ASSIGNMENT-1

Descriptive Analytics and Data Preprocessing

Dataset: Sales Data with Discounts

1. Load the Dataset

```
In [1]: import pandas as pd
        df = pd.read_csv("sales_data_with_discounts.csv")
        df
```

Out[1]:

	Date	Day	SKU	City	Volume	BU	Brand	Model	Avg Price	Total Sales Value	Discount Rate (%)	Discount Amount	Net Sales Value
0	01-04-2021	Thursday	M01	C	15	Mobiles	RealU	RU-10	12100	181500	11.654820	21153.498820	160346.501180
1	01-04-2021	Thursday	M02	C	10	Mobiles	RealU	RU-9 Plus	10100	101000	11.560498	11676.102961	89323.897039
2	01-04-2021	Thursday	M03	C	7	Mobiles	YouM	YM-99	16100	112700	9.456886	10657.910157	102042.089843
3	01-04-2021	Thursday	M04	C	6	Mobiles	YouM	YM-99 Plus	20100	120600	6.935385	8364.074702	112235.925298
4	01-04-2021	Thursday	M05	C	3	Mobiles	YouM	YM-98	8100	24300	17.995663	4372.946230	19927.053770
...	...	...	...	...	...	...	...	...	...	...	...	...	...
445	15-04-2021	Thursday	L06	C	2	Lifestyle	Jeera	M-Casuals	1300	2600	15.475687	402.367873	2197.632127
446	15-04-2021	Thursday	L07	C	6	Lifestyle	Viva	W-Western	2600	15600	17.057027	2660.896242	12939.103758
447	15-04-2021	Thursday	L08	C	2	Lifestyle	Viva	W-Lounge	1600	3200	18.965550	606.897606	2593.102394
448	15-04-2021	Thursday	L09	C	3	Lifestyle	Jeera	M-Formals	1900	5700	16.793014	957.201826	4742.798174
449	15-04-2021	Thursday	L10	C	1	Lifestyle	Jeera	M-Shoes	3100	3100	15.333300	475.332295	2624.667705

450 rows × 13 columns

2. Identify Numerical and Categorical Columns

```
In [2]: numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
        categorical_cols = df.select_dtypes(include=['object']).columns

        print("Numerical Columns:", numerical_cols)
        print("Categorical Columns:", categorical_cols)
```

```
Numerical Columns: Index(['Volume', 'Avg Price', 'Total Sales Value', 'Discount Rate (%)',
       'Discount Amount', 'Net Sales Value'],
      dtype='object')
Categorical Columns: Index(['Date', 'Day', 'SKU', 'City', 'BU', 'Brand', 'Model'], dtype='object')
```

3. Basic Statistics (Calculate Mean, Median, Mode and Std Dev)

```
In [5]: df[numerical_cols].describe()
```

Out[5]:

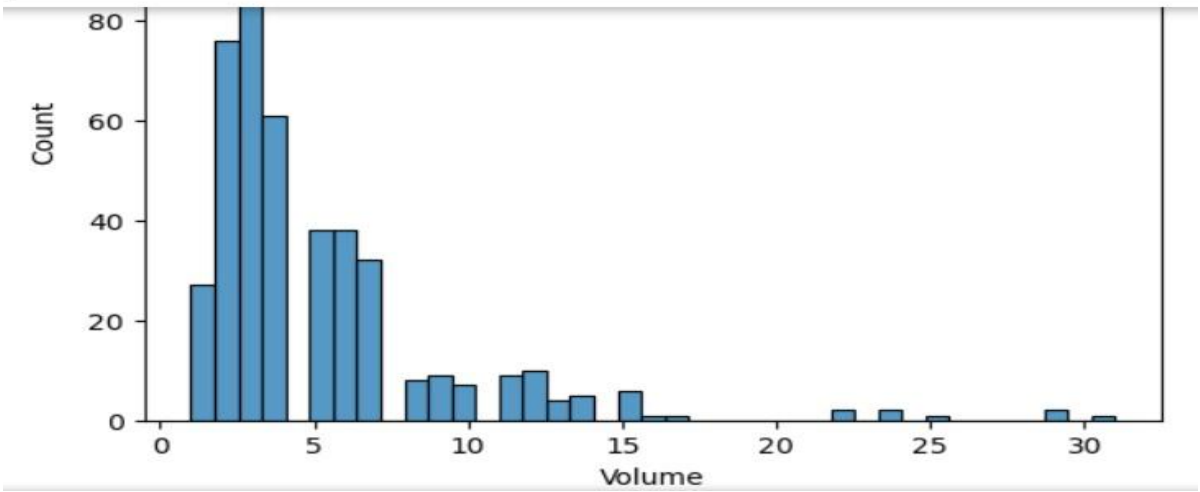
	Volume	Avg Price	Total Sales Value	Discount Rate (%)	Discount Amount	Net Sales Value
count	450.000000	450.000000	450.000000	450.000000	450.000000	450.000000
mean	5.066667	10453.433333	33812.835556	15.155242	3346.499424	30466.336131
std	4.231602	18079.904840	50535.074173	4.220602	4509.902963	46358.656624
min	1.000000	290.000000	400.000000	5.007822	69.177942	326.974801
25%	3.000000	465.000000	2700.000000	13.965063	460.459304	2202.208645
50%	4.000000	1450.000000	5700.000000	16.577766	988.933733	4677.788059
75%	6.000000	10100.000000	53200.000000	18.114718	5316.495427	47847.912852
max	31.000000	60100.000000	196400.000000	19.992407	25738.022194	179507.479049

4.Data Visualization

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

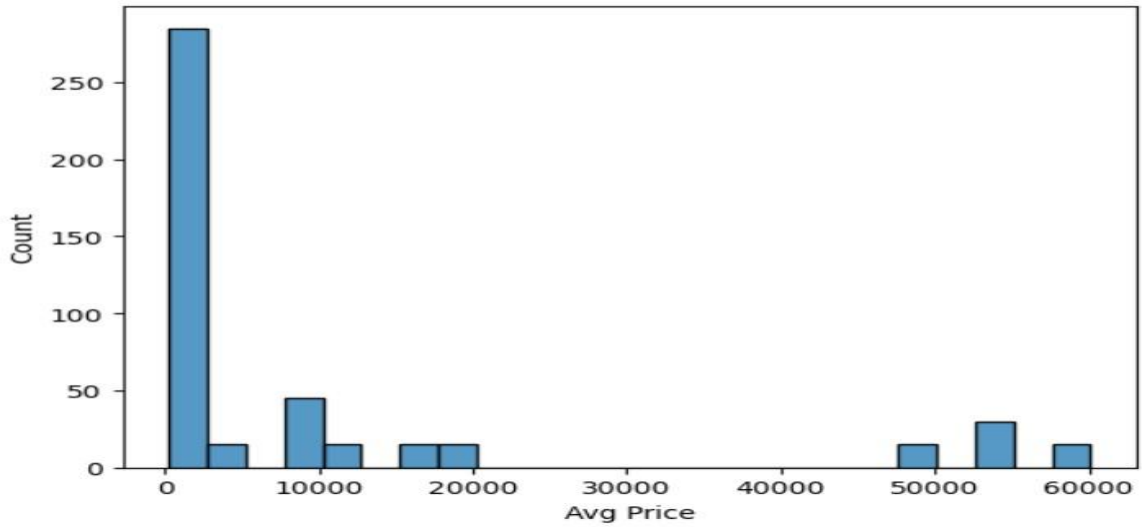
A.Histograms

```
sns.histplot(data=df,x='Volume')
```



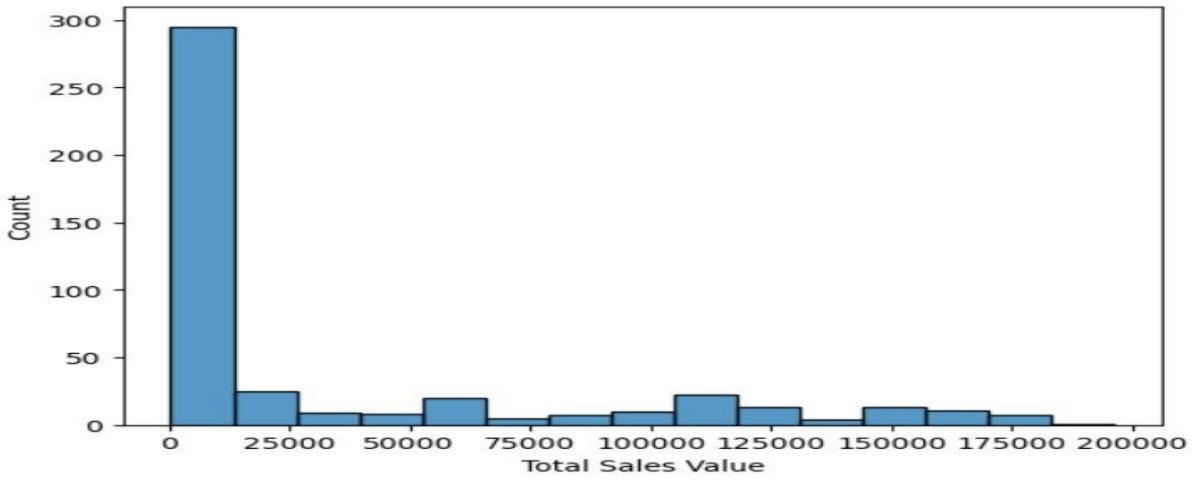

```
sns.histplot(data=df,x='Avg Price')
```

```
<Axes: xlabel='Avg Price', ylabel='Count'>
```



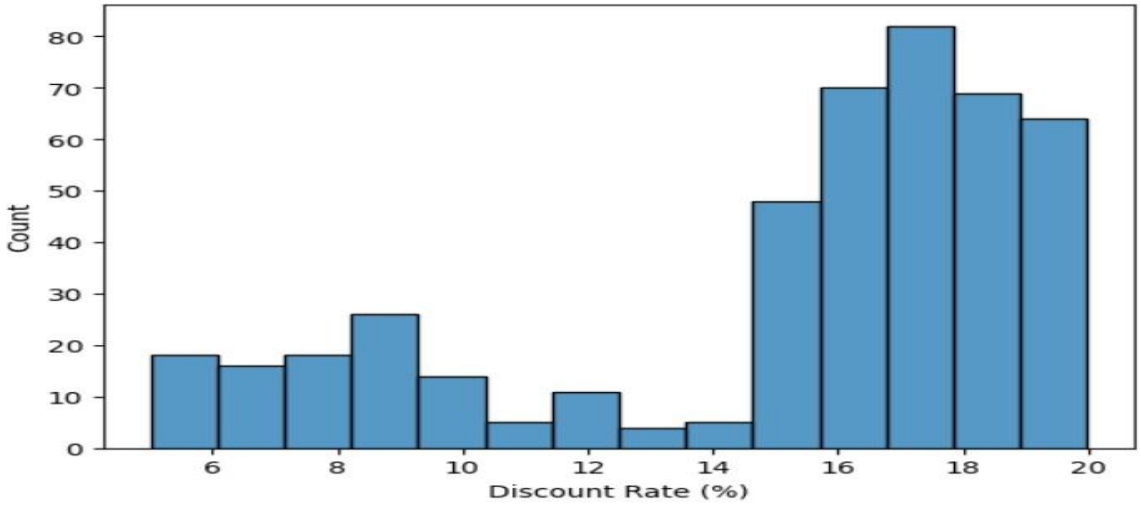
```
sns.histplot(data=df,x='Total Sales Value')
```

```
<Axes: xlabel='Total Sales Value', ylabel='Count'>
```



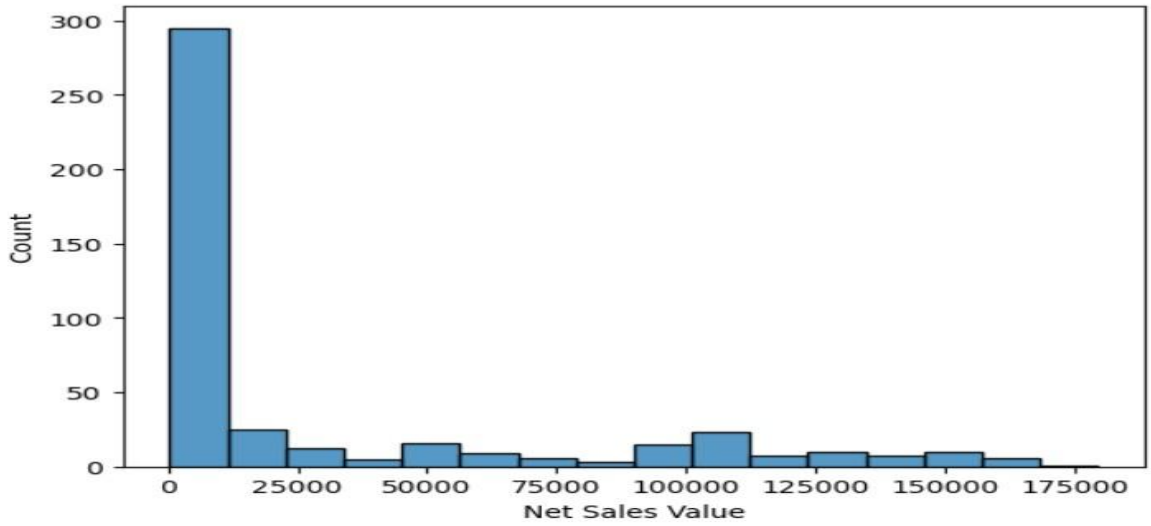

```
sns.histplot(data=df,x='Discount Rate (%)')
```

```
<Axes: xlabel='Discount Rate (%)', ylabel='Count'>
```



```
sns.histplot(data=df,x='Net Sales Value')
```

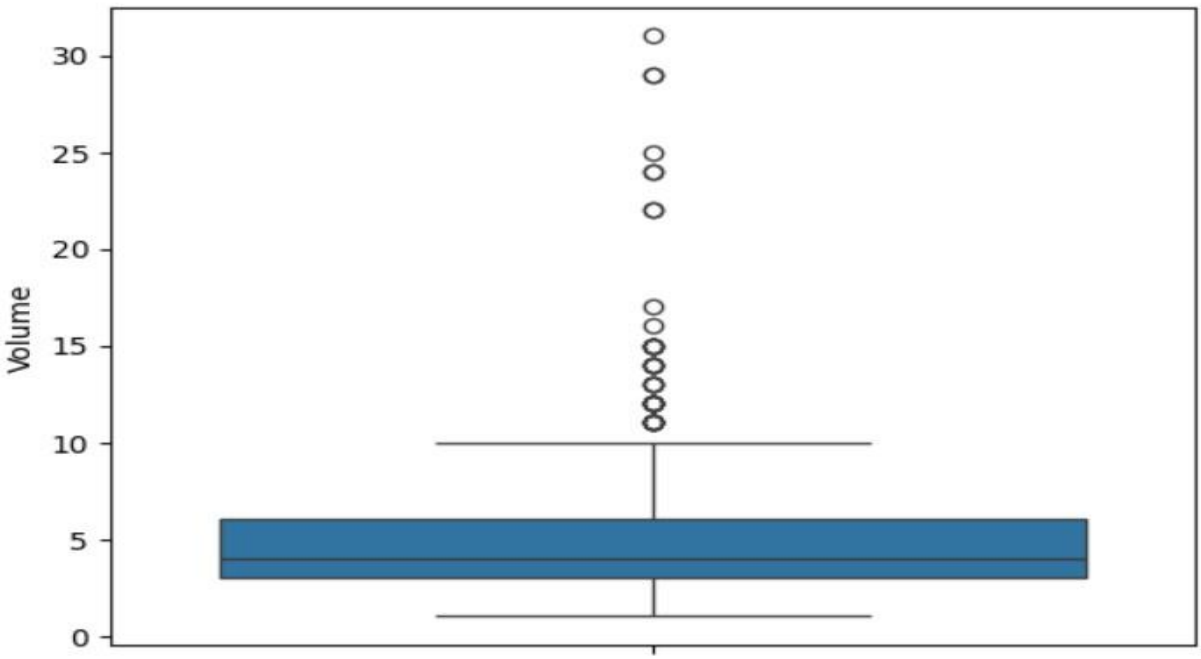
```
<Axes: xlabel='Net Sales Value', ylabel='Count'>
```



B.Boxplots

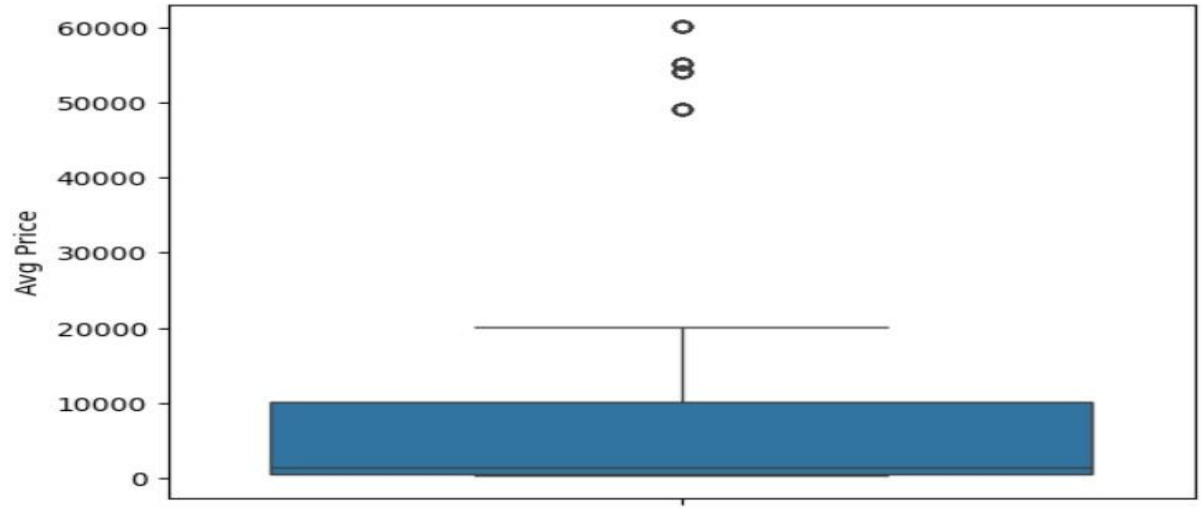
```
sns.boxplot(data=df,y="Volume")
```

```
<Axes: ylabel='Volume'>
```



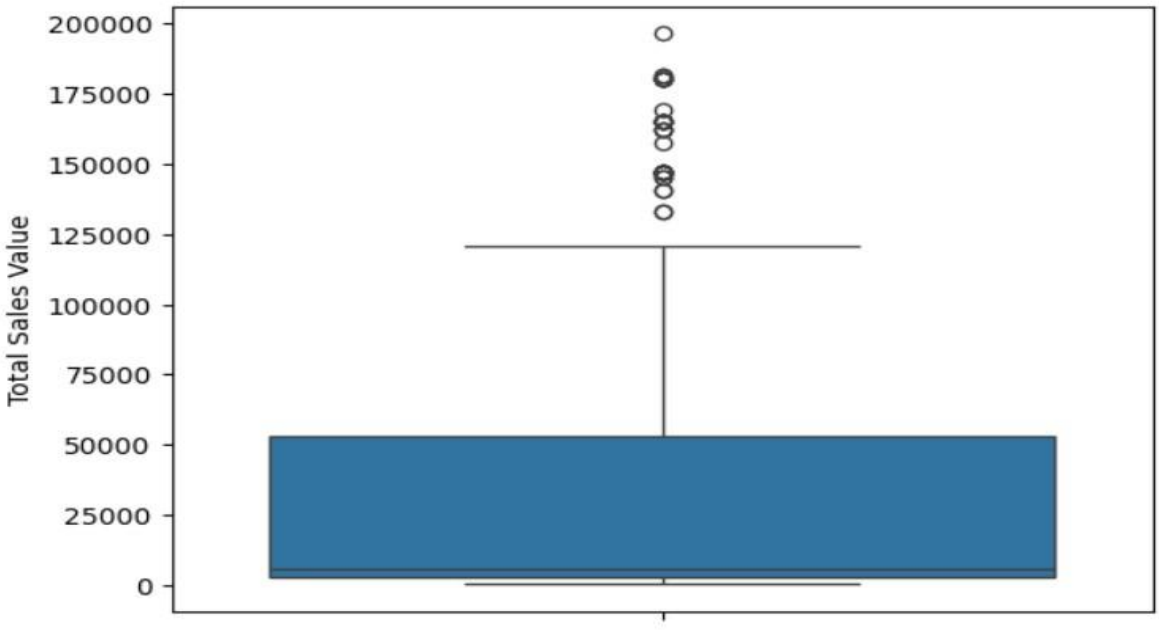
```
sns.boxplot(data=df,y="Avg Price")
```

```
<Axes: ylabel='Avg Price'>
```



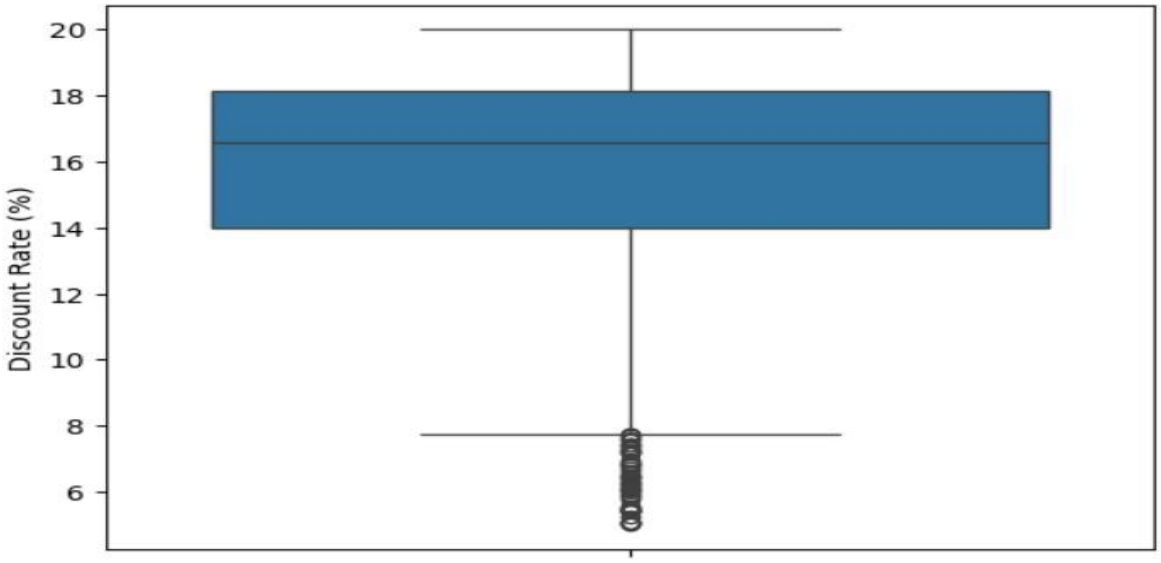

```
sns.boxplot(data=df,y="Total Sales Value")
```

```
<Axes: ylabel='Total Sales Value'>
```



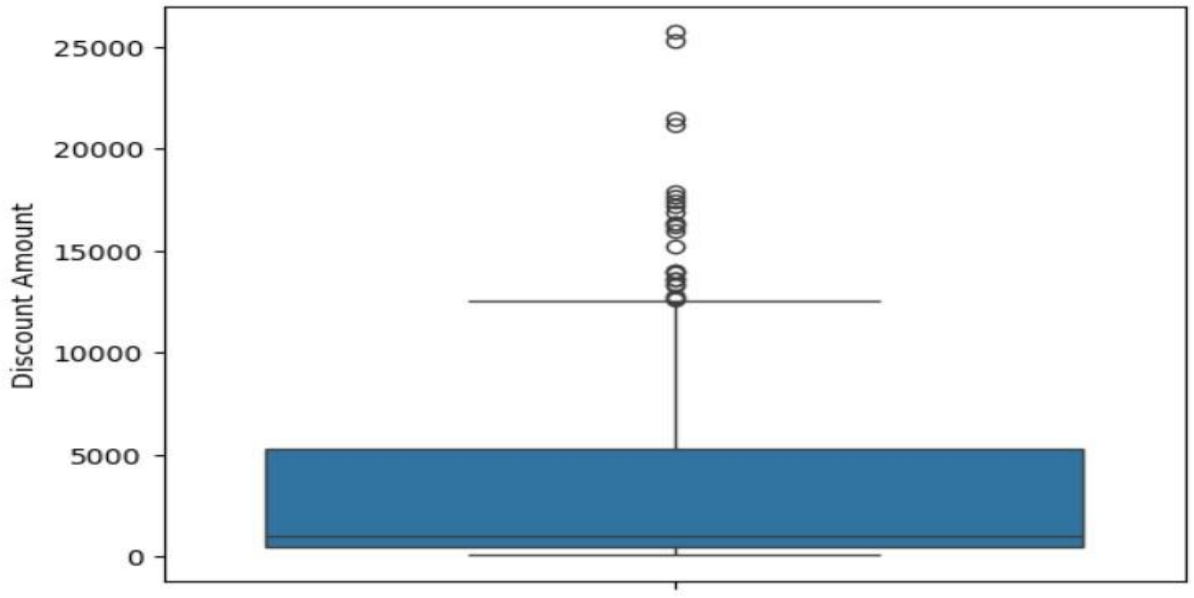
```
sns.boxplot(data=df,y="Discount Rate (%)")
```

```
<Axes: ylabel='Discount Rate (%)'>
```



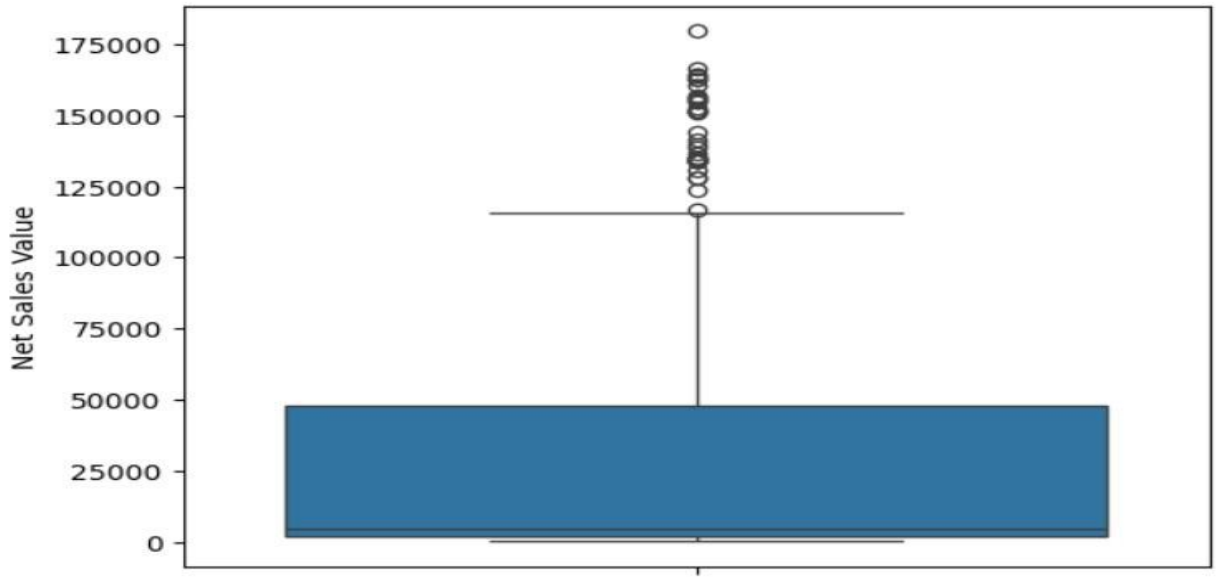

```
sns.boxplot(data=df,y="Discount Amount")
```

```
<Axes: ylabel='Discount Amount'>
```



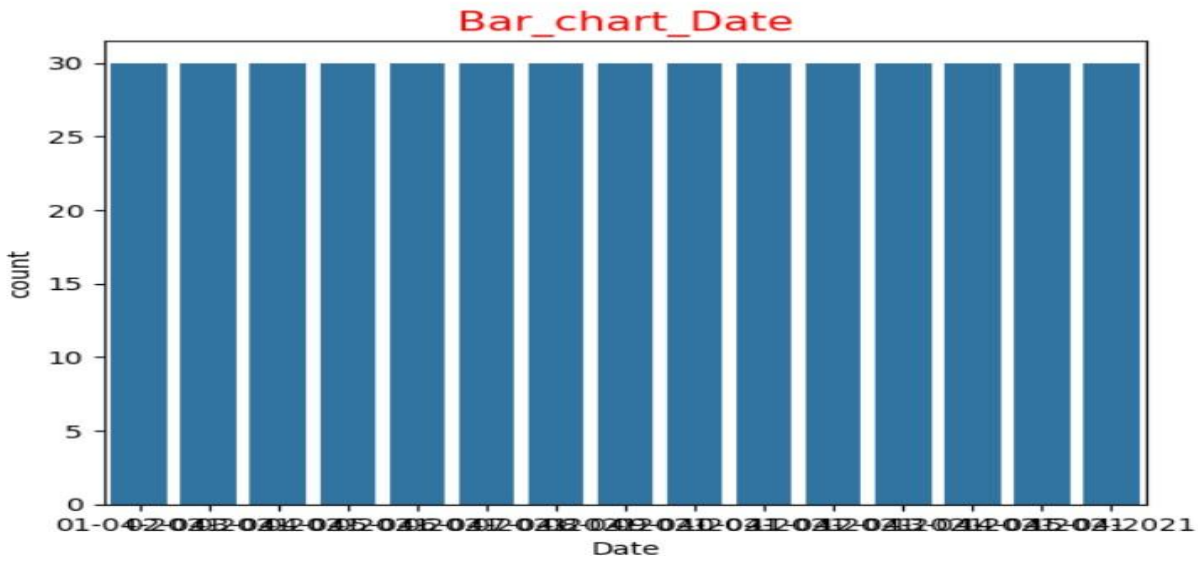
```
sns.boxplot(data=df,y="Net Sales Value")
```

```
<Axes: ylabel='Net Sales Value'>
```

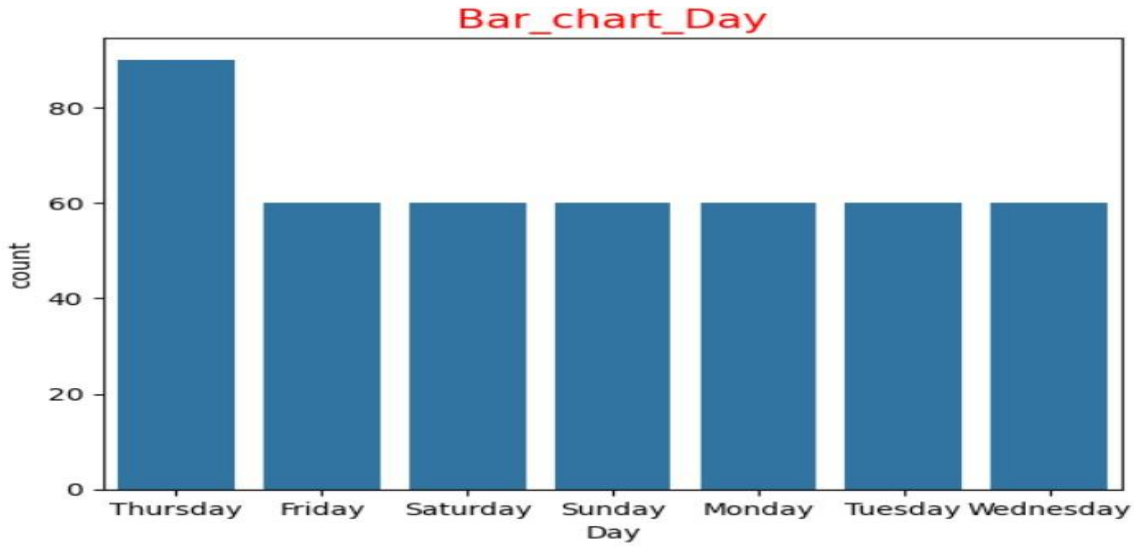


C.Bar Charts (Categorical columns)

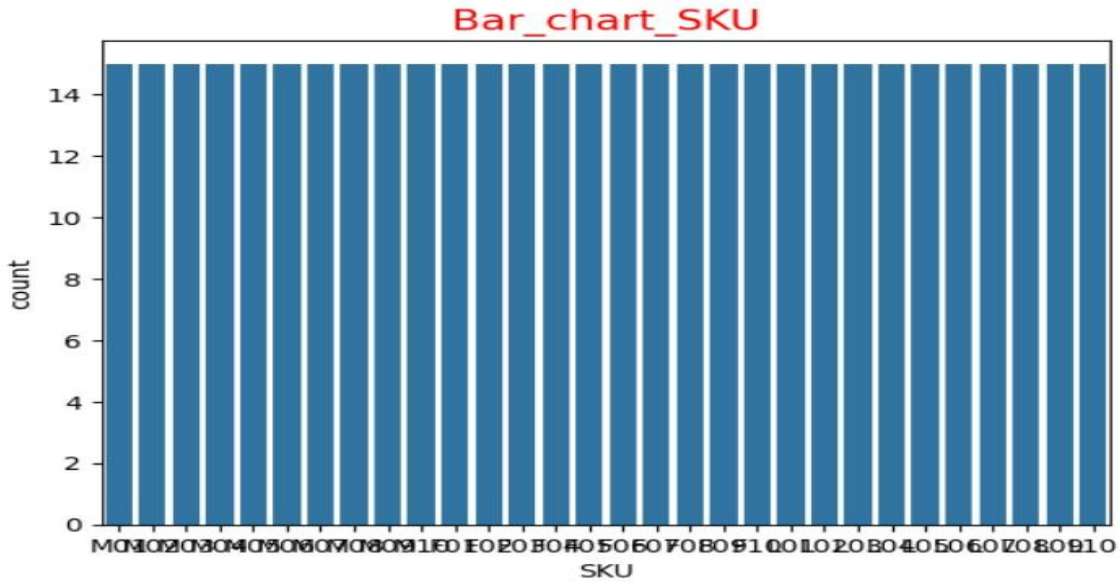
```
sns.countplot(data=df,x="Date");
plt.title("Bar_chart_Date",color="red",size=15);
```



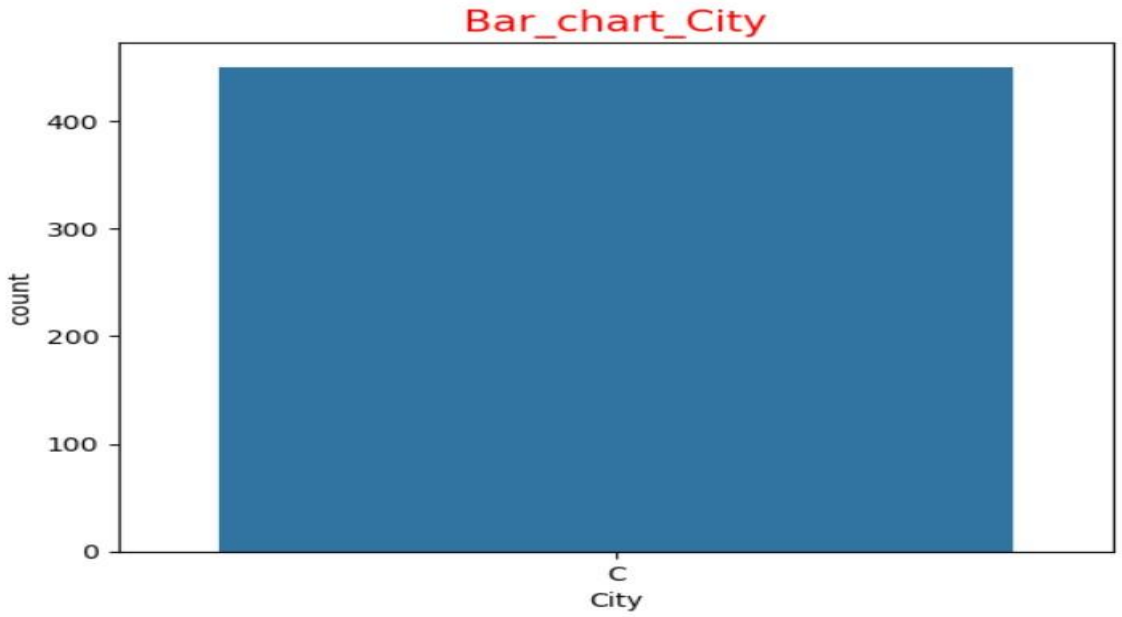
```
sns.countplot(data=df,x="Day");
plt.title("Bar_chart_Day",color="red",size=15);
```



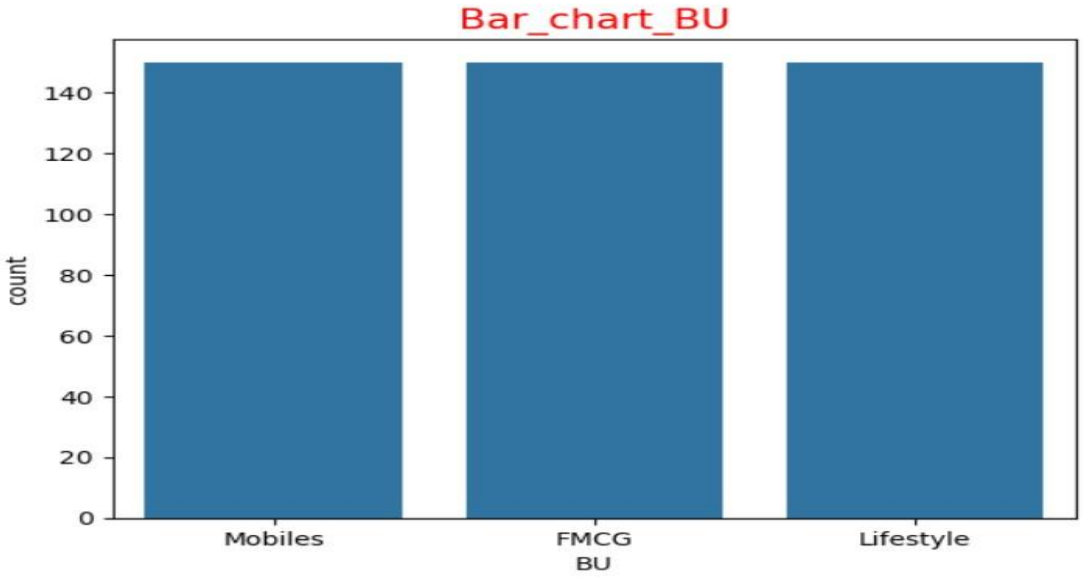

```
sns.countplot(data=df,x="SKU");
plt.title("Bar_chart_SKU",color="red",size=15);
```



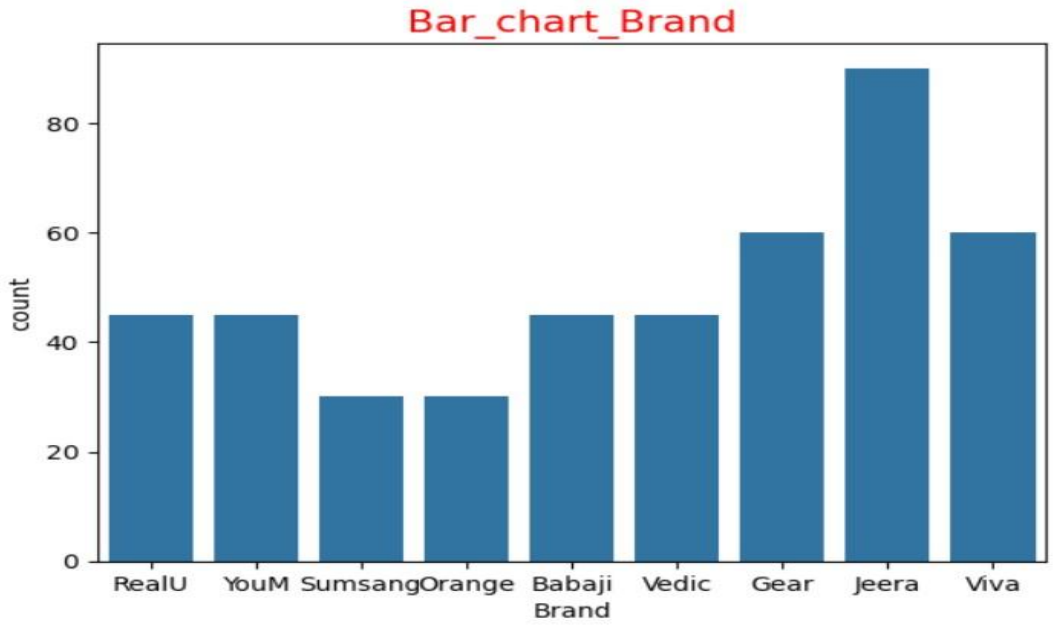
```
sns.countplot(data=df,x="City");
plt.title("Bar_chart_City",color="red",size=15);
```



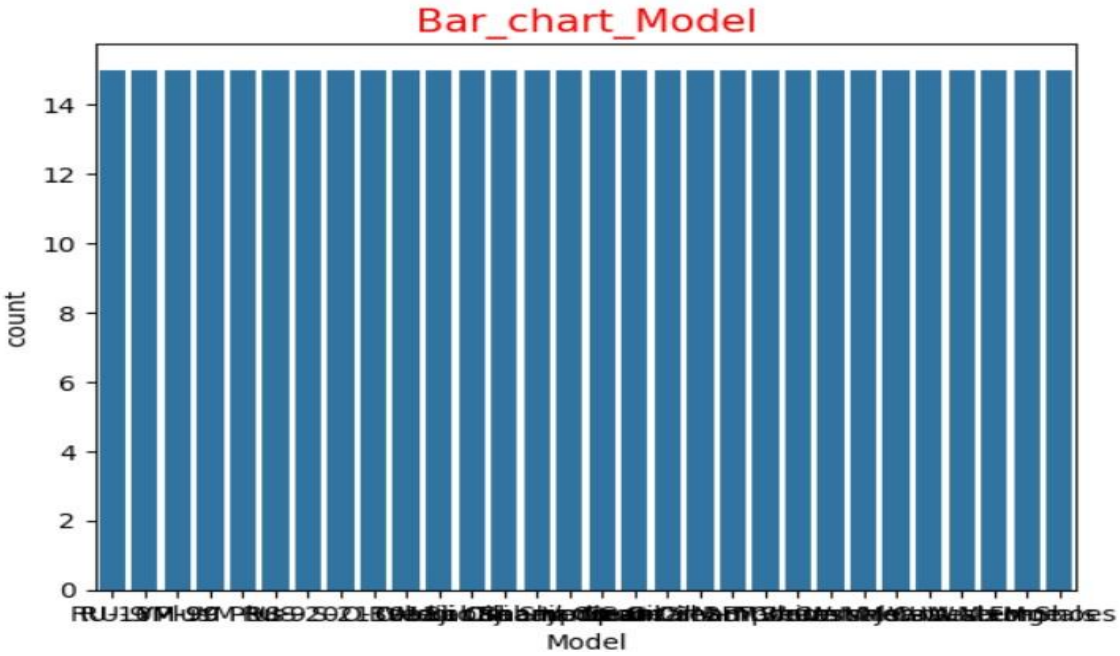

```
sns.countplot(data=df,x="BU");
plt.title("Bar_chart_BU",color="red",size=15);
```



```
sns.countplot(data=df,x="Brand");
plt.title("Bar_chart_Brand",color="red",size=15);
```




```
sns.countplot(data=df,x="Model");
plt.title("Bar_chart_Model",color="red",size=15);
```



5.Standardization of Numerical Variable

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

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
standardized_data = scaler.fit_transform(df[numerical_cols])
standardized_df = pd.DataFrame(standardized_data, columns=numerical_cols)
standardized_df.head()
```

	Volume	Avg Price	Total Sales Value	Discount Rate (%)	Discount Amount	Net Sales Value
0	2.350029	0.091173	2.925721	-0.830289	3.952816	2.804756
1	1.167129	-0.019570	1.330995	-0.852661	1.849014	1.271026
2	0.457388	0.312659	1.562775	-1.351631	1.622995	1.545675
3	0.220808	0.534146	1.719276	-1.949723	1.113807	1.765810
4	-0.488932	-0.130313	-0.188452	0.673739	0.227852	-0.227595

6.Conversion of Categorical Data into Dummy Variables

```
df_encoded = pd.get_dummies(df, columns=categorical_cols)
df_encoded.head()
```

	Volume	Avg Price	Total Sales Value	Discount Rate (%)	Discount Amount	Net Sales Value	Date_01-04-2021	Date_02-04-2021	Date_03-04-2021	Date_04-04-2021	...	Model_Vedic Cream	Model_Vedic Oil	Model_Vedic Shampoo	Model_W-Casuals
0	15	12100	181500	11.654820	21153.498820	160346.501180	True	False	False	False	...	False	False	False	False
1	10	10100	101000	11.560498	11676.102961	89323.897039	True	False	False	False	...	False	False	False	False
2	7	16100	112700	9.456886	10657.910157	102042.089843	True	False	False	False	...	False	False	False	False
3	6	20100	120600	6.935385	8364.074702	112235.925298	True	False	False	False	...	False	False	False	False
4	3	8100	24300	17.995663	4372.946230	19927.053770	True	False	False	False	...	False	False	False	False

5 rows × 101 columns

Conclusion:

- **Descriptive Analytics** revealed key trends in sales such as average spend, product category popularity, and customer demographics.
- **Visualizations** helped detect outliers and understand data spread.
- **Standardization** ensured uniform scale for numerical variables.
- **One-hot encoding** transformed categorical data into a usable format for ML models.