```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_rel, ttest_1samp, t
```

```
Start coding or generate with AI.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# ----------------------------------------
# TASK 1 - Sales Data Analysis and Preprocessing
# ----------------------------------------

# Load Dataset
sales_df = pd.read_excel("/sales_data_with_discounts.xlsx")
sales_df.head()
```

	Date	Day	SKU	City	Volume	BU	Brand	Model	Avg Price	Total Sales Value	Discount Rate (%)
0	2021-04-01	Thursday	M01	C	15	Mobiles	RealU	RU-10	12100	181500	11.654820
1	2021-04-01	Thursday	M02	C	10	Mobiles	RealU	RU-9 Plus	10100	101000	11.560498
2	2021-04-01	Thursday	M03	C	7	Mobiles	YouM	YM-99	16100	112700	9.456886
3	2021-04-01	Thursday	M04	C	6	Mobiles	YouM	YM-99 Plus	20100	120600	6.935385
4	2021-04-01	Thursday	M05	C	3	Mobiles	YouM	YM-98	8100	24300	17.995663

Next steps: Generate code with sales_df | View recommended plots | New interactive sheet

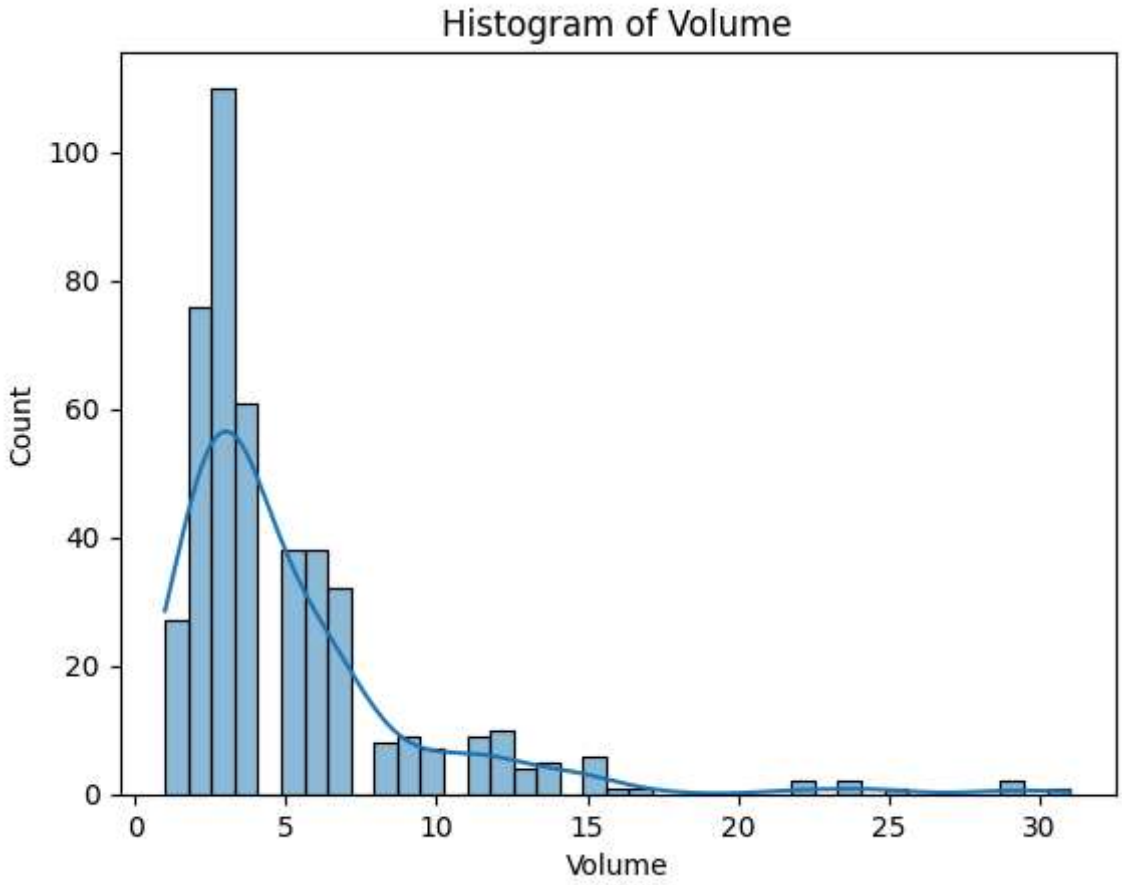
```
# 1. Descriptive Analytics
numerical_cols = sales_df.select_dtypes(include=[np.number]).columns
desc_stats = sales_df[numerical_cols].describe().T

# Mean, Median, Mode, Std
desc_stats['median'] = sales_df[numerical_cols].median()
```

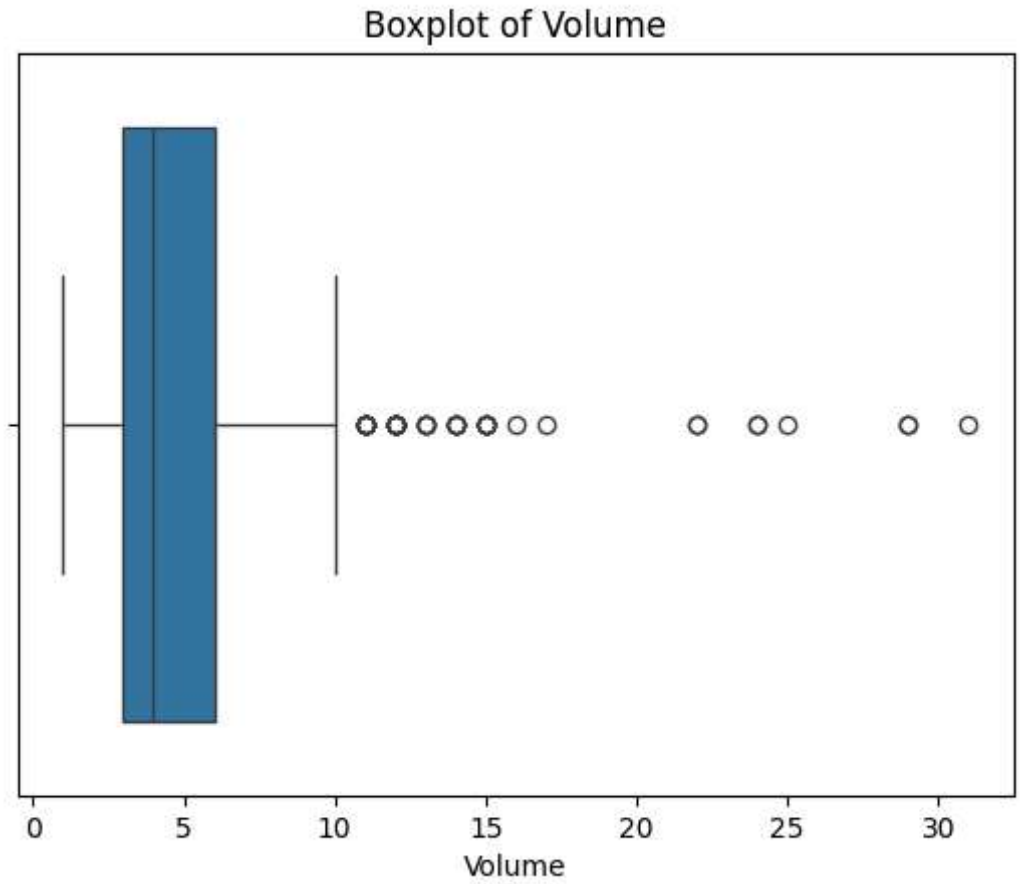
```
desc_stats['mode'] = sales_df[numerical_cols].mode().iloc[0]
desc_stats
```

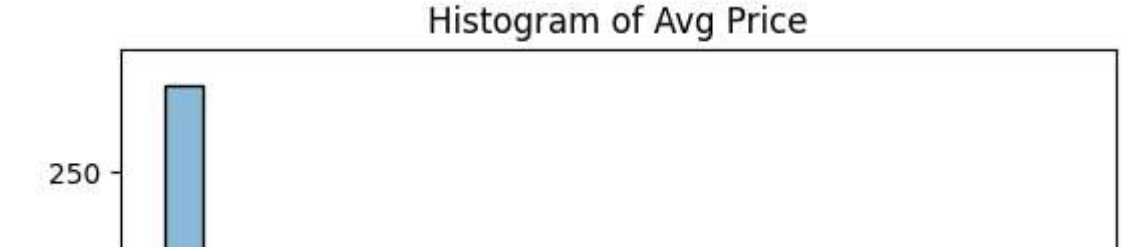
	count	mean	std	min	25%	50%	
Volume	450.0	5.066667	4.231602	1.000000	3.000000	4.000000	
Avg Price	450.0	10453.433333	18079.904840	290.000000	465.000000	1450.000000	1010
Total Sales Value	450.0	33812.835556	50535.074173	400.000000	2700.000000	5700.000000	5320
Discount Rate (%)	450.0	15.155242	4.220602	5.007822	13.965063	16.577766	1
Discount Amount	450.0	3346.499424	4509.902963	69.177942	460.459304	988.933733	531
Net Sales Value	450.0	30466.336131	46358.656624	326.974801	2202.208645	4677.788059	4784

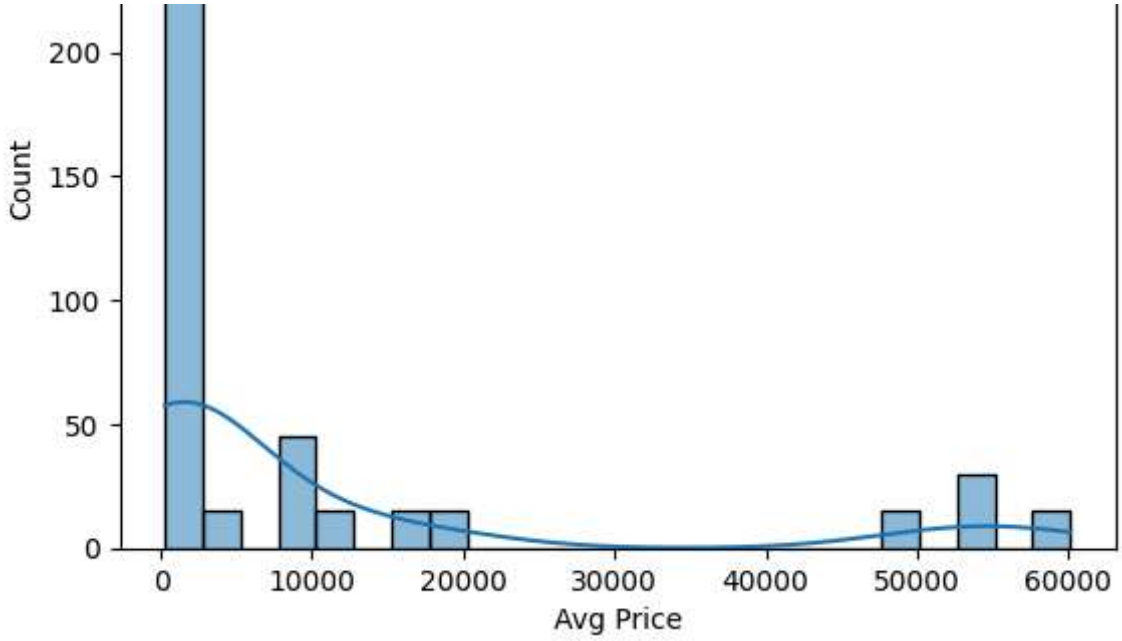
Next steps: Generate code with `desc_stats` View recommended plots New interactive sheet

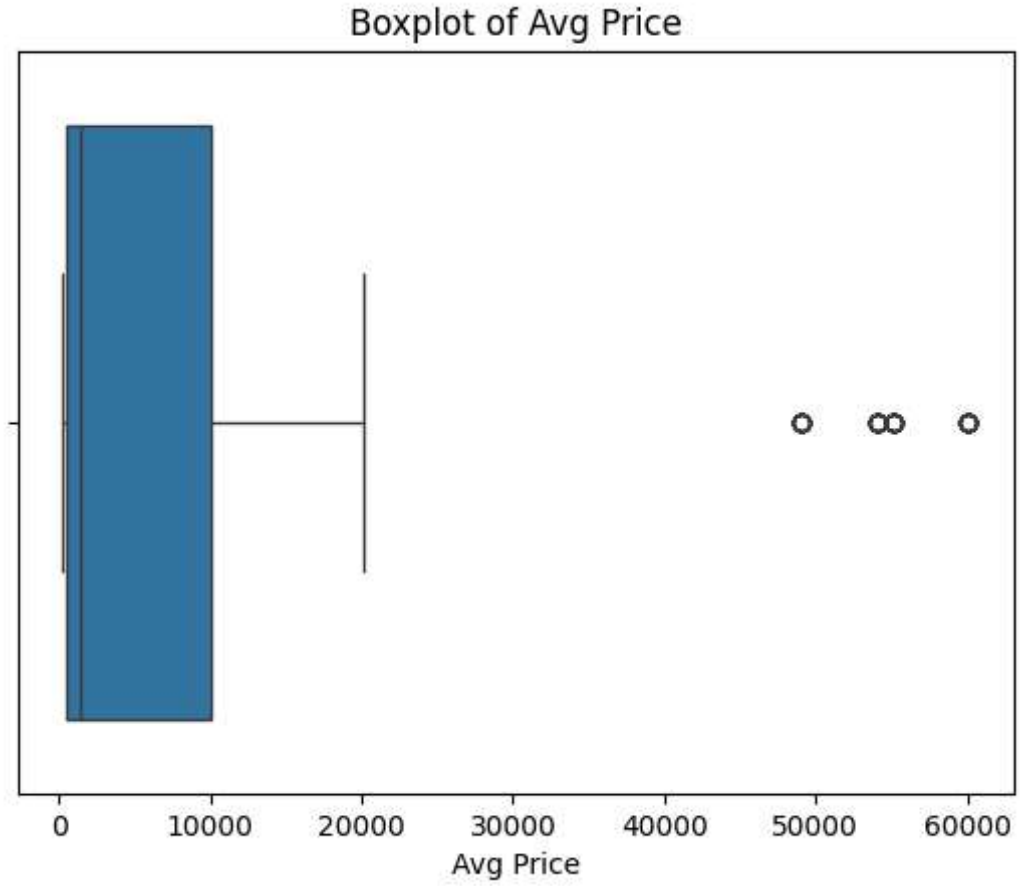
```
# 2. Data Visualization
for col in numerical_cols:
    plt.figure()
    sns.histplot(sales_df[col].dropna(), kde=True)
    plt.title(f"Histogram of {col}")
    plt.show()

    plt.figure()
    sns.boxplot(x=sales_df[col])
    plt.title(f"Boxplot of {col}")
    plt.show()

# Categorical Analysis
categorical_cols = sales_df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    plt.figure()
    sales_df[col].value_counts().plot(kind='bar')
    plt.title(f"Category Count: {col}")
    plt.ylabel("Frequency")
    plt.show()
```

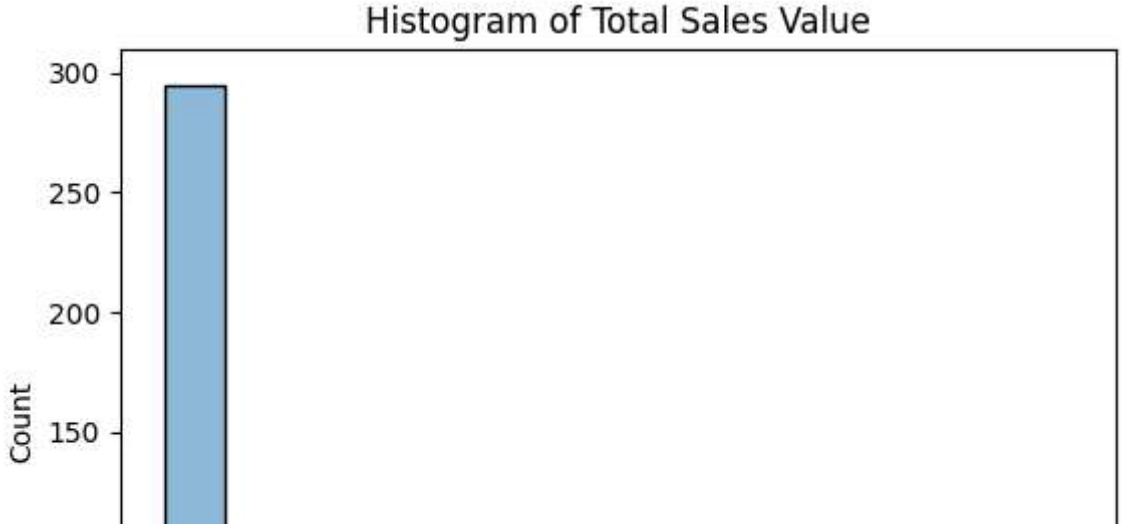

Histogram of Volume
Count
Volume
0
20
40
60
80
100
0
5
10
15
20
25
30

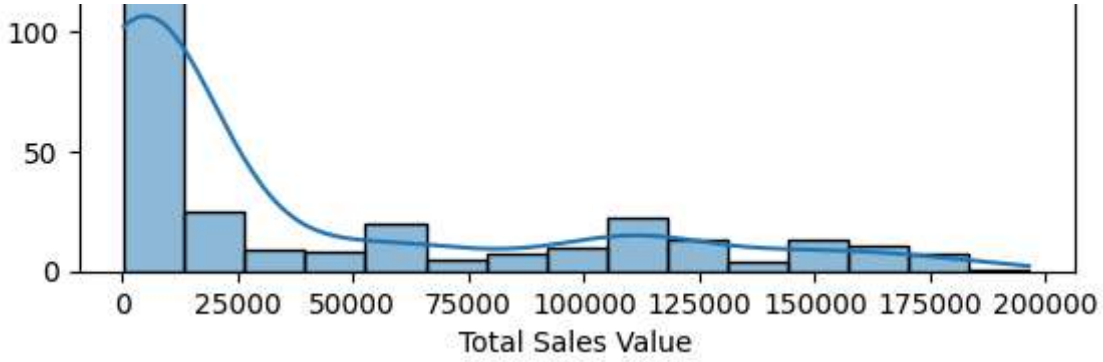

Boxplot of Volume
Volume
0
5
10
15
20
25
30

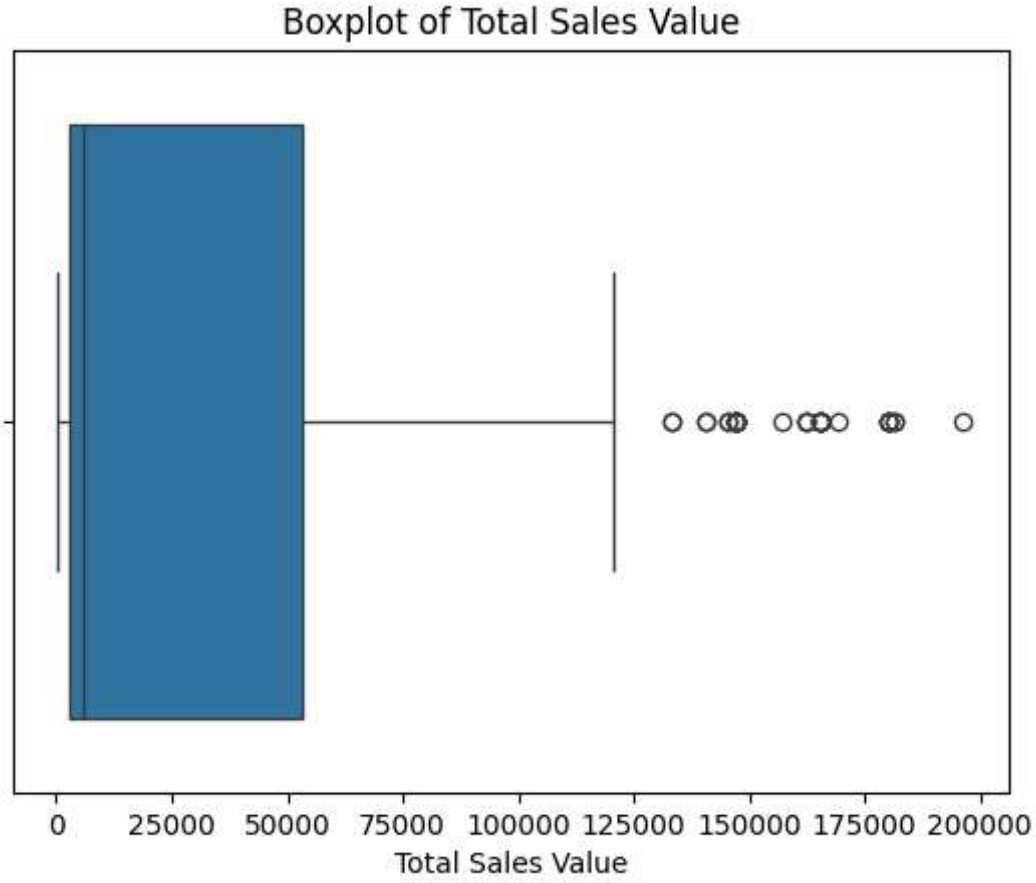

Histogram of Avg Price
250

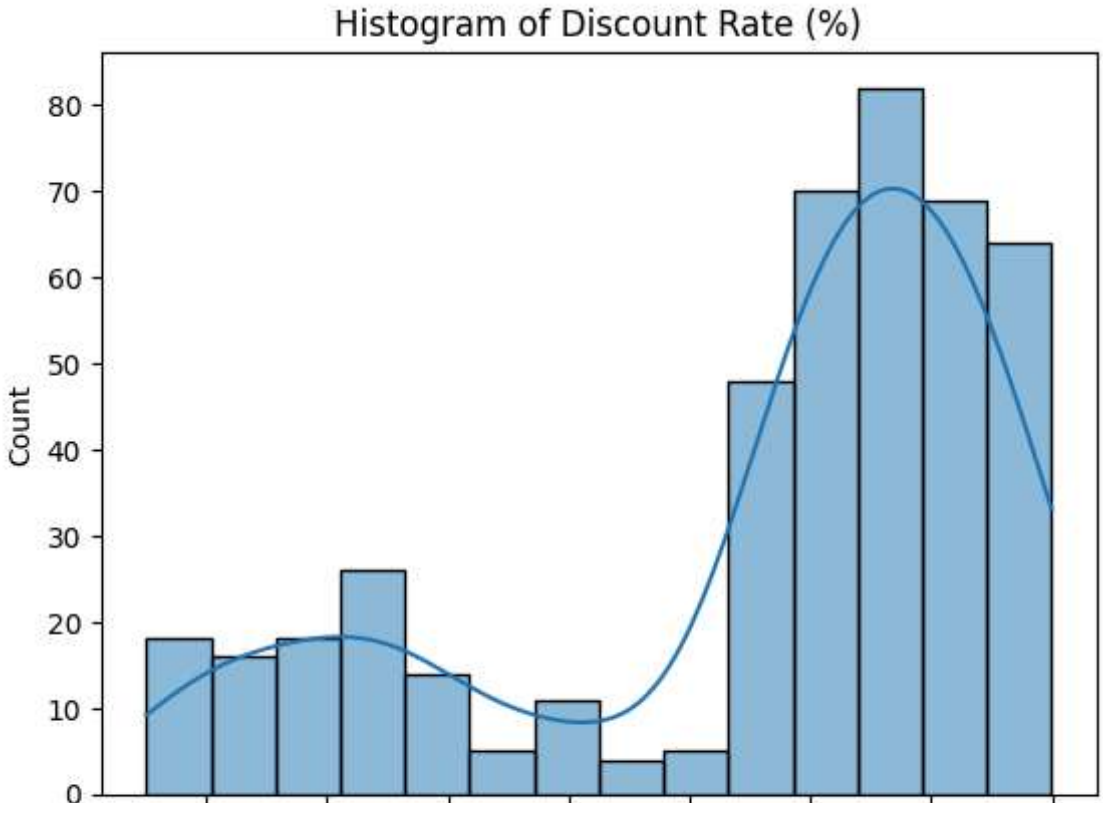

Count
200
150
100
50
0
0
10000
20000
30000
40000
50000
60000
Avg Price

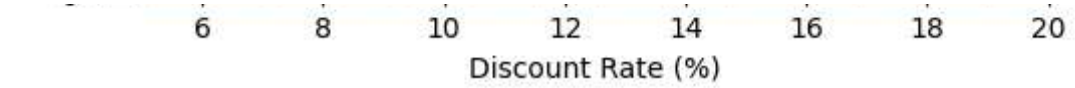

Boxplot of Avg Price
0
10000
20000
30000
40000
50000
60000
Avg Price

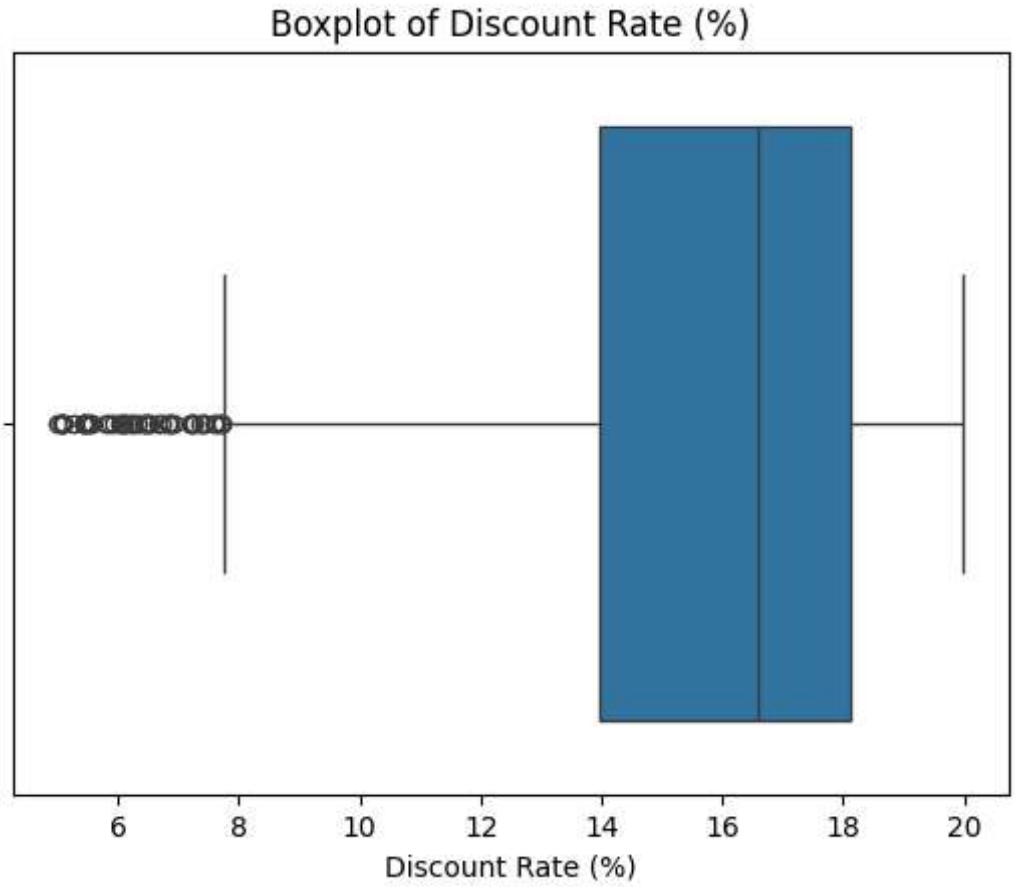

Histogram of Total Sales Value
Count
300
250
200
150

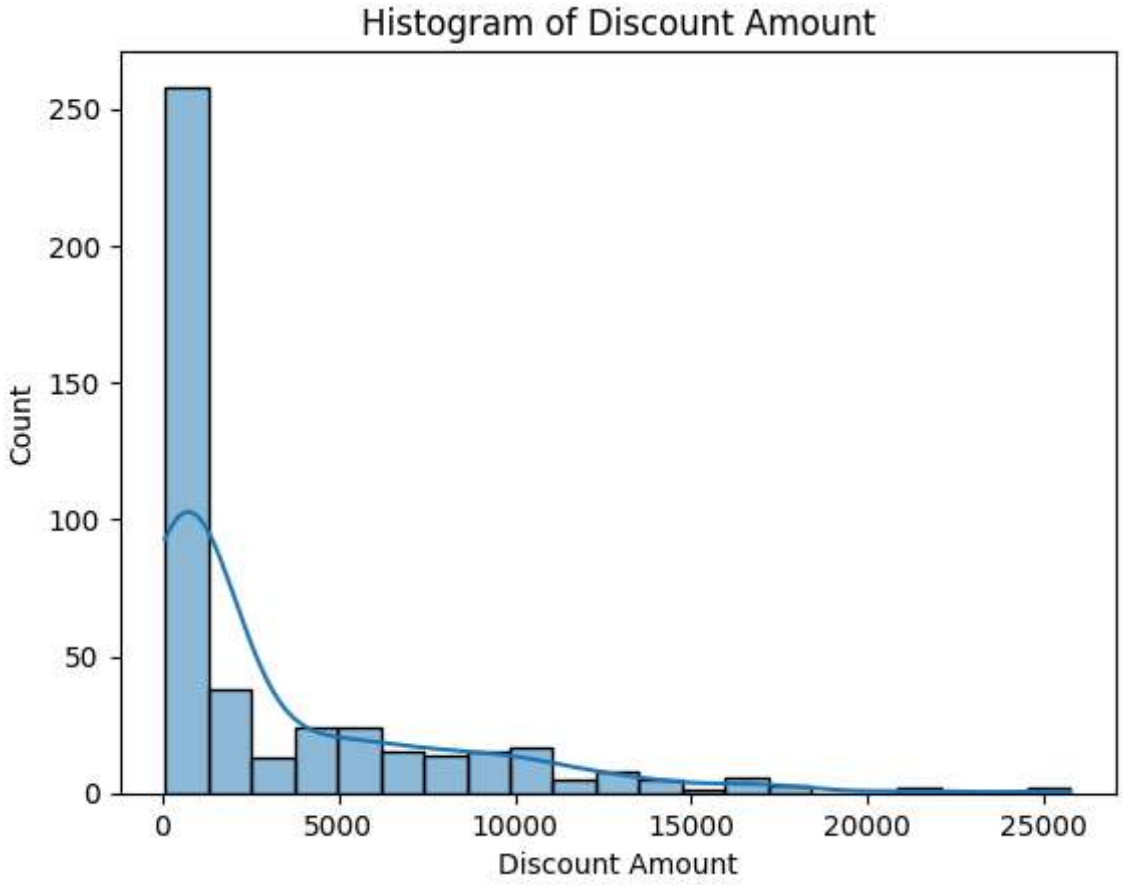

100
50
0
0
25000
50000
75000
100000
125000
150000
175000
200000
Total Sales Value


Boxplot of Total Sales Value
0
25000
50000
75000
100000
125000
150000
175000
200000
Total Sales Value


Histogram of Discount Rate (%)
Count
80
70
60
50
40
30
20
10
0


6
8
10
12
14
16
18
20
Discount Rate (%)


Boxplot of Discount Rate (%)
6
8
10
12
14
16
18
20
Discount Rate (%)


Histogram of Discount Amount
250
200
150
100
50
0
Count
0
5000
10000
15000
20000
25000
Discount Amount


Boxplot of Discount Amount