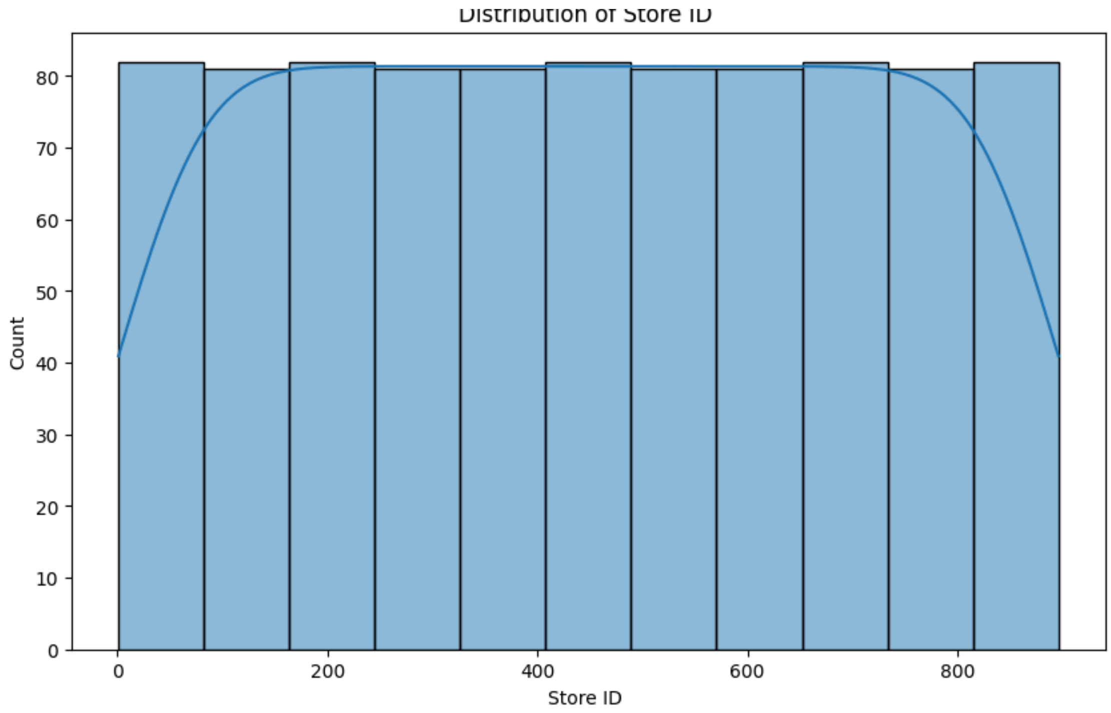
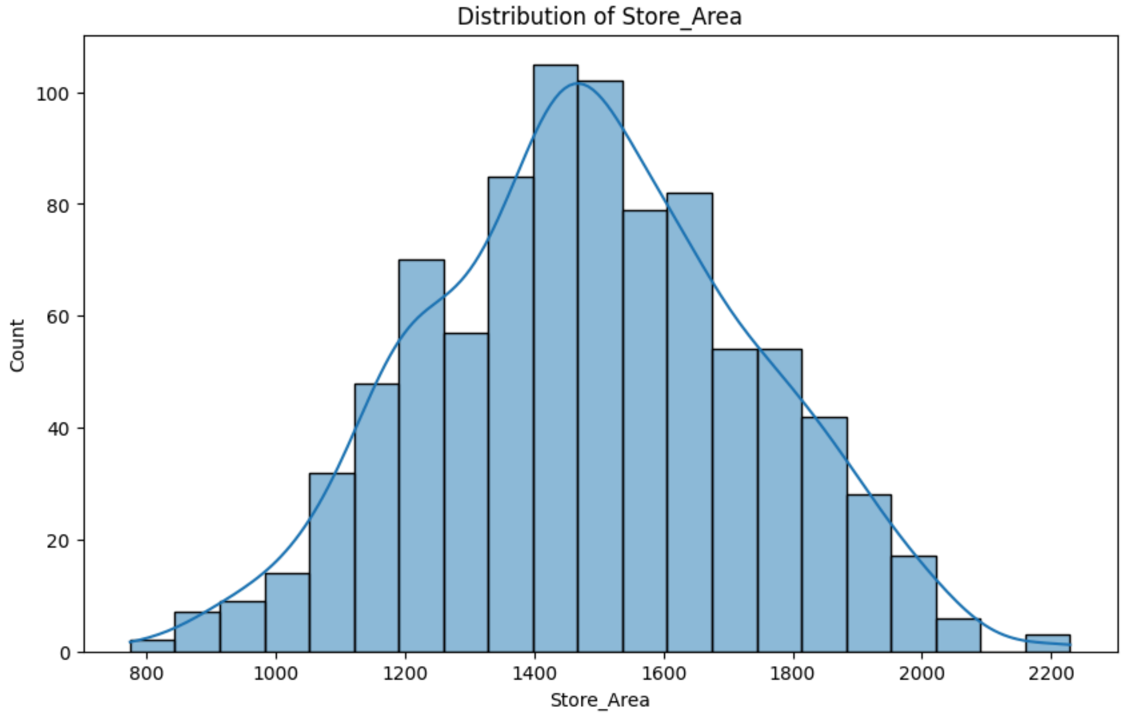
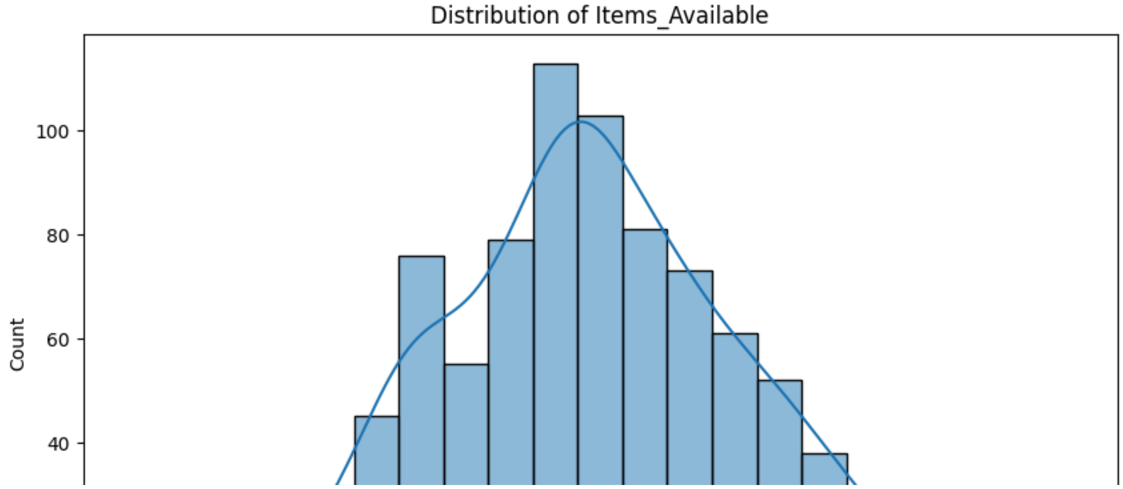
```
import pandas as pd

# Load the data
file_path = '/content/Stores.csv'
df = pd.read_csv(file_path)

print(df.info())

print(df.head())

print(df.isnull().sum())

print(df.describe())
```

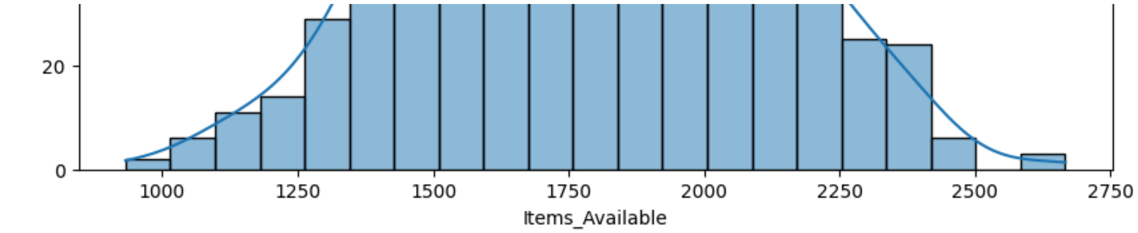
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 896 entries, 0 to 895
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Store ID              896 non-null    int64
 1   Store_Area            896 non-null    int64
 2   Items_Available       896 non-null    int64
 3   Daily_Customer_Count  896 non-null    int64
 4   Store_Sales           896 non-null    int64
dtypes: int64(5)
memory usage: 35.1 KB
None
   Store ID   Store_Area  Items_Available  Daily_Customer_Count  Store_Sales
0         1        1659             1961                   530        66490
1         2        1461             1752                   210        39820
2         3        1340             1609                   720        54010
3         4        1451             1748                   620        53730
4         5        1770             2111                   450        46620
Store ID                0
Store_Area              0
Items_Available         0
Daily_Customer_Count    0
Store_Sales             0
dtype: int64
         Store ID    Store_Area  Items_Available  Daily_Customer_Count  \
count  896.000000   896.000000       896.000000            896.000000
mean   448.500000  1485.409598      1782.035714            786.350446
std    258.797218   250.237011       299.872053            265.389281
min      1.000000   775.000000       932.000000             10.000000
25%    224.750000  1316.750000      1575.500000            600.000000
50%    448.500000  1477.000000      1773.500000            780.000000
75%    672.250000  1653.500000      1982.750000            970.000000
max    896.000000  2229.000000      2667.000000           1560.000000

         Store_Sales
count     896.000000
mean    59351.305804
std     17190.741895
min     14920.000000
25%     46530.000000
50%     58605.000000
75%     71872.500000
max    116320.000000
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting distribution for numeric features
numeric_features = df.select_dtypes(include=['int64', 'float64']).columns

for feature in numeric_features:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[feature], kde=True)
    plt.title(f'Distribution of {feature}')
    plt.show()
categorical_features = df.select_dtypes(include=['object']).columns

for feature in categorical_features:
    plt.figure(figsize=(10, 6))
    sns.countplot(x=df[feature])
    plt.title(f'Distribution of {feature}')
    plt.show()
```

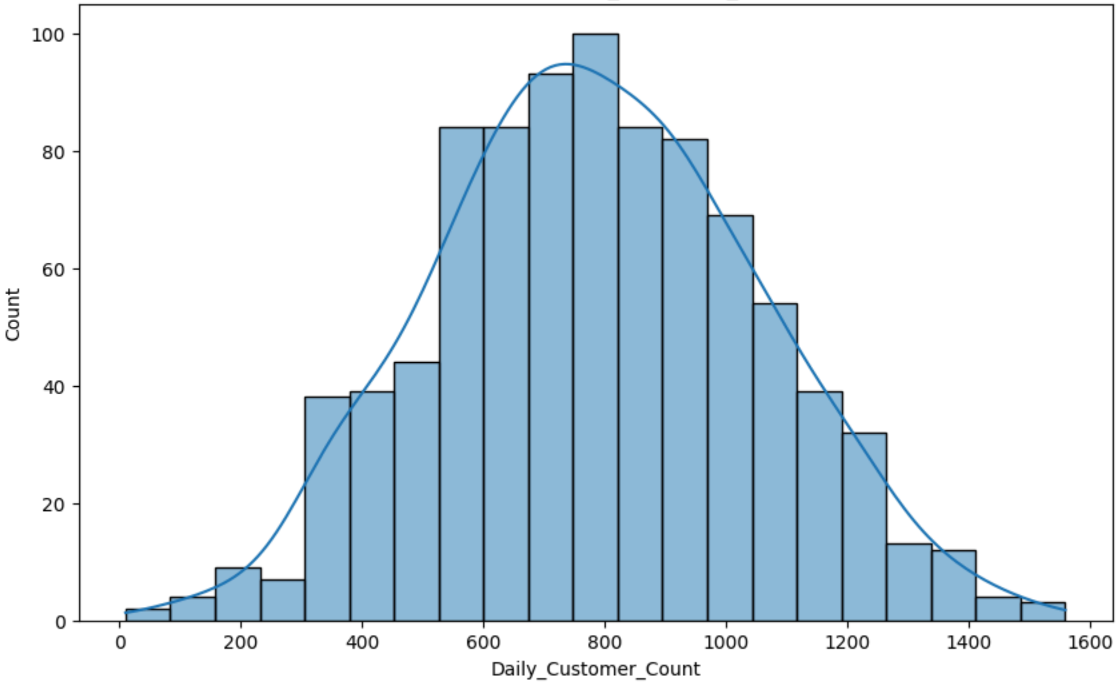

Distribution of Store ID
Count
0
10
20
30
40
50
60
70
80
0
200
400
600
800
Store ID


Distribution of Store_Area
Count
0
20
40
60
80
100
800
1000
1200
1400
1600
1800
2000
2200
Store_Area

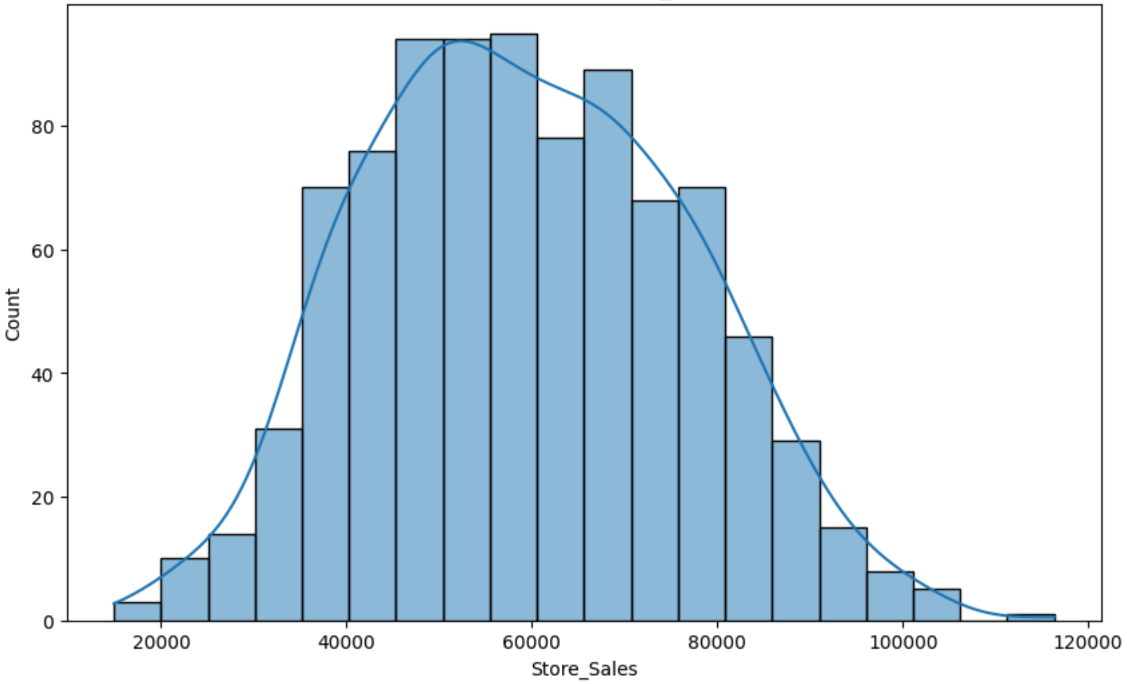

Distribution of Items_Available
Count
40
60
80
100



Distribution of Daily_Customer_Count



Distribution of Store_Sales



```
# Correlation matrix for numeric features
plt.figure(figsize=(12, 8))
correlation_matrix = df[numeric_features].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Pairplot to visualize relationships between numeric variables
sns.pairplot(df[numeric_features])
plt.show()

# Bivariate analysis for categorical vs numeric features
for cat_feature in categorical_features:
    for num_feature in numeric_features:
        plt.figure(figsize=(10, 6))
        sns.boxplot(x=df[cat_feature], y=df[num_feature])
        plt.title(f'{cat_feature} vs {num_feature}')
        plt.show()
```