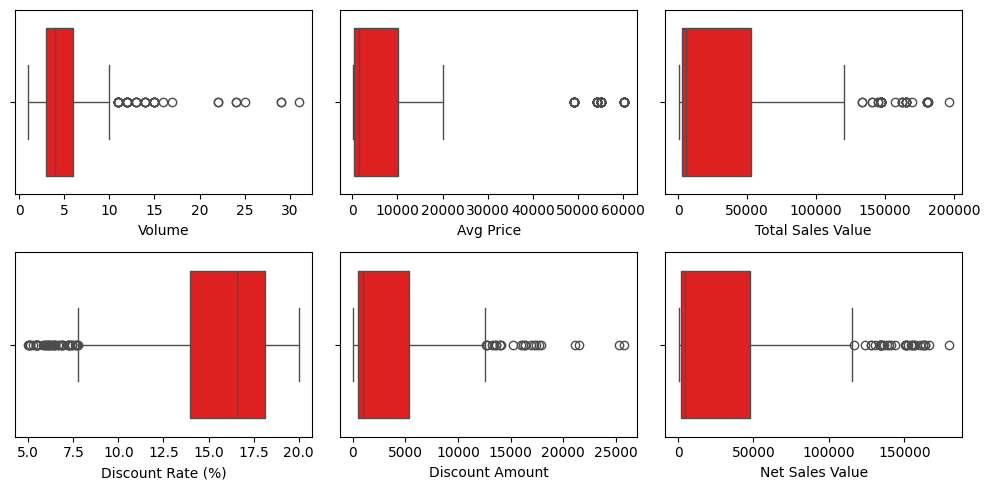
5. **Discount Amount**:

* Most discounts are below ₹10,000, with very few transactions receiving larger discounts.

6. **Net Sales Value**:

* The distribution is heavily skewed toward lower sales values, with a peak below ₹25,000, and a long tail for higher values.

**Box Plots**

****

**Observations**

1. **Volume:**

* Most transactions have low volumes (0-5 units)
* There are multiple outliers extending up to around 30 units
* The distribution is right-skewed
* The box (interquartile range) appears to be compressed, suggesting most values are concentrated in the lower range

1. **Average Price:**

* Majority of prices cluster in the lower range (below 20,000)
* Several outliers extend up to around 60,000
* Right-skewed distribution
* The median appears to be relatively low compared to the full range

1. **Total Sales Value:**

* Most values concentrate in the lower range (below 50,000)
* Multiple outliers extend up to about 200,000
* Strongly right-skewed distribution
* Large gap between typical values and outliers

1. **Discount Rate (%):**

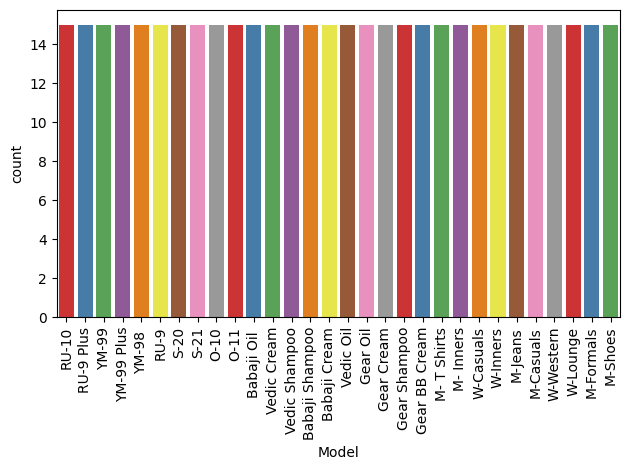
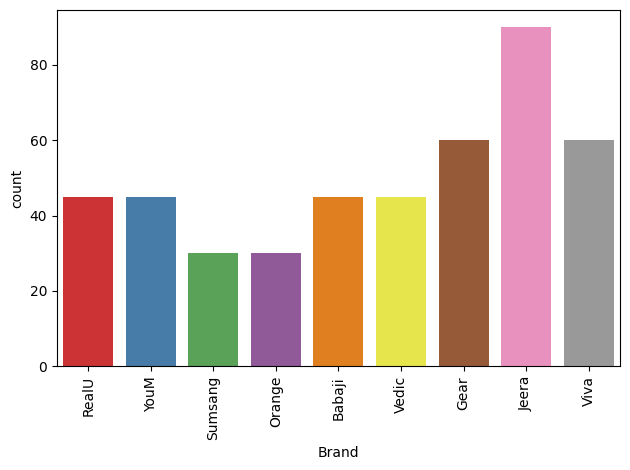
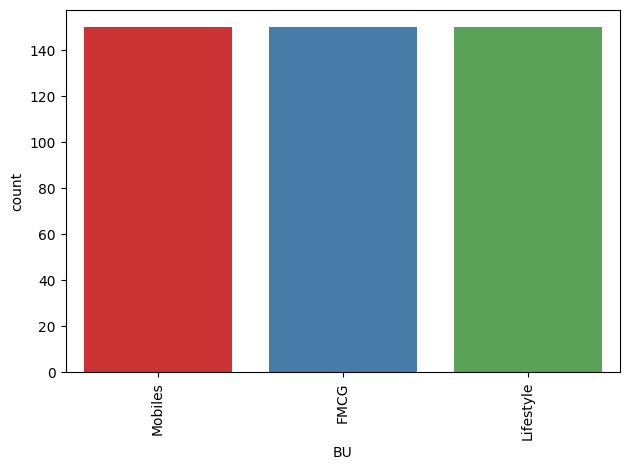
* Interesting bi-modal distribution
* One cluster around 5-10%
* Another cluster around 15-20%
* Few values in the middle range (10-15%)
* Distribution appears more symmetric than the others

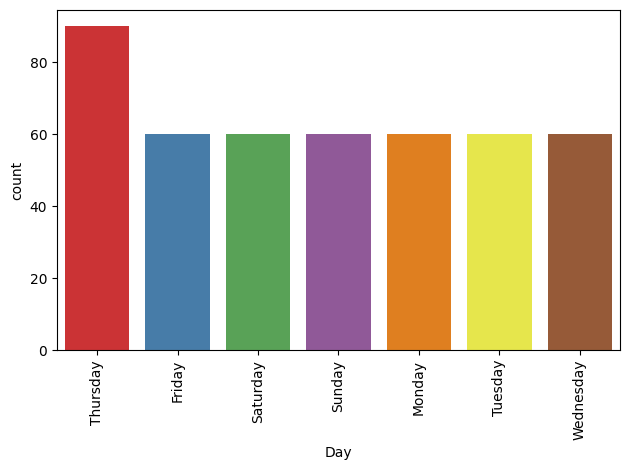
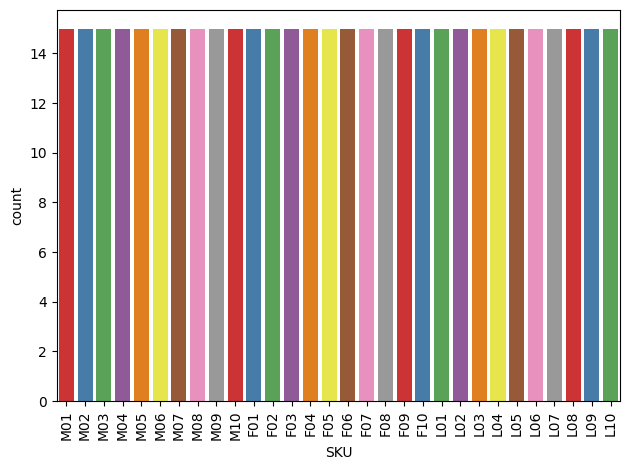
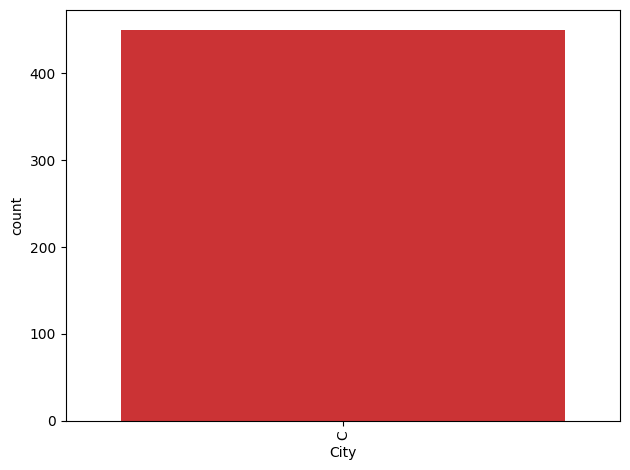
1. **Discount Amount:**

* Most discount amounts are in the lower range (below 5,000)
* Several outliers extend up to about 25,000
* Right-skewed distribution
* Clear separation between typical values and outliers

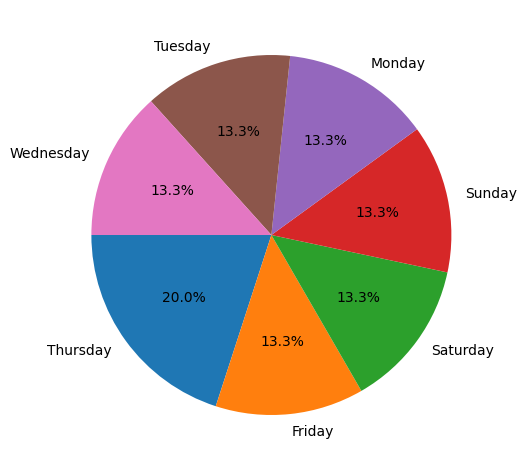
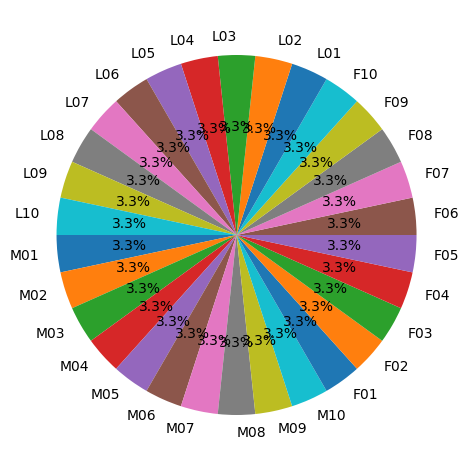
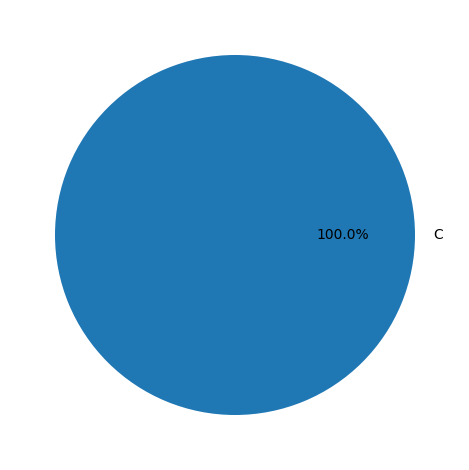
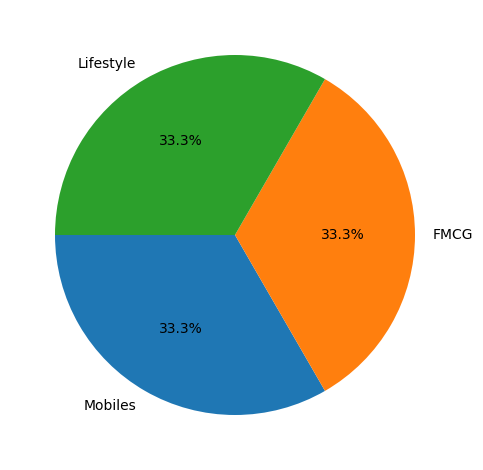
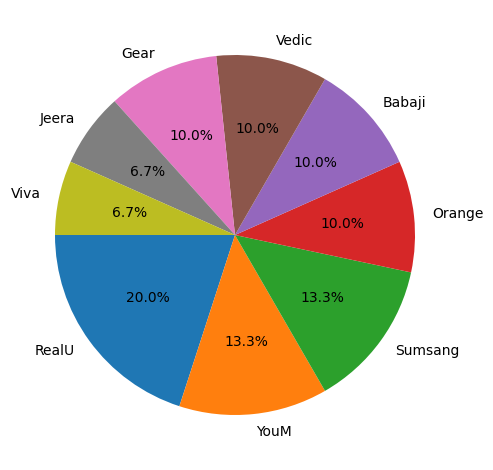
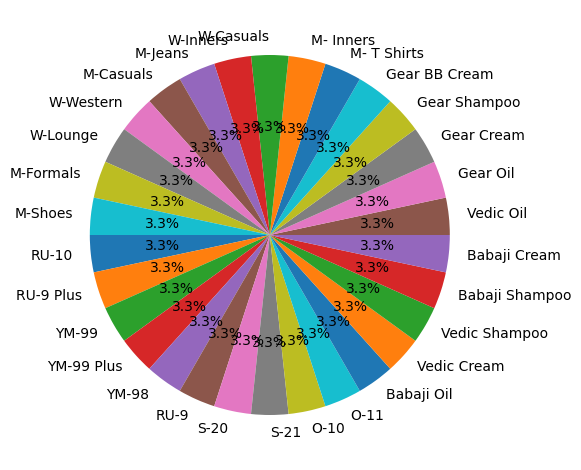
1. **Net Sales Value:**

* Concentrated in lower range (below 50,000)
* Multiple outliers extending to about 150,000
* Right-skewed distribution
* Similar pattern to Total Sales Value but with slightly lower values overall

**Bar/Count Charts **

****

**Pie Charts**

****

**Standardization of Numerical Variables**

**Objective:** To scale numerical variables for uniformity, improving the dataset’s suitability for analytical models.

**Standardization**, also known as **Z-score normalization**, is a technique used to rescale data so that it has a mean of 0 and a standard deviation of 1. This is especially useful in machine learning algorithms that rely on distance measures (e.g., KNN, SVM, logistic regression) or when features are on different scales.

**Formula for Standardization (Z-Score Normalization):**

For each value ​ in a dataset, the standardized value is calculated as:

Where:

* ​ = The original data point.
* μ = Mean of the feature (the average value).
* σ = Standard deviation of the feature (a measure of spread).

**Key Points:**

1. **Mean of 0**: After standardization, the transformed data will have a mean of 0.
2. **Standard Deviation of 1**: The spread of the data will be in terms of standard deviation, with a variance of 1.
3. **Centers the Data**: It shifts the distribution of the data to be centered around the mean, making it easier to compare features of different scales.

**Example:**

If you have a dataset of heights and weights:

* Heights may range between 150 cm and 200 cm.
* Weights may range between 50 kg and 100 kg.

These two features are on different scales, which could cause problems in some algorithms. After standardization:

* Heights will have a mean of 0 and standard deviation of 1.
* Weights will also have a mean of 0 and standard deviation of 1.

**Why Standardization is Important:**

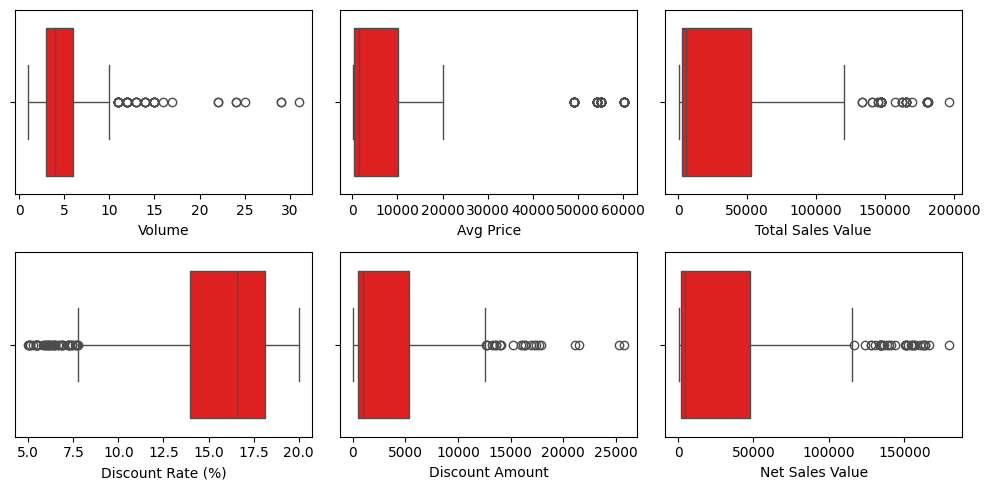
1. **Comparing Features on the Same Scale**: Some models (like distance-based algorithms or gradient-based methods) are sensitive to the scale of the features. Standardization ensures that all features are treated equally.
2. **Faster Convergence**: In optimization algorithms, like gradient descent, standardizing data can help achieve faster convergence because features with very different scales might slow down the learning process.
3. **Reduces Bias**: When features have different units or scales, models may give more weight to features with larger scales, leading to bias. Standardization mitigates this issue.

**When to Use Standardization:**

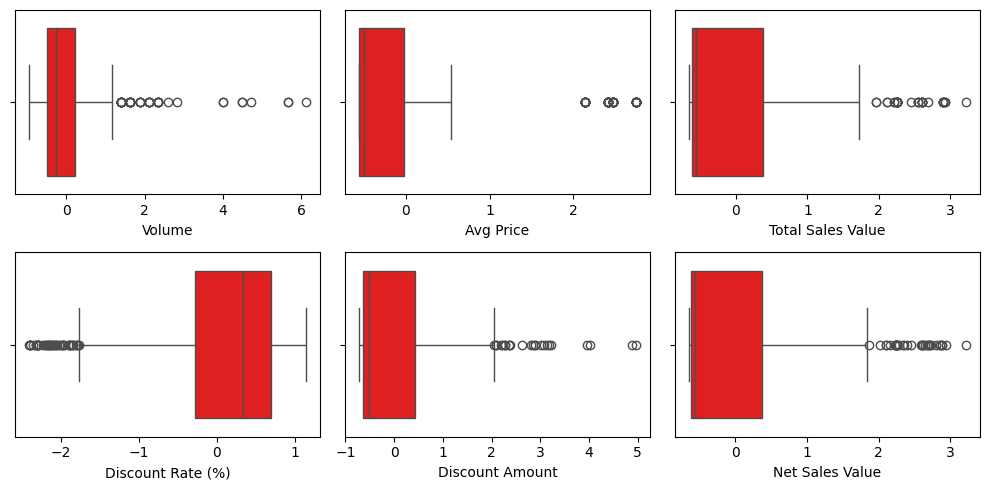
* **Distance-based algorithms**: e.g., K-nearest neighbors, K-means clustering.
* **Algorithms with regularization**: e.g., Ridge regression, Lasso regression.
* **When your data contains features on different scales**.

**Box Plots before vs after Standardization**

**Before**

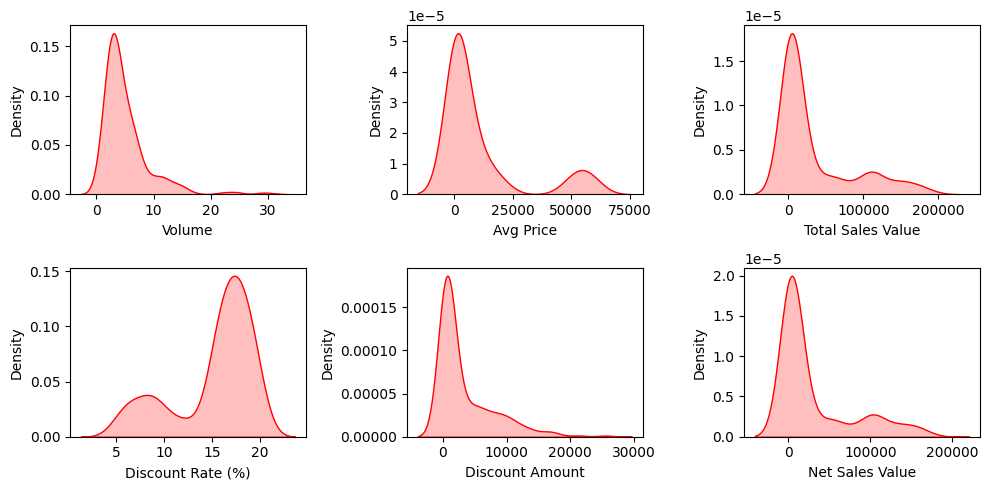
****

**After**

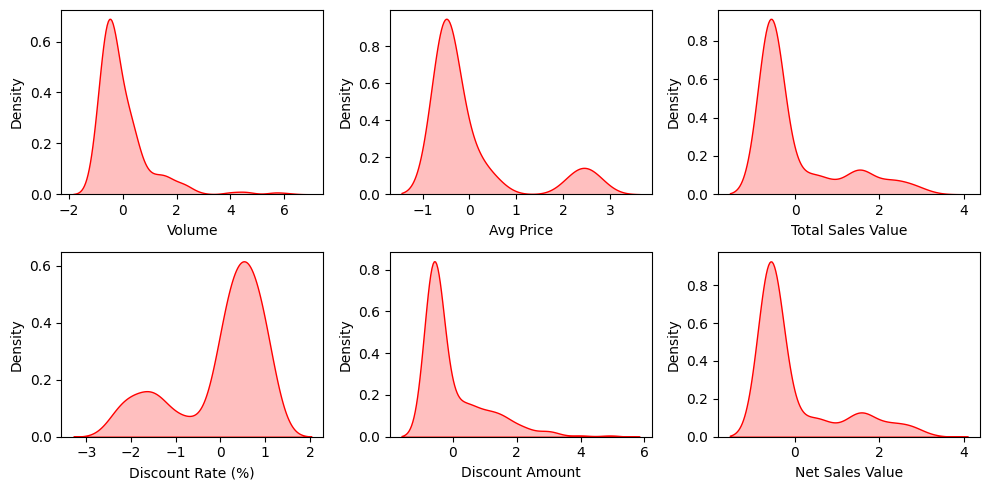
****

**Kde plots before vs after Standardization**

**Before**

****

**After**

****

**Conversion of Categorical Data into Dummy Variables**

**Objective:** To transform categorical variables into a format that can be provided to ML algorithms

Converting categorical data into dummy variables, often referred to as one-hot encoding, is an essential preprocessing step in machine learning and data analysis. Here’s an in-depth discussion of why this process is necessary:

**1. Machine Learning Algorithms Require Numerical Input**

Most machine learning algorithms, including linear regression, logistic regression, support vector machines, and neural networks, require numerical input. Categorical variables, which represent categories or labels (like "red," "blue," "green"), cannot be directly used in these algorithms. One-hot encoding transforms these categories into a numerical format that algorithms can process.

**2. Avoiding Ordinality Assumptions**

Categorical data does not have a natural order. For example, in the categorical variable "color," there is no ranking among "red," "blue," and "green." If you simply convert these categories to numbers (e.g., red = 1, blue = 2, green = 3), you create an artificial ordinal relationship where blue is considered greater than red. One-hot encoding avoids this issue by creating separate binary columns for each category, thereby eliminating any unintended ordinal interpretation.

**3. Improving Model Performance**

One-hot encoding can improve the performance of machine learning models. By providing distinct columns for each category, models can better capture the relationship between features and the target variable. For instance, in a classification task, one-hot encoded features allow the model to learn the unique contribution of each category rather than averaging them together.

**4. Handling Non-Binary Categories**

One-hot encoding is particularly useful for handling categorical variables with multiple levels or categories. For example, if a feature has five possible values, one-hot encoding will create five new binary features, each indicating the presence or absence of a specific category. This enables the model to learn from a richer set of features.

**5. Facilitating Interpretability**

One-hot encoding can enhance the interpretability of models. By representing categories as separate features, it becomes clearer how each category influences the predictions. For example, if a model is predicting sales based on marketing channels, one-hot encoding allows analysts to see how each channel (e.g., social media, email, TV) contributes independently to sales.

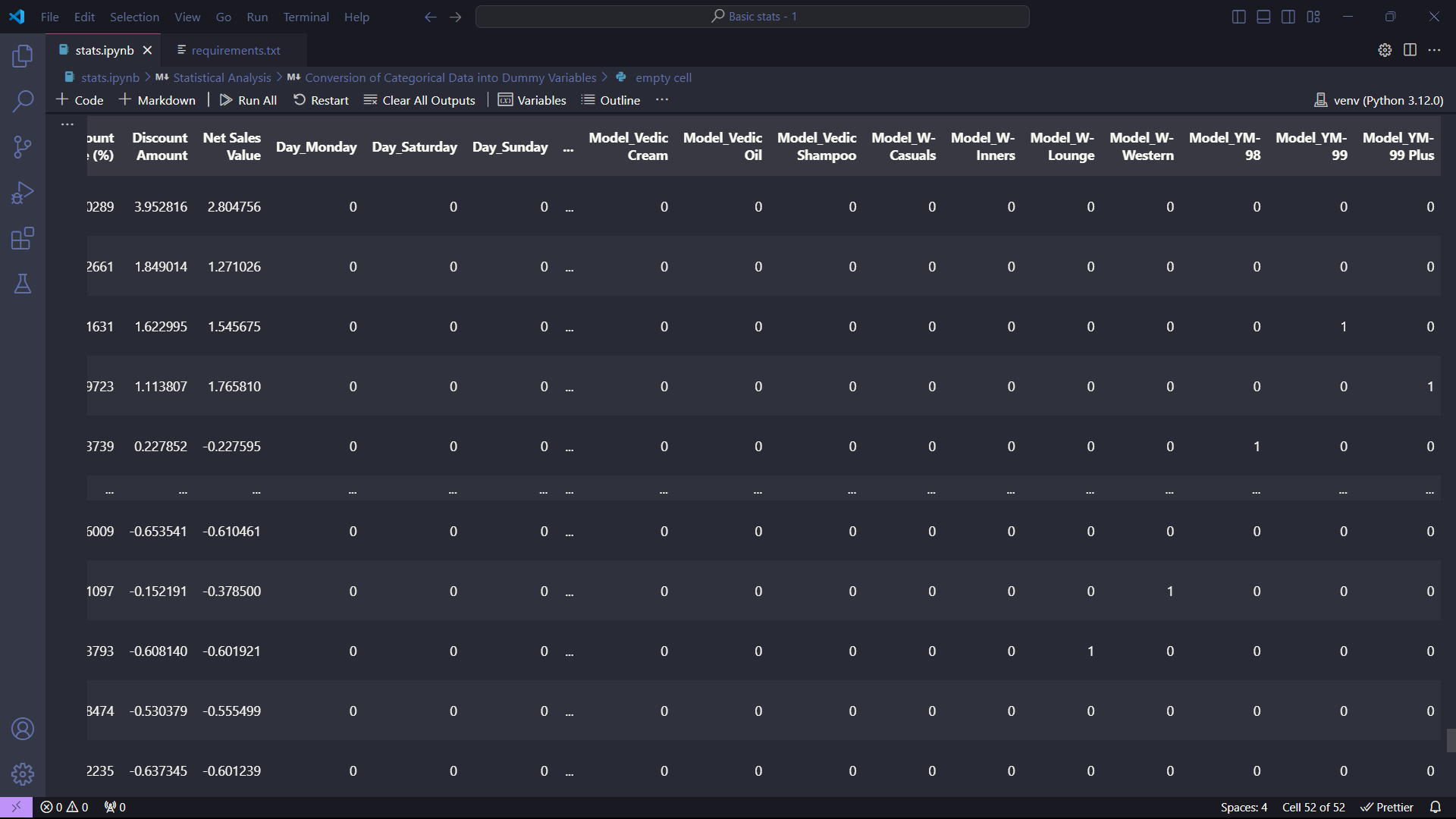
**6. Improved Handling of Categorical Variables in Tree-Based Models**

While tree-based algorithms (like decision trees, random forests, and gradient boosting) can handle categorical variables natively, one-hot encoding can still be beneficial. It allows for more granular splits based on the specific category, which can lead to improved accuracy.

**7. Limiting the Curse of Dimensionality**

One potential downside of one-hot encoding is that it can lead to a large number of features, especially if the categorical variable has many unique categories. This increase in dimensionality can sometimes result in overfitting. To mitigate this, techniques such as feature selection, dimensionality reduction (like PCA), or using target encoding can be considered.

**Dataset after OHE(One Hot Encoding)**



**Summary of Key Findings from Descriptive Analytics and Data Visualizations**

1. **Descriptive Statistics:**
   * **Mean, Median, and Mode**: These metrics provided insights into the central tendency of numerical columns, indicating where the bulk of the data points lie. For instance, if the mean and median are close, the distribution is likely symmetric; significant differences may suggest skewness.
   * **Standard Deviation**: This measure helped understand the variability within the numerical data. A high standard deviation indicates that the data points are spread out over a wider range of values, while a low standard deviation suggests they are clustered around the mean.
2. **Visualizations:**
   * **Histograms**: These plots revealed the distribution of numerical data, allowing for quick identification of normality, skewness, and potential outliers. For instance, a bell-shaped histogram indicates a normal distribution.
   * **Kernel Density Estimation (KDE) Plots**: These provided a smoother estimate of the distribution compared to histograms, offering more insight into the probability density of the data.
   * **Count Plots**: These visualized the frequency of categorical variables, highlighting the most common categories. For example, if a count plot for a categorical variable showed a dominant category, it would indicate that this category is significantly more prevalent than others.
   * **Pie Charts**: Though less favored in data analysis due to difficulty in precise comparisons, they visually represented the proportion of each category in categorical variables, useful for demonstrating relative sizes.
3. **Standardization**:
   * After standardizing the numerical columns, all features were brought to a similar scale, allowing models to interpret them appropriately. This step is crucial for distance-based algorithms like K-Nearest Neighbors (KNN) and gradient descent optimization techniques.
4. **One-Hot Encoding (OHE)**:
   * Converting categorical variables into a format suitable for modeling facilitated their inclusion in machine learning algorithms. OHE expanded categorical features into binary columns, enabling models to better capture the relationship between categories and the target variable.

**Importance of Data Preprocessing Steps**

1. **Standardization**:
   * **Model Performance**: Many machine learning models assume that data is normally distributed or that features are on the same scale. Standardization improves convergence in optimization algorithms and enhances model performance.
   * **Interpretability**: Standardized features make it easier to interpret coefficients in regression models since they are on the same scale.
2. **One-Hot Encoding**:
   * **Avoiding Ordinal Relationships**: By creating binary columns for each category, one-hot encoding prevents any unintended ordinal assumptions that might arise if categories were assigned numerical values directly.
   * **Enhanced Learning**: It allows algorithms to learn the unique contributions of each category without mixing information, improving accuracy and robustness.
   * **Flexibility**: OHE is applicable to various algorithms, making it a versatile tool in the preprocessing pipeline.

**Conclusion**

The combination of descriptive analytics and data visualization provided valuable insights into the dataset's characteristics. Key statistics and visualizations facilitated a better understanding of the data distribution, variability, and relationships among features. Following this, the preprocessing steps of standardization and one-hot encoding ensured that the data was appropriately prepared for machine learning applications. These steps are not just routine; they are foundational practices that significantly influence the performance and accuracy of predictive models. Properly preprocessed data leads to more reliable insights, improved model training, and ultimately better decision-making based on data analysis.