Sales Data Analysis for Retail Store

In [2]:
```
# Import Library

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

In [3]:
```
# Create the dataset
np.random.seed(42)              # random number wont be generate

# create dataset

data = {
    'product_id' : range(1,21),
    'product_name':[f'Product{i}' for i in range(1, 21)],
    'category' : np.random.choice(['Electronic','Clothing', 'Home', 'Sports
    'units_sold': np.random.poisson(lam = 20, size = 20),
    'sales_data' : pd.date_range(start = '2023-01-01', periods=20, freq='D'
}

sales_data = pd.DataFrame(data)

print('Sales Data:')
print(sales_data)
```

```
Sales Data:
    product_id product_name    category  units_sold sales_data
0            1     Product1        Home          25 2023-01-01
1            2     Product2      Sports          15 2023-01-02
2            3     Product3  Electronic          17 2023-01-03
3            4     Product4        Home          19 2023-01-04
4            5     Product5        Home          21 2023-01-05
5            6     Product6      Sports          17 2023-01-06
6            7     Product7  Electronic          19 2023-01-07
7            8     Product8  Electronic          16 2023-01-08
8            9     Product9        Home          21 2023-01-09
9           10    Product10    Clothing          21 2023-01-10
10          11    Product11        Home          17 2023-01-11
11          12    Product12        Home          22 2023-01-12
12          13    Product13        Home          14 2023-01-13
13          14    Product14        Home          17 2023-01-14
14          15    Product15      Sports          17 2023-01-15
15          16    Product16  Electronic          21 2023-01-16
16          17    Product17      Sports          21 2023-01-17
17          18    Product18      Sports          13 2023-01-18
18          19    Product19      Sports          18 2023-01-19
19          20    Product20        Home          25 2023-01-20
```

In [4]:
```
# save the dataframe to csv file

sales_data.to_csv('sales_data.csv',index=False)
```

In [5]:
```
import os
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Prachi\\FSDS SENAPATI SIR\\Statistics'
```

In [6]:
```
# escriptive statistics
descriptive_stats = sales_data['units_sold'].describe()
print("\n Descriptive statistics for Units Sold:")
print(descriptive_stats)

mean_sales = sales_data['units_sold'].mean()
median_sales = sales_data['units_sold'].median()
mode_sales = sales_data['units_sold'].mode()[0]
variance_sales = sales_data['units_sold'].var()
std_deviation_sales = sales_data['units_sold'].std()

category_stats = sales_data.groupby('category')['units_sold'].agg(['sum', '

# display the result
print('\n Statistical Analysis:')
print(f"Mean Units Sold:  {mean_sales}")
print(f"Median Units Sold: {median_sales}")
print(f"Variance Units Sold: {variance_sales}")
print(f"Standard Deviation Units Sold: {mean_sales}")
print("\n Category Statistics")
print(category_stats)
```

```
 Descriptive statistics for Units Sold:
count    20.000000
mean     18.800000
std       3.302312
min      13.000000
25%      17.000000
50%      18.500000
75%      21.000000
max      25.000000
Name: units_sold, dtype: float64

 Statistical Analysis:
Mean Units Sold:  18.8
Median Units Sold: 18.5
Variance Units Sold: 10.90526315789474
Standard Deviation Units Sold: 18.8

 Category Statistics
     category  sum       mean       std
0    Clothing   21  21.000000       NaN
1  Electronic   73  18.250000  2.217356
2        Home  181  20.111111  3.723051
3      Sports  101  16.833333  2.714160
```

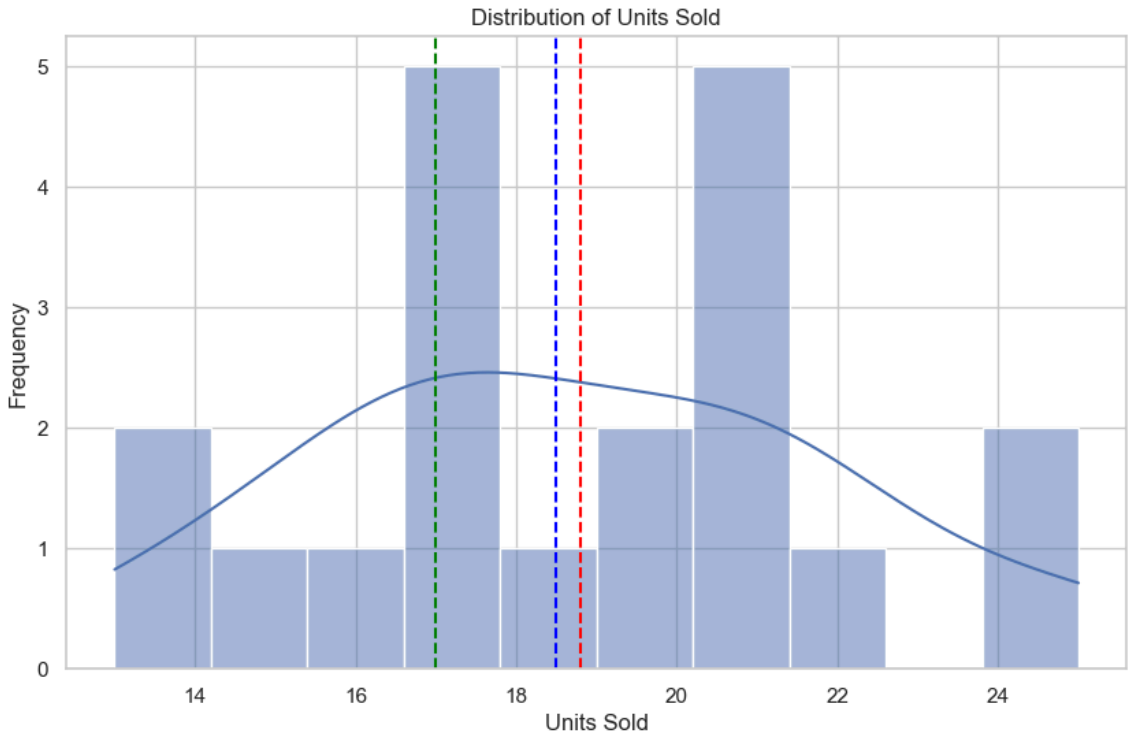
In [7]:
```
# inferential Statistics

confidence_level = 0.95

degrees_freedom = len(sales_data['units_sold']) - 1
sample_mean = mean_sales
sample_standard_error = std_deviation_sales / np.sqrt(len((sales_data['unit

# t-score
t_score = stats.t.ppf((1 + confidence_level) / 2, degrees_freedom)
margin_of_error = t_score + sample_standard_error

confidence_interval = (sample_mean - margin_of_error, sample_mean + margin_
print("\n Confidence Interval for mean of Units Sold")
print(confidence_interval)
```

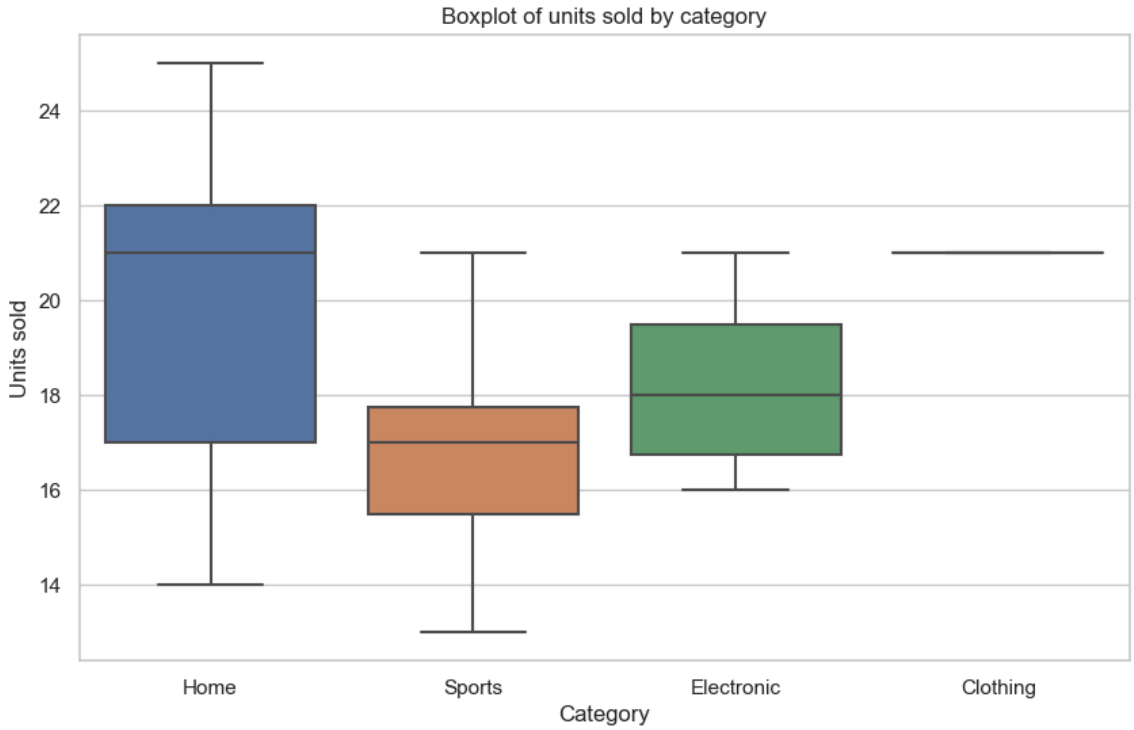
```
 Confidence Interval for mean of Units Sold
(15.96855658107403, 21.631443418925972)
```

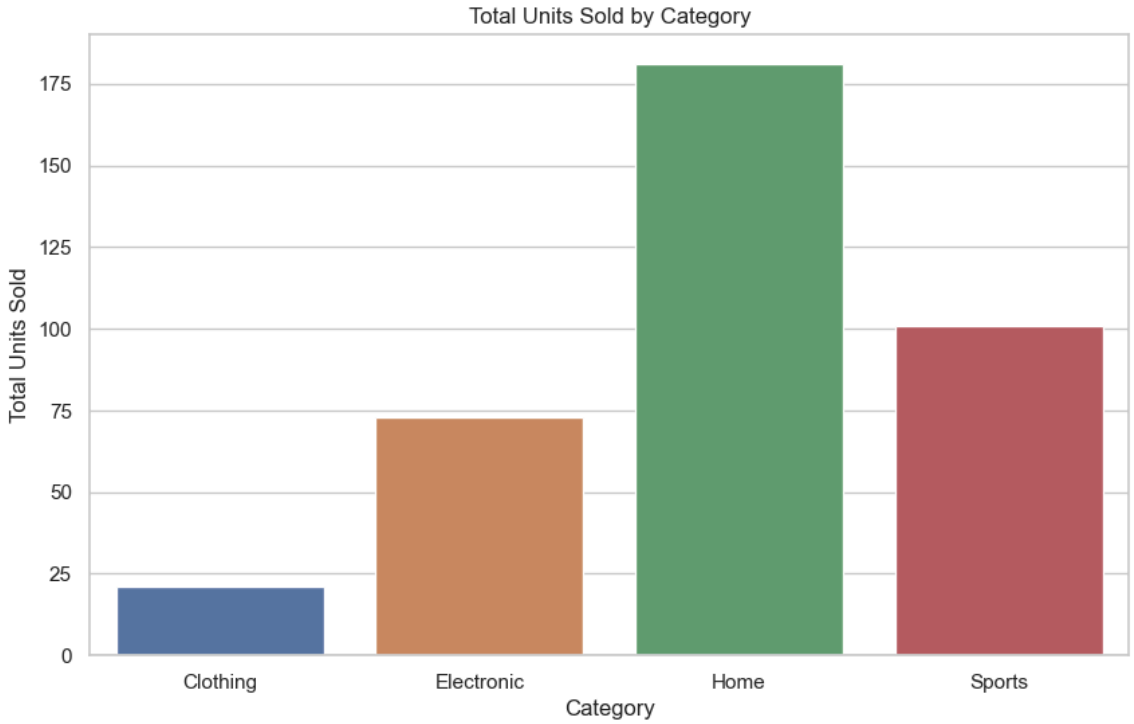
In [8]:
```
t_statistic, p_value = stats.ttest_1samp(sales_data['units_sold'], 20)

print("\n Hypothesis Testing (t-test):")
print(f"T-statistic : {t_statistic}, p-value: {p_value}")

if p_value < 0.05 :
    print("Reject the null hypothesis: The mean units sold is different  fr
else:
    print("Fail to reject the null hypothesis: The mean units sold is not d
```

```
 Hypothesis Testing (t-test):
T-statistic : -1.6250928099424466, p-value: 0.12061572226781002
Fail to reject the null hypothesis: The mean units sold is not different f
rom 20
```

In [9]:
```
# Visualization
sns.set(style = 'whitegrid')

#plot the distribution of units sold

plt.figure(figsize=(10,6))
sns.histplot(sales_data['units_sold'], bins=10, kde=True)
plt.title("Distribution of Units Sold")
plt.xlabel('Units Sold')
plt.ylabel('Frequency')
plt.axvline(mean_sales, color = 'red', linestyle = '--', label = 'Mean')
plt.axvline(median_sales, color = 'blue', linestyle = '--', label = 'Median
plt.axvline(mode_sales, color = 'green', linestyle = '--', label = 'Mode')
plt.legend
plt.show()

# Boxplot for units sold by category
plt.figure(figsize=(10,6))
sns.boxplot(x='category', y='units_sold', data= sales_data)
plt.title('Boxplot of units sold by category')
plt.xlabel('Category')
plt.ylabel('Units sold')
plt.show()


category_stats = sales_data.groupby("category")["units_sold"].sum().reset_i



#bar plot for total units sold by category
plt.figure(figsize=(10, 6))
sns.barplot(x='category', y ='units_sold', data=category_stats)
plt.title('Total Units Sold by Category')
plt.xlabel('Category')
plt.ylabel('Total Units Sold')
plt.show()
```




Boxplot of units sold by category
Units sold
24
22
20
18
16
14
Home
Sports
Electronic
Clothing
Category


Total Units Sold by Category
Total Units Sold
175
150
125
100
75
50
25
0
Clothing
Electronic
Home
Sports
Category

In [ ]: