basic-stats1

November 20, 2024

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv("/content/sales_data_with_discounts.csv")
df
```

[1]:
```
           Date       Day  SKU City  Volume         BU  Brand       Model  \
0    01-04-2021  Thursday  M01    C      15    Mobiles  RealU       RU-10
1    01-04-2021  Thursday  M02    C      10    Mobiles  RealU   RU-9 Plus
2    01-04-2021  Thursday  M03    C       7    Mobiles   YouM       YM-99
3    01-04-2021  Thursday  M04    C       6    Mobiles   YouM  YM-99 Plus
4    01-04-2021  Thursday  M05    C       3    Mobiles   YouM      YM-98
..          …         …   …   …      …       …      …
445  15-04-2021  Thursday  L06    C       2  Lifestyle  Jeera   M-Casuals
446  15-04-2021  Thursday  L07    C       6  Lifestyle   Viva   W-Western
447  15-04-2021  Thursday  L08    C       2  Lifestyle   Viva    W-Lounge
448  15-04-2021  Thursday  L09    C       3  Lifestyle  Jeera   M-Formals
449  15-04-2021  Thursday  L10    C       1  Lifestyle  Jeera     M-Shoes

     Avg Price  Total Sales Value  Discount Rate (%)  Discount Amount  \
0        12100             181500          11.654820     21153.498820
1        10100             101000          11.560498     11676.102961
2        16100             112700           9.456886     10657.910157
3        20100             120600           6.935385      8364.074702
4         8100              24300          17.995663      4372.946230
..         …                …                …               …
445       1300               2600          15.475687       402.367873
446       2600              15600          17.057027      2660.896242
447       1600               3200          18.965550       606.897606
448       1900               5700          16.793014       957.201826
449       3100               3100          15.333300       475.332295

     Net Sales Value
0      160346.501180
1       89323.897039
2      102042.089843
3      112235.925298
```

```
4       19927.053770
..               …
445      2197.632127
446     12939.103758
447      2593.102394
448      4742.798174
449      2624.667705

[450 rows x 13 columns]
```

[2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 450 entries, 0 to 449
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Date               450 non-null    object
 1   Day                450 non-null    object
 2   SKU                450 non-null    object
 3   City               450 non-null    object
 4   Volume             450 non-null    int64
 5   BU                 450 non-null    object
 6   Brand              450 non-null    object
 7   Model              450 non-null    object
 8   Avg Price          450 non-null    int64
 9   Total Sales Value  450 non-null    int64
 10  Discount Rate (%)  450 non-null    float64
 11  Discount Amount    450 non-null    float64
 12  Net Sales Value    450 non-null    float64
dtypes: float64(3), int64(3), object(7)
memory usage: 45.8+ KB
```

[4]:
```
df.describe()
```

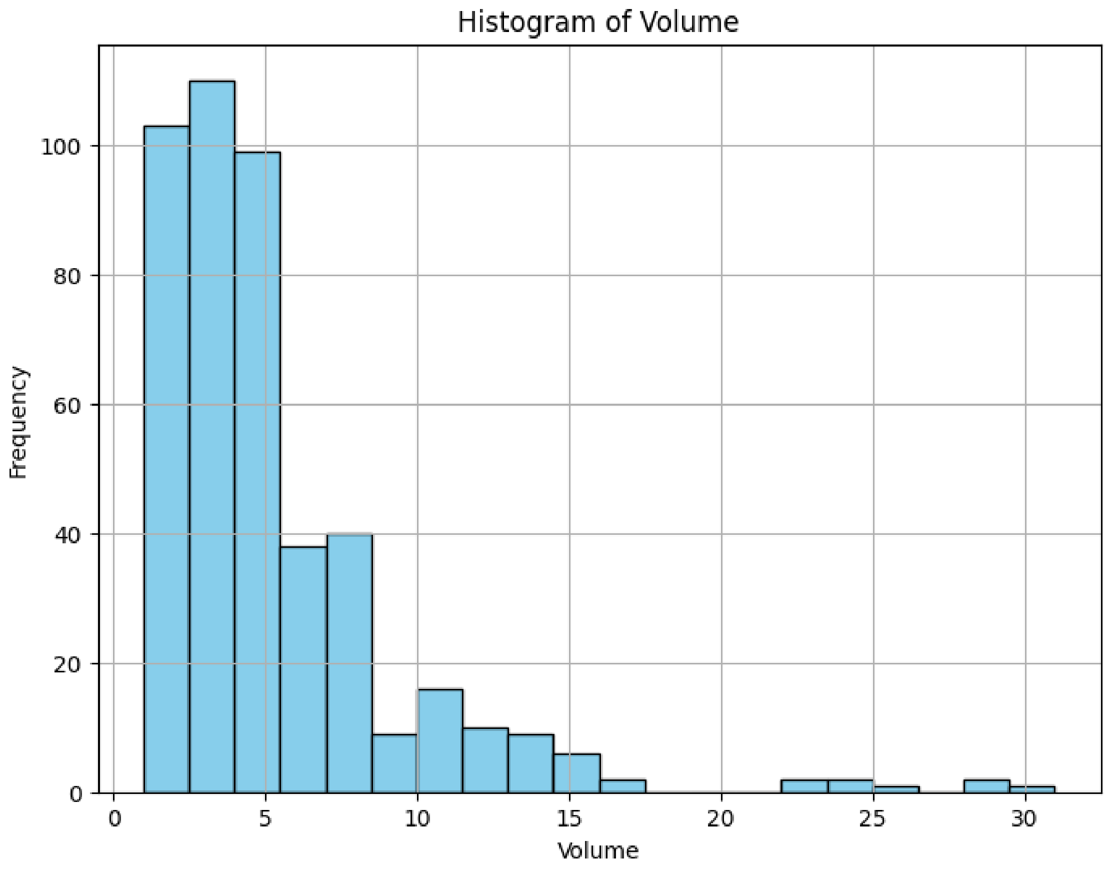
[4]:
```
           Volume     Avg Price  Total Sales Value  Discount Rate (%)  \
count  450.000000    450.000000         450.000000         450.000000
mean     5.066667  10453.433333       33812.835556          15.155242
std      4.231602  18079.904840       50535.074173           4.220602
min      1.000000    290.000000         400.000000           5.007822
25%      3.000000    465.000000        2700.000000          13.965063
50%      4.000000   1450.000000        5700.000000          16.577766
75%      6.000000  10100.000000       53200.000000          18.114718
max     31.000000  60100.000000      196400.000000          19.992407

       Discount Amount  Net Sales Value
count       450.000000       450.000000
```

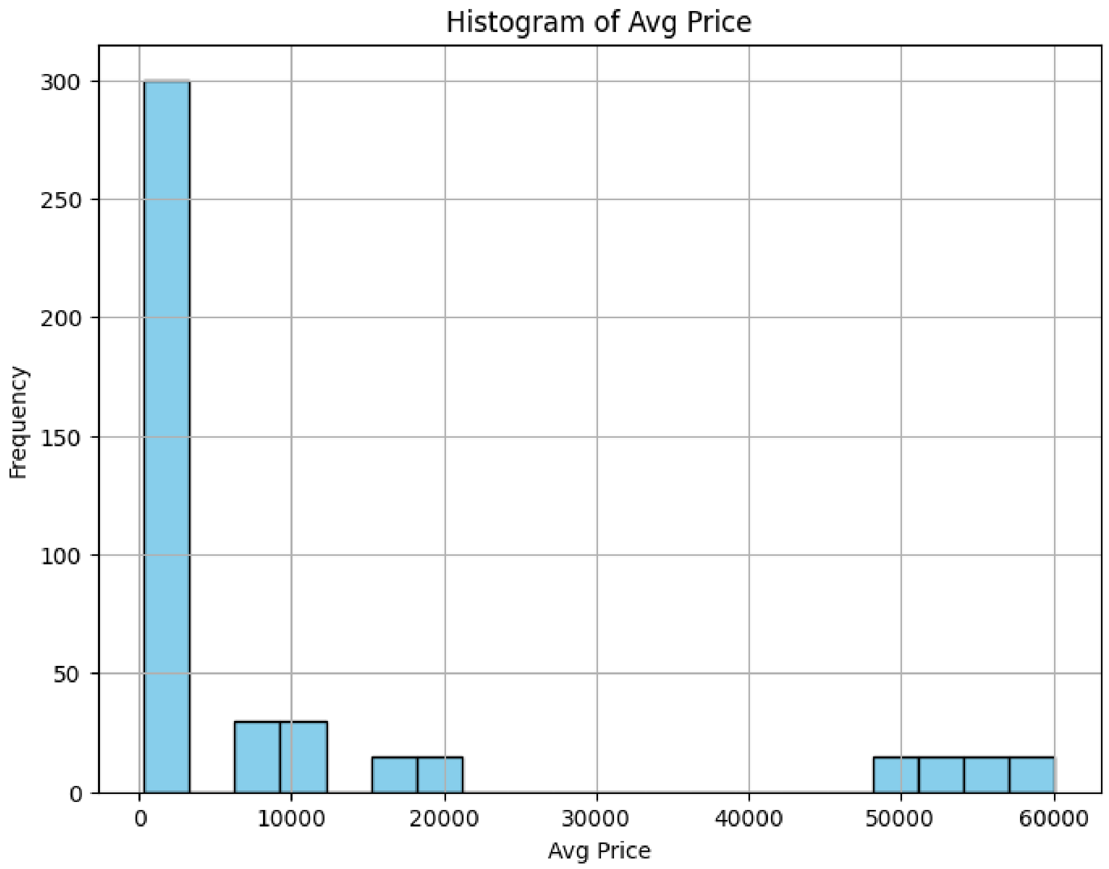
```
mean       3346.499424     30466.336131
std        4509.902963     46358.656624
min          69.177942       326.974801
25%         460.459304      2202.208645
50%         988.933733      4677.788059
75%        5316.495427     47847.912852
max       25738.022194    179507.479049
```

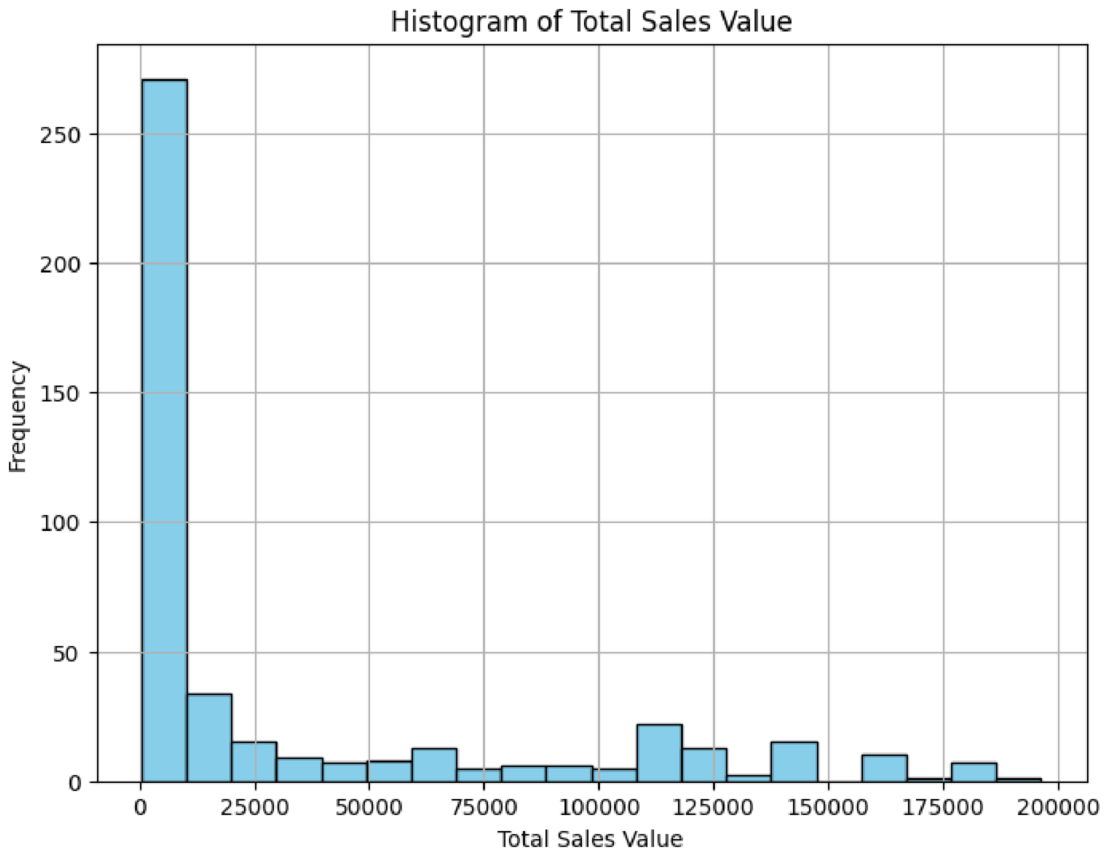
[5]:
```
numerical_columns = df.select_dtypes(include=['int','float']).columns
numerical_columns
```

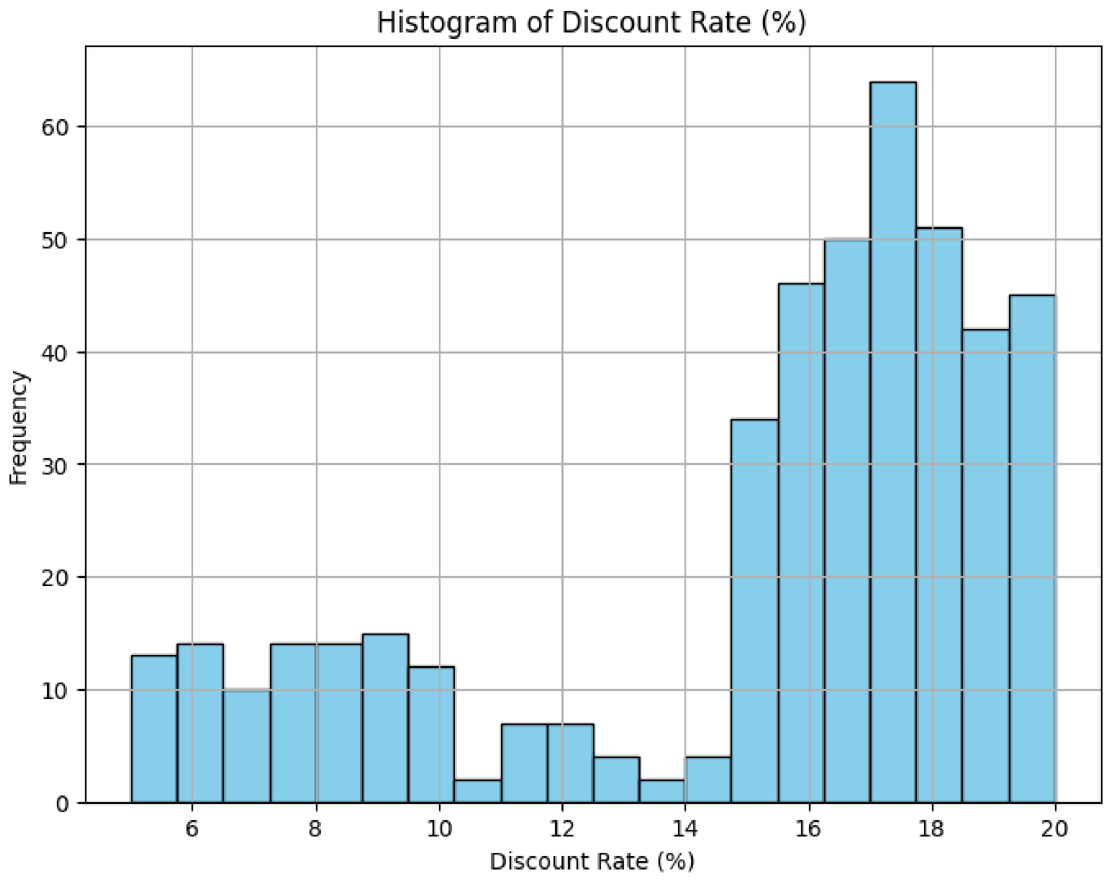
[5]:
```
Index(['Volume', 'Avg Price', 'Total Sales Value', 'Discount Rate (%)',
       'Discount Amount', 'Net Sales Value'],
      dtype='object')
```

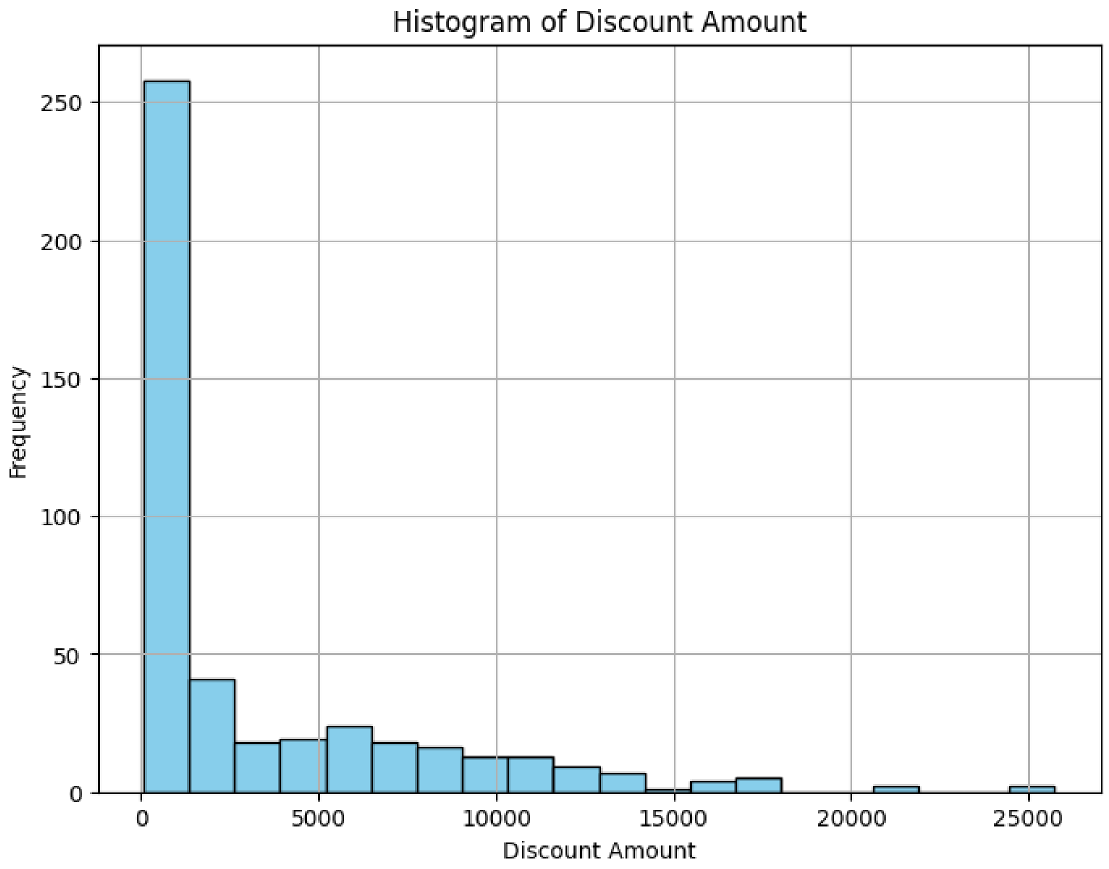
[6]:
```
for column in numerical_columns:
    plt.figure(figsize=(8, 6))
    plt.hist(df[column], bins=20, color='skyblue', edgecolor='black')
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()
```

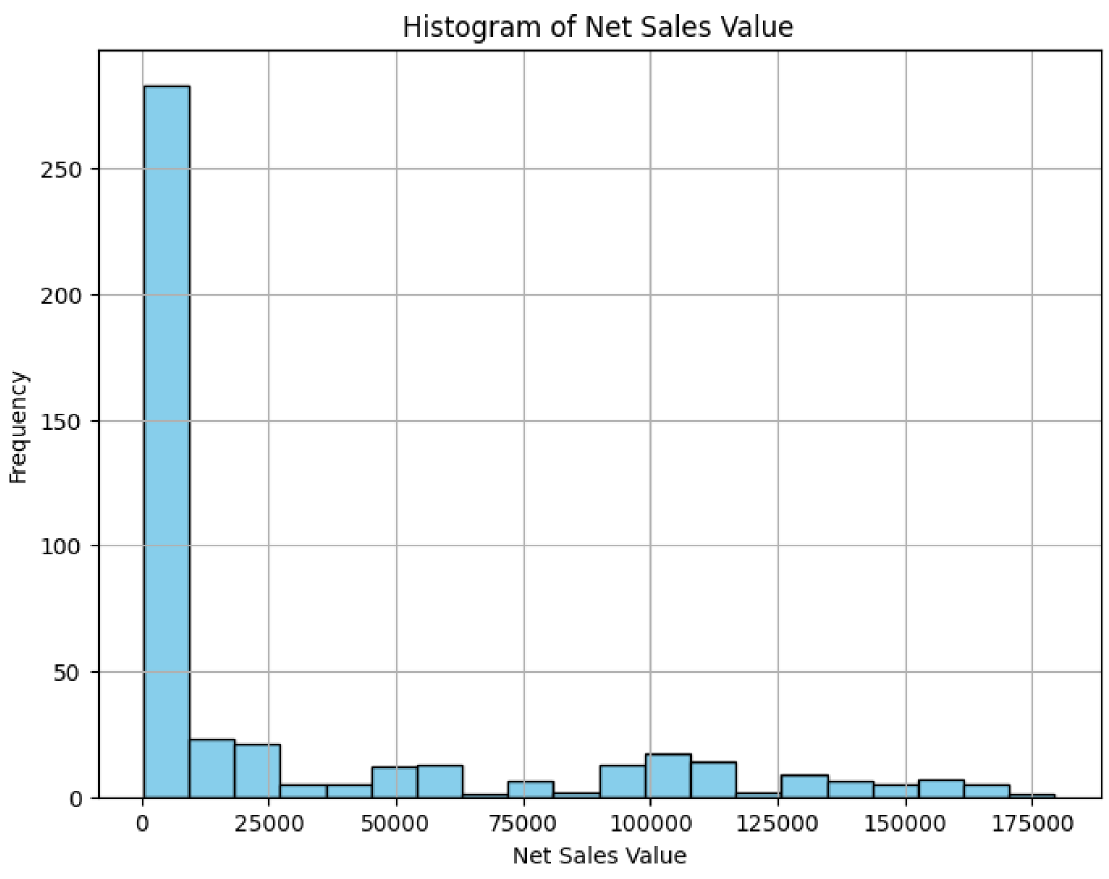

Histogram of Volume
Frequency
Volume
0
20
40
60
80
100
0
5
10
15
20
25
30


Histogram of Avg Price
Frequency
Avg Price
0
50
100
150
200
250
300
0
10000
20000
30000
40000
50000
60000

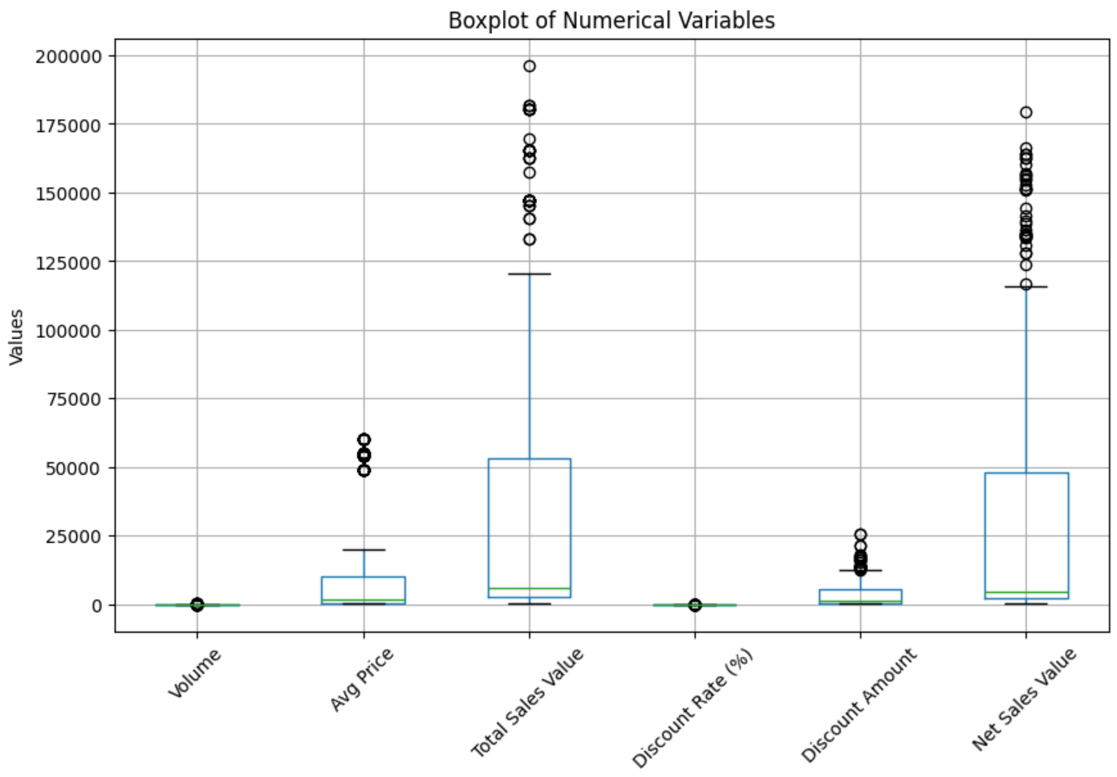

Histogram of Total Sales Value
Frequency
Total Sales Value
0
50
100
150
200
250
0
25000
50000
75000
100000
125000
150000
175000
200000


Histogram of Discount Rate (%)
Frequency
Discount Rate (%)
0
10
20
30
40
50
60
6
8
10
12
14
16
18
20


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



[7]:
```
# Creating boxplots for all numerical columns
plt.figure(figsize=(10, 6))
df[numerical_columns].boxplot()
plt.title('Boxplot of Numerical Variables')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



```
[8]: # Selecting categorical columns
     categorical_columns = df.select_dtypes(include=['object']).columns
     categorical_columns
```

```
[8]: Index(['Date', 'Day', 'SKU', 'City', 'BU', 'Brand', 'Model'], dtype='object')
```

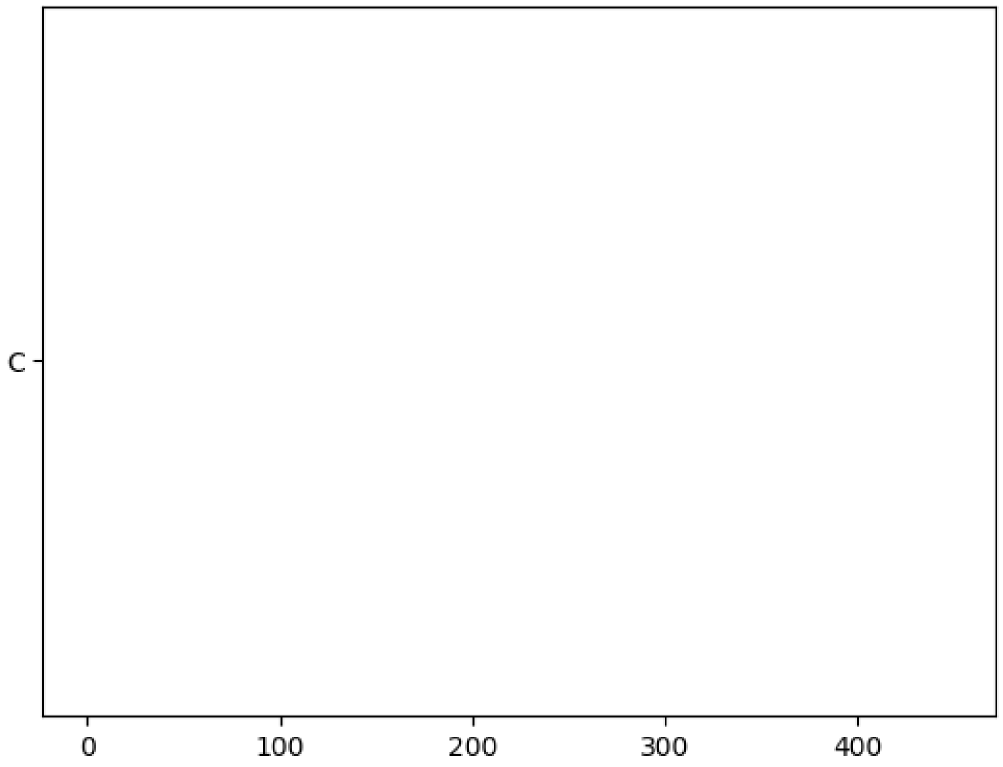
```
[9]: categorical_columns=categorical_columns.drop(['Date','Day','SKU'])
```

```
[10]: df.City.value_counts()
```

```
[10]: City
      C    450
      Name: count, dtype: int64
```

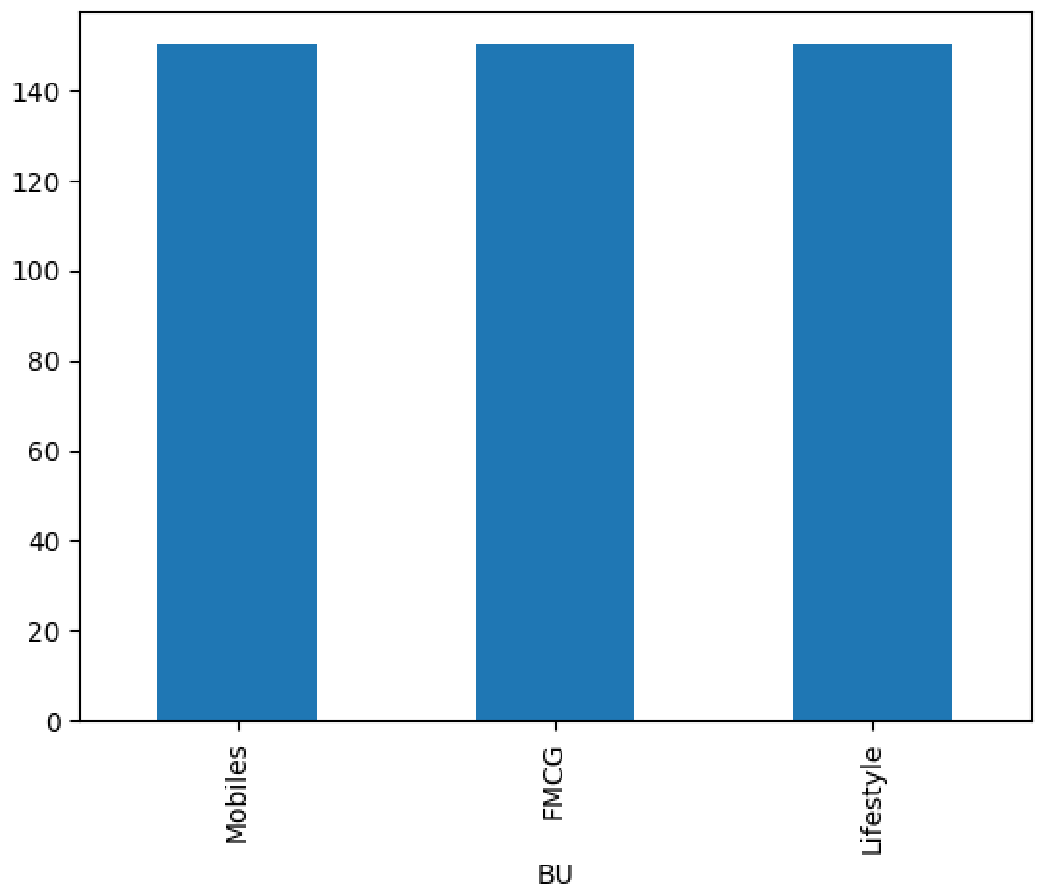
```
[11]: plt.bar(df.index,df.City)
```

```
[11]: <BarContainer object of 450 artists>
```



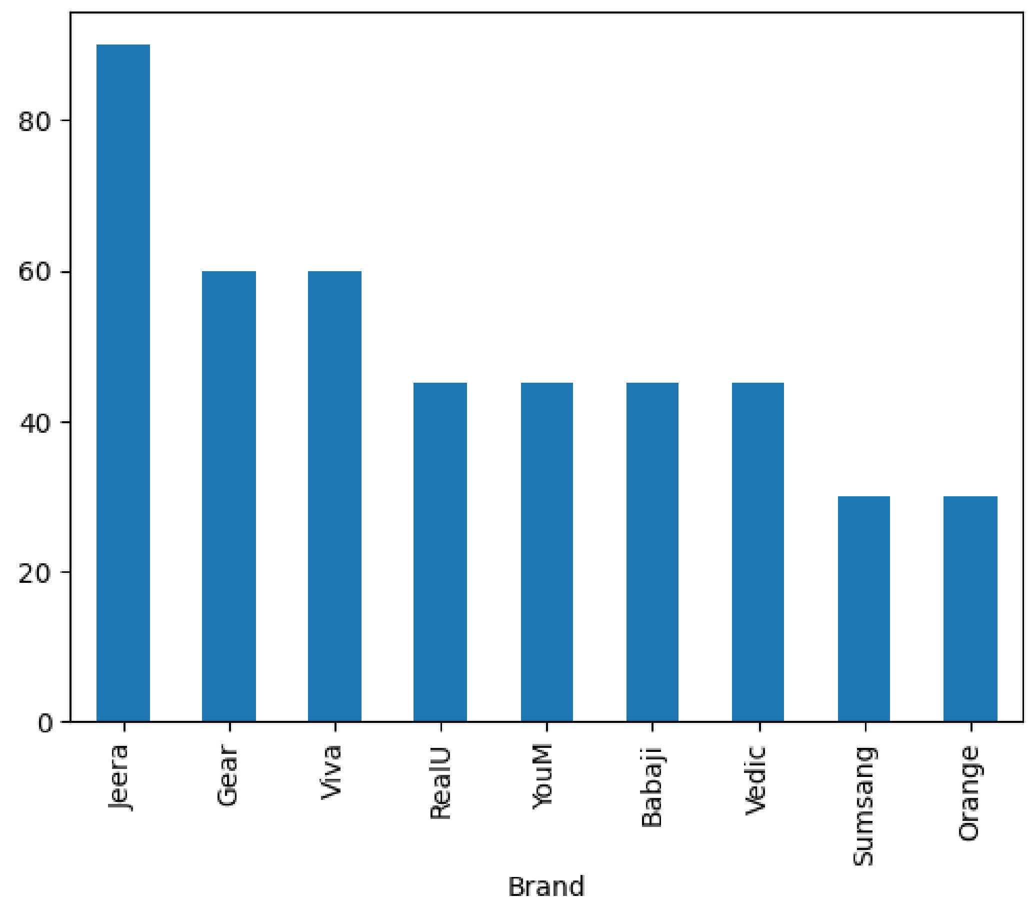
```
[12]: df.BU.value_counts().plot(kind='bar')
```

```
[12]: <Axes: xlabel='BU'>
```



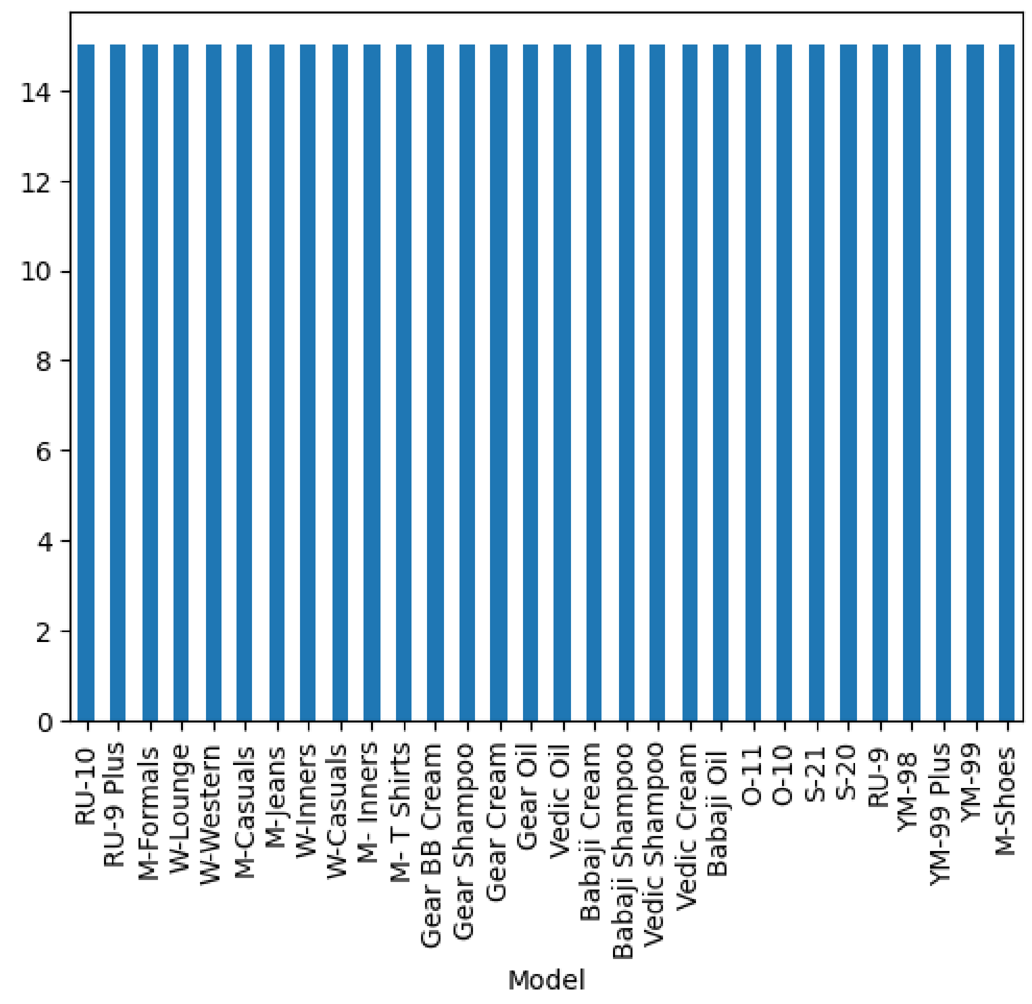
```
[13]: df.Brand.value_counts().plot(kind="bar")
```

```
[13]: <Axes: xlabel='Brand'>
```



```
[14]: df.Model.value_counts().plot(kind="bar")
```

```
[14]: <Axes: xlabel='Model'>
```



[15]:
```
#Standardization
def standardize_column(column):
    mean = column.mean()
    std_dev = column.std()
    standardized_column = (column - mean) / std_dev
    return standardized_column
```

[16]:
```
# Applying standardization to each numerical column
for column in numerical_columns:
    df[column] = standardize_column(df[column])

# Now, numerical columns are standardized
print(df[numerical_columns].head())
```

```
     Volume  Avg Price  Total Sales Value  Discount Rate (%)  Discount Amount  \
0  2.347417   0.091072           2.922469          -0.829365         3.948422
```

```
1  1.165831  -0.019548           1.329516           -0.851714           1.846958
2  0.456880   0.312312           1.561038           -1.350129           1.621190
3  0.220563   0.533552           1.717365           -1.947555           1.112568
4 -0.488389  -0.130168          -0.188242            0.672990           0.227598

   Net Sales Value
0         2.801638
1         1.269613
2         1.543957
3         1.763847
4        -0.227342
```

[17]:
```python
# Converting numerical columns into dummy variables
dummy_df = pd.get_dummies(df[categorical_columns])

dummy_df=dummy_df.astype(int)
```

[18]:
```python
# Concatenating the original DataFrame with the dummy variables
df_with_dummies = pd.concat([df.drop(categorical_columns, axis=1), dummy_df],␣
 ↪axis=1)
```

[19]:
```python
# Now, df_with_dummies contains the original DataFrame with numerical columns␣
 ↪converted to dummy variables
print(df_with_dummies.head())
```

```
         Date       Day  SKU    Volume  Avg Price  Total Sales Value  \
0  01-04-2021  Thursday  M01  2.347417   0.091072           2.922469
1  01-04-2021  Thursday  M02  1.165831  -0.019548           1.329516
2  01-04-2021  Thursday  M03  0.456880   0.312312           1.561038
3  01-04-2021  Thursday  M04  0.220563   0.533552           1.717365
4  01-04-2021  Thursday  M05 -0.488389  -0.130168          -0.188242

   Discount Rate (%)  Discount Amount  Net Sales Value  City_C  …  \
0          -0.829365         3.948422         2.801638       1  …
1          -0.851714         1.846958         1.269613       1  …
2          -1.350129         1.621190         1.543957       1  …
3          -1.947555         1.112568         1.763847       1  …
4           0.672990         0.227598        -0.227342       1  …

   Model_Vedic Cream  Model_Vedic Oil  Model_Vedic Shampoo  Model_W-Casuals  \
0                  0                0                    0                0
1                  0                0                    0                0
2                  0                0                    0                0
3                  0                0                    0                0
4                  0                0                    0                0

   Model_W-Inners  Model_W-Lounge  Model_W-Western  Model_YM-98  Model_YM-99  \
0               0               0                0            0            0
```

```
1                0                0                 0              0             0
2                0                0                 0              0             1
3                0                0                 0              0             0
4                0                0                 0              1             0

   Model_YM-99 Plus
0                 0
1                 0
2                 0
3                 1
4                 0

[5 rows x 52 columns]
```

```
[ ]:
```