Project Name - E-commerce Furniture Dataset 2024 (Part 2)

Project Type - Some more Charts and Visualization

Industry - Unified Mentor

Contribution - Individual

Member Name - Hare Krishana Mishra

Task - 2

Project Summary -

Project Description:

The E-commerce Furniture Dataset 2024 project involves analyzing 2,000 entries of furniture product data scraped from AliExpress. The dataset contains product details, pricing, sales numbers, and additional tags, offering insights into consumer purchasing patterns and online furniture market trends. The project applies data analytics and machine learning techniques to explore, visualize, and model sales predictions.

Objective:

Predict the number of furniture items sold (sold) based on product attributes such as:

- productTitle
- originalPrice
- price
- tagText

Key Project Details:

Domain: Data Analytics & Machine Learning

Tech Stack: Python, Pandas, Scikit-learn, Matplotlib, Seaborn

Dataset Features:

- productTitle: Furniture item name
- originalPrice: Price before discounts
- price: Current selling price
- price: Current selling price

- sold: Units sold
- tagText: Extra product info (e.g., "Free shipping")

Let's Begin:-

In [ ]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
df = pd.read_csv('/content/ecommerce_furniture_dataset_2024.csv')
```

Data Collection

In [ ]:
```
df.head()
```

Out[ ]:

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	NaN	$46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	NaN	$169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	$78.4	$39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	NaN	$111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	$48.82	$21.37	1	Free shipping

Data Preprocessing

In [ ]:
```
# Check for missing values
print(df.isnull().sum())
```

```
productTitle        0
originalPrice    1513
price               0
sold                0
tagText             3
dtype: int64
```

In [ ]:
```
df.shape
```

Out[ ]:
```
(2000, 5)
```

In [ ]:
```
#Dropping any rows with missing values (if applicable
df.drop(['originalPrice'],axis=1,inplace=True)
```

In [ ]:

```
df.head()
```

Out[ ]:

	productTitle	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	$46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	$169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	$39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	$111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	$21.37	1	Free shipping

In [ ]:
```
df['tagText'].nunique()
```

Out[ ]:
```
100
```

In [ ]:
```
df['tagText'].value_counts()
```

Out[ ]:

tagText	count
Free shipping	1880
+Shipping: $5.09	9
+Shipping: $239.64	2
+Shipping: $80.21	2
+Shipping: $94.92	2
...	...
+Shipping: $134.27	1
+Shipping: $151.69	1
+Shipping: $41.93	1
+Shipping: $78.61	1
+Shipping: $171.49	1

100 rows × 1 columns

dtype: int64

In [ ]:
```
# Replace all values except 'Free shipping' and '+Shipping: $5.09' with 'others'
df['tagText'] = df['tagText'].apply(
    lambda x: x if isinstance(x, str) and x in ['Free shipping', '+Shipping: $5.09'] els
)
```

In [ ]:

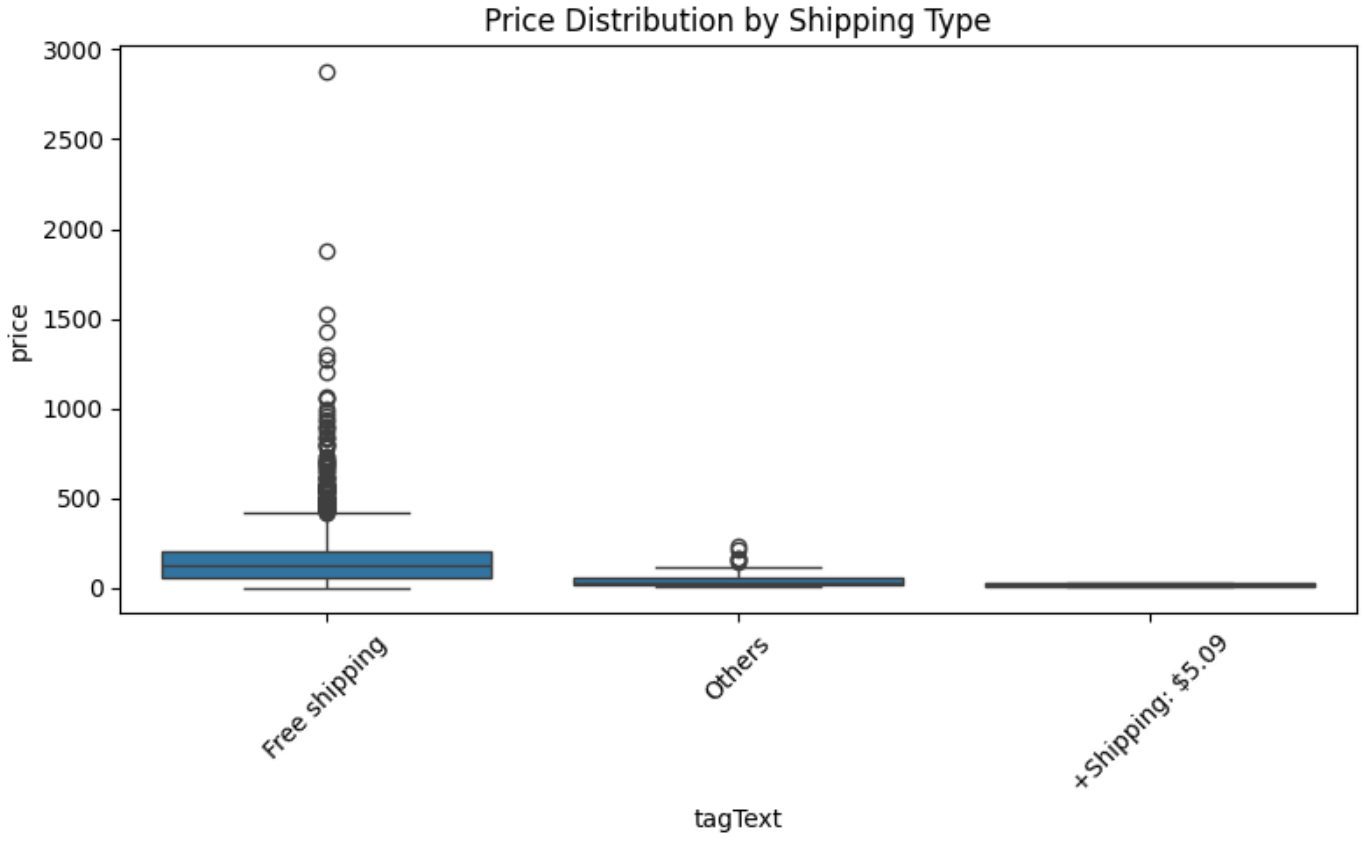
```
# Display the modified value counts
print(df['tagText'].value_counts())
```

```
tagText
Free shipping       1880
others               111
+Shipping: $5.09       9
Name: count, dtype: int64
```

Boxplot: Price distribution by Shipping Type

In [ ]:

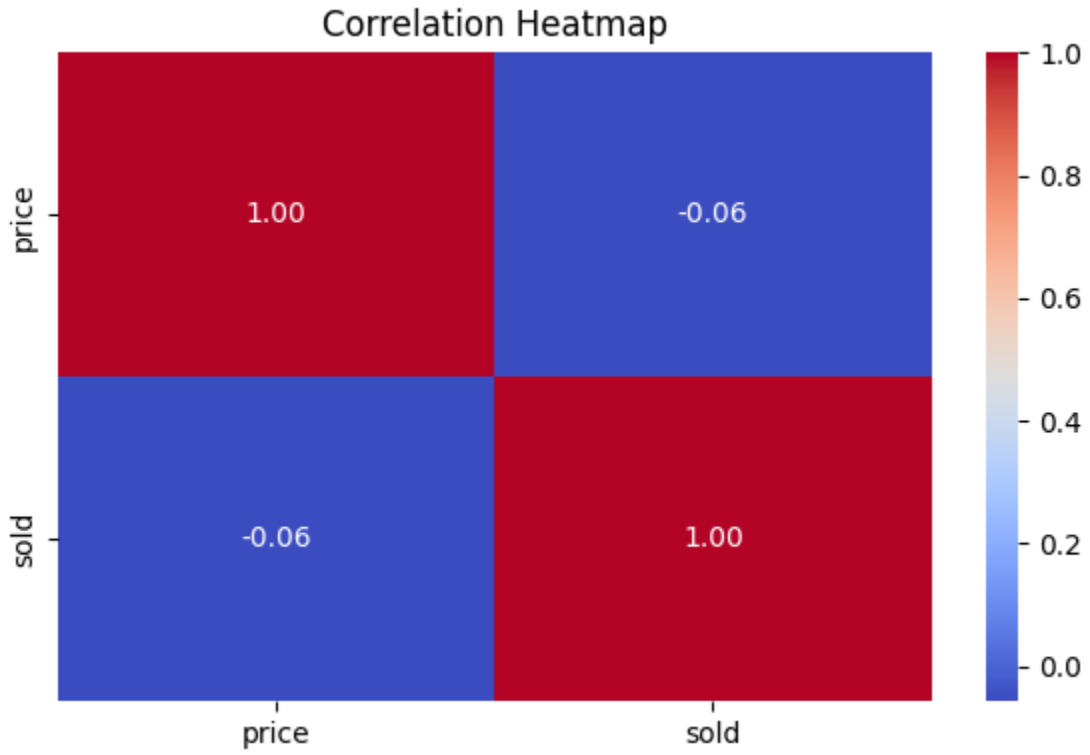
```
# ---------------- Data Cleaning ----------------
# Remove $ sign and commas, convert price to float
df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)

# Handle missing tagText values
df['tagText'] = df['tagText'].fillna('Unknown')

# Group shipping types into categories
df['tagText'] = df['tagText'].apply(lambda x: x if x in ['Free shipping', '+Shipping: $5

# ---------------- 1. Boxplot: Price distribution by Shipping Type ----------------
plt.figure(figsize=(8,5))
sns.boxplot(x='tagText', y='price', data=df)
plt.title('Price Distribution by Shipping Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Correlation Heatmap

In [ ]:

```
plt.figure(figsize=(6,4))
numeric_df = df[['price', 'sold']]
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



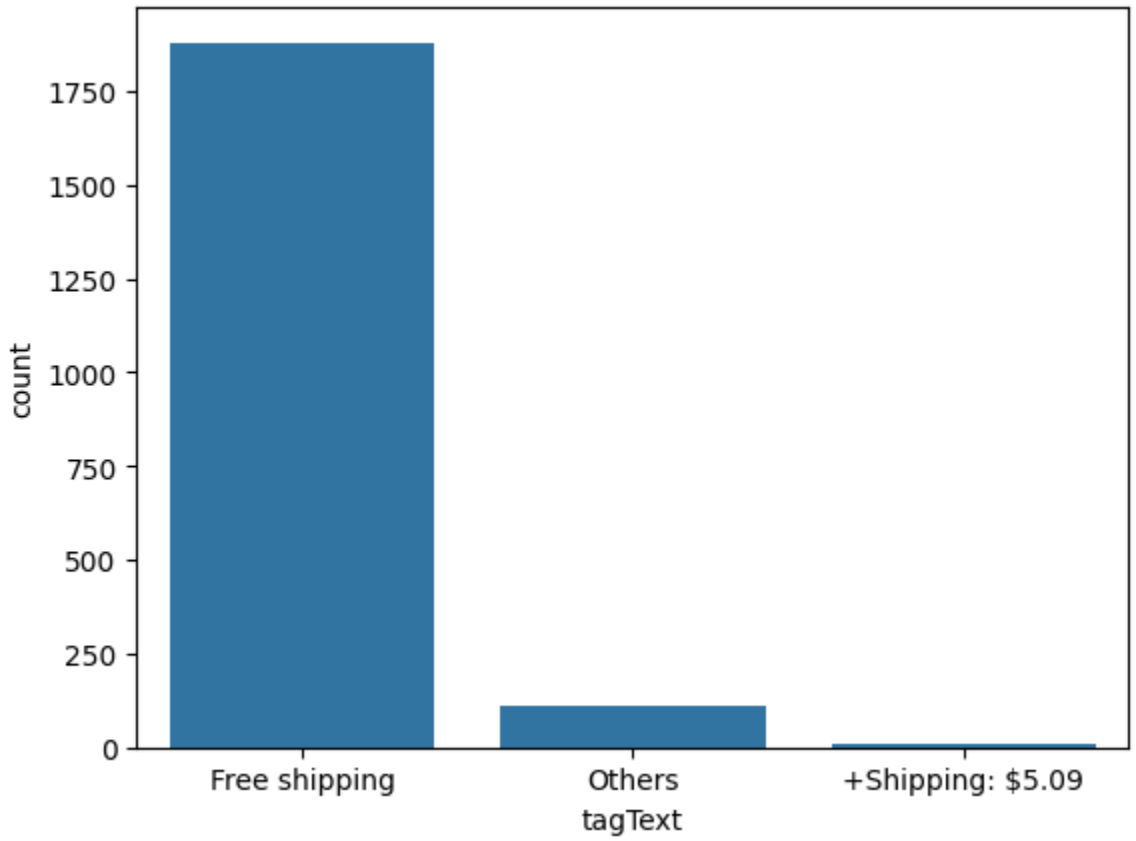
Distribution of Shipping Tags

In [ ]:

```
sns.countplot(x='tagText',data=df)
```

Out[ ]:

```
<Axes: xlabel='tagText', ylabel='count'>
```

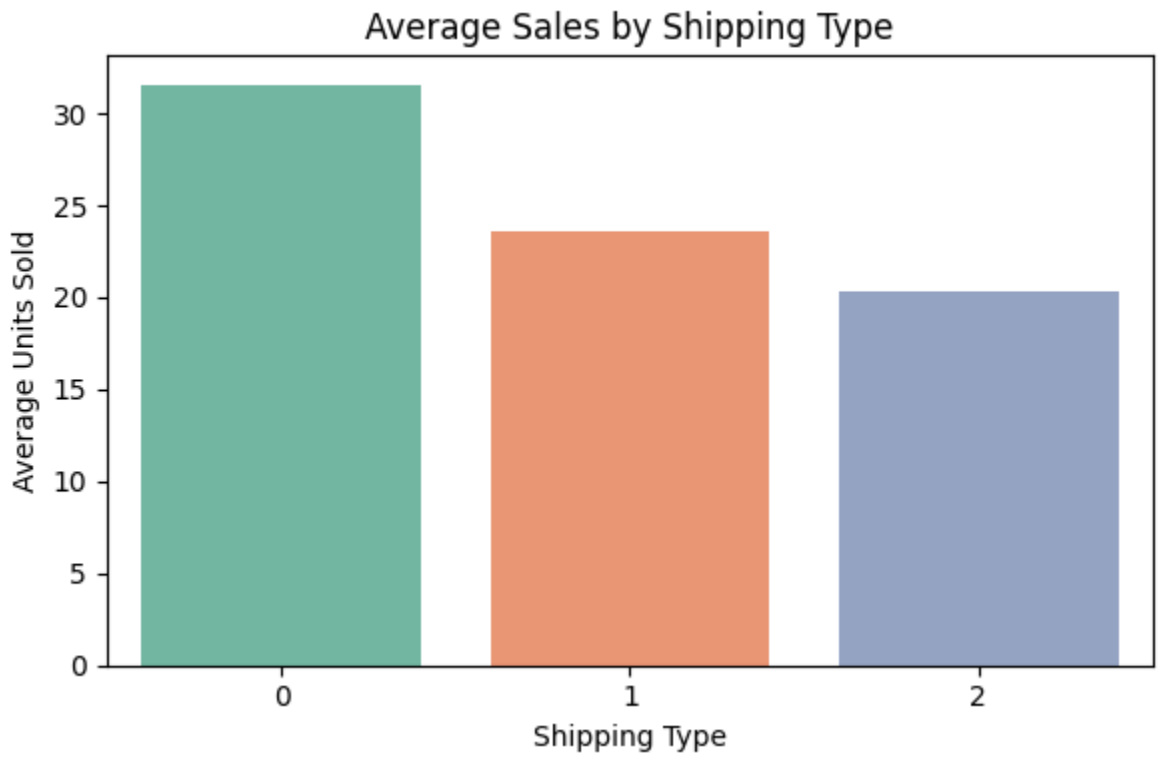


Average Sales by Shipping Type

In [ ]:

```
avg_sales = df.groupby('tagText')['sold'].mean().reset_index()
plt.figure(figsize=(6,4))
sns.barplot(x='tagText', y='sold', data=avg_sales, palette='Set2')
plt.title('Average Sales by Shipping Type')
plt.xlabel('Shipping Type')
plt.ylabel('Average Units Sold')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-3320275257.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.
Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='tagText', y='sold', data=avg_sales, palette='Set2')
```



In [ ]:

```
df['price'] = df['price'].replace('[\$,]', '',
regex=True).astype(float)
```

In [ ]:

```
df.head()
```

Out[ ]:

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	NaN	46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	NaN	169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	$78.4	39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	NaN	111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	$48.82	21.37	1	Free shipping

Distribution of Product Prices

In [ ]:

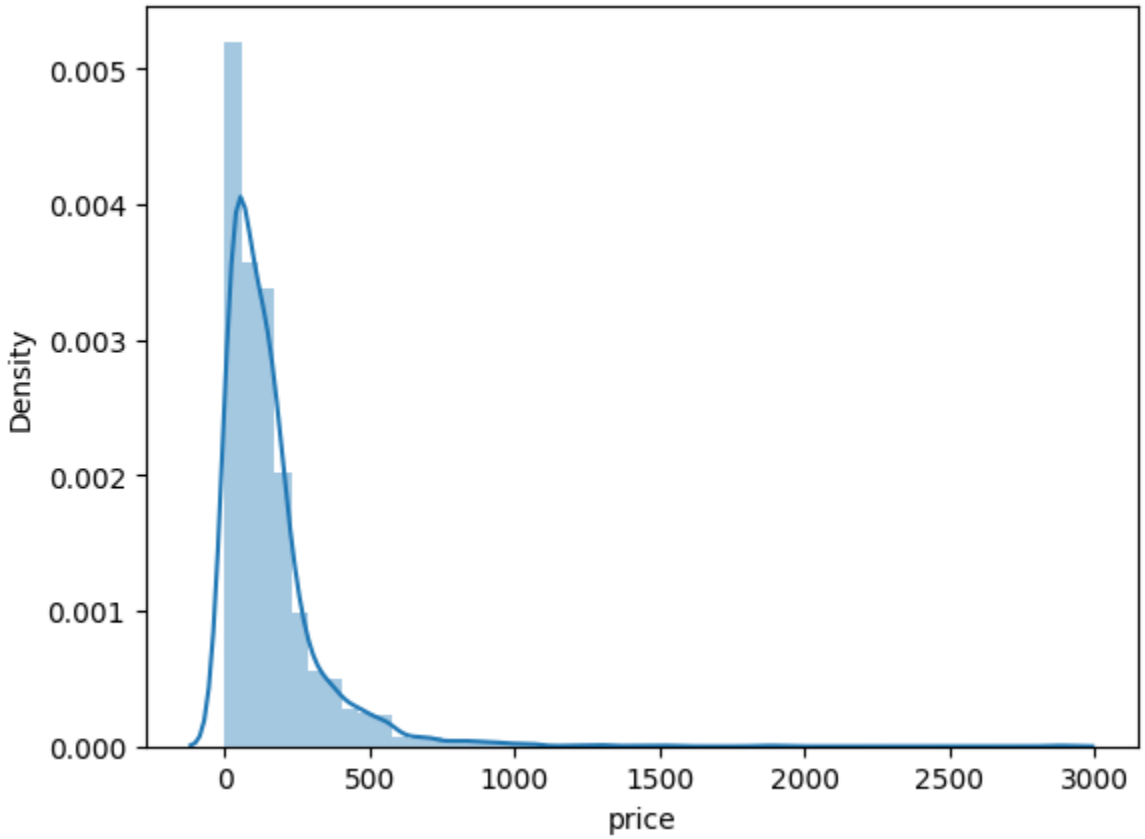
```
sns.distplot(df['price'])
```

```
/tmp/ipython-input-444587821.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(df['price'])
```

Out[ ]:

```
<Axes: xlabel='price', ylabel='Density'>
```



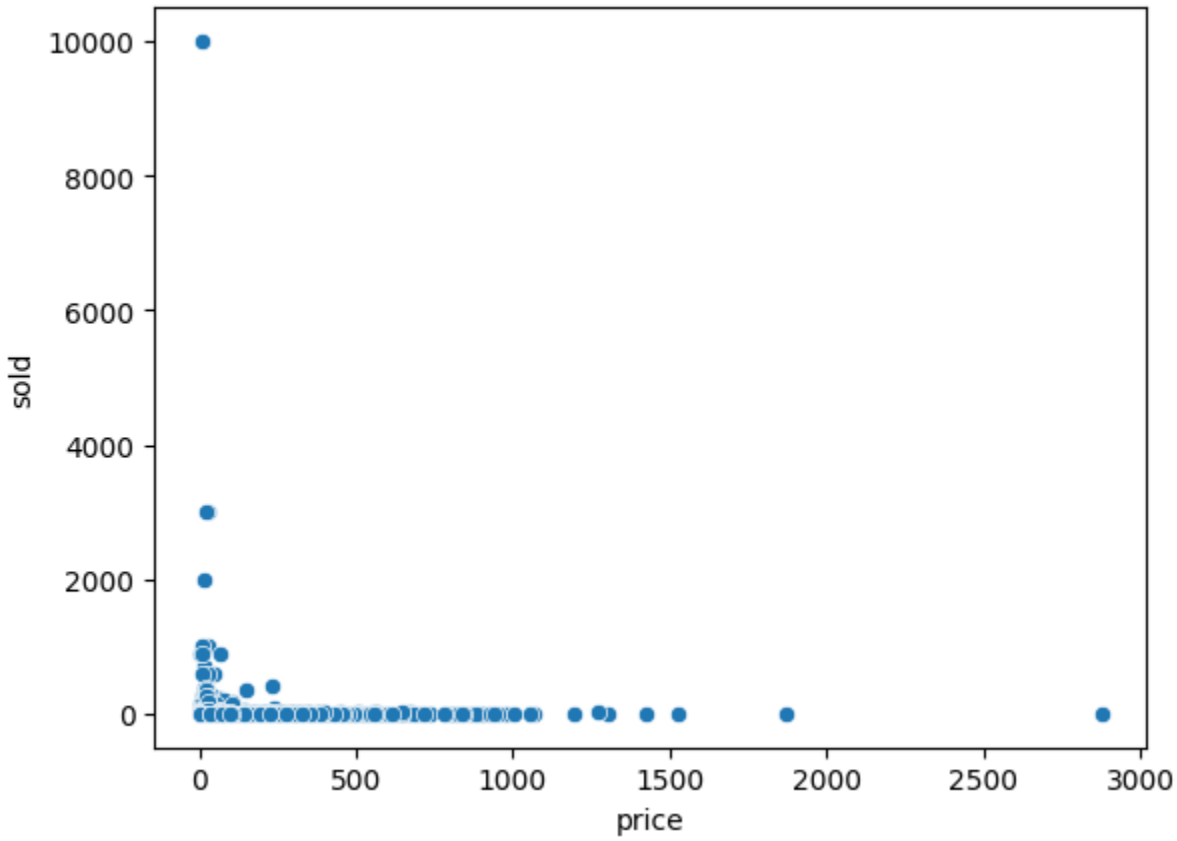
Relationship Between Price and Units Sold

In [ ]:

```
sns.scatterplot(x='price', y='sold', data=df)
```

Out[ ]:

```
<Axes: xlabel='price', ylabel='sold'>
```



Distribution of Items Sold

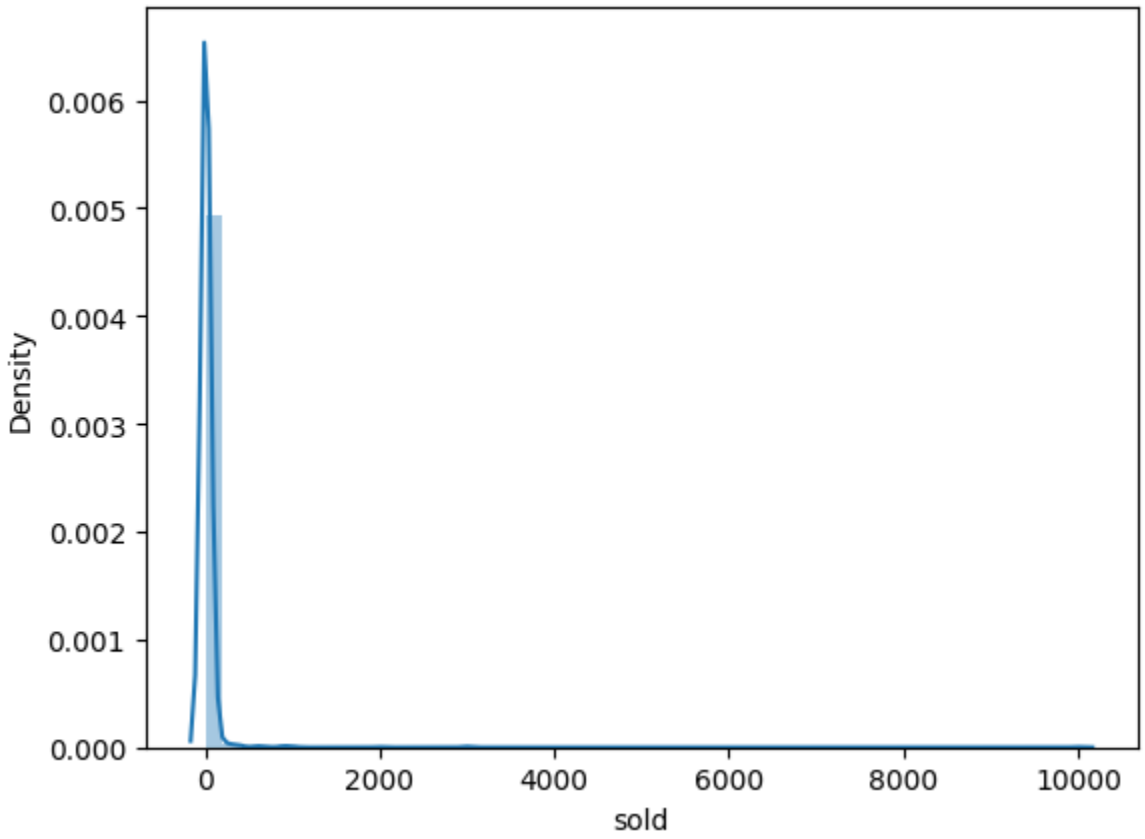
In [ ]:

```
sns.distplot(df['sold'])
```

```
/tmp/ipython-input-2507294489.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['sold'])
```

Out[ ]:

```
<Axes: xlabel='sold', ylabel='Density'>
```



In [ ]:

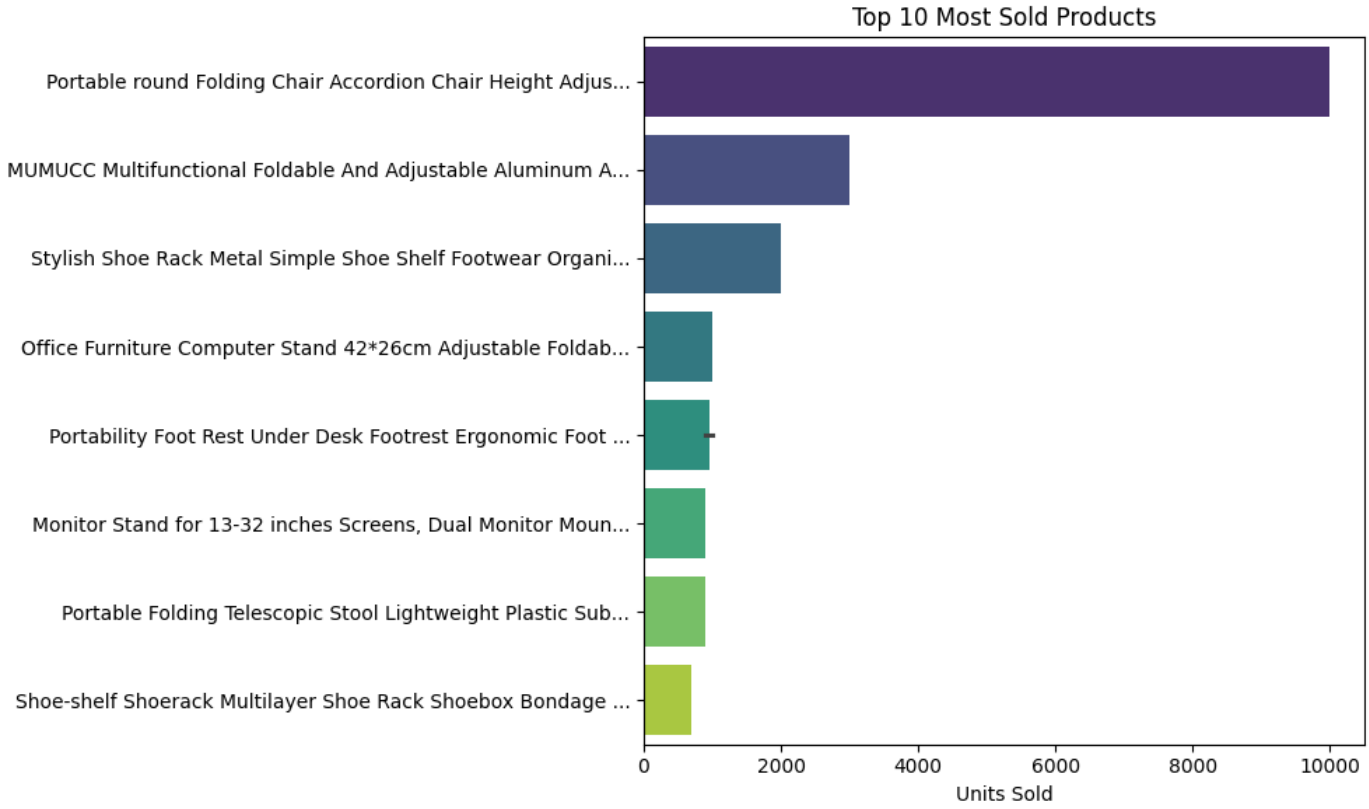
```
filtered_df = df[df['tagText'] == 'Free shipping']
```

Top 10 Best-Selling Furniture Items in 2024 – Horizontal Bar Chart

In [ ]:

```
# 1) Ensure 'sold' is numeric
df['sold'] = pd.to_numeric(df['sold'], errors='coerce').fillna(0).astype(int)

# 2) Clean product titles
df['productTitle'] = df['productTitle'].astype(str).str.replace(r'\s+', ' ', regex=True)

# 3) Get top 10 by 'sold'
top_sold = df.sort_values('sold', ascending=False).head(10).copy()

# 4) Shorten long titles
def short_title(s, n=60):
    return s if len(s) <= n else s[:n-3] + '...'
top_sold['shortTitle'] = top_sold['productTitle'].apply(lambda s: short_title(s, 60))

# 5) Plot horizontal bar chart without warning
plt.figure(figsize=(10,6))
sns.barplot(
    x='sold',
    y='shortTitle',
    hue='shortTitle',      # Add hue to satisfy new API
    data=top_sold,
    orient='h',
    palette='viridis',
    legend=False           # Hide legend since hue duplicates y-axis
)
plt.title('Top 10 Most Sold Products')
```

```
plt.xlabel('Units Sold')
plt.ylabel('')
plt.tight_layout()
plt.show()
```



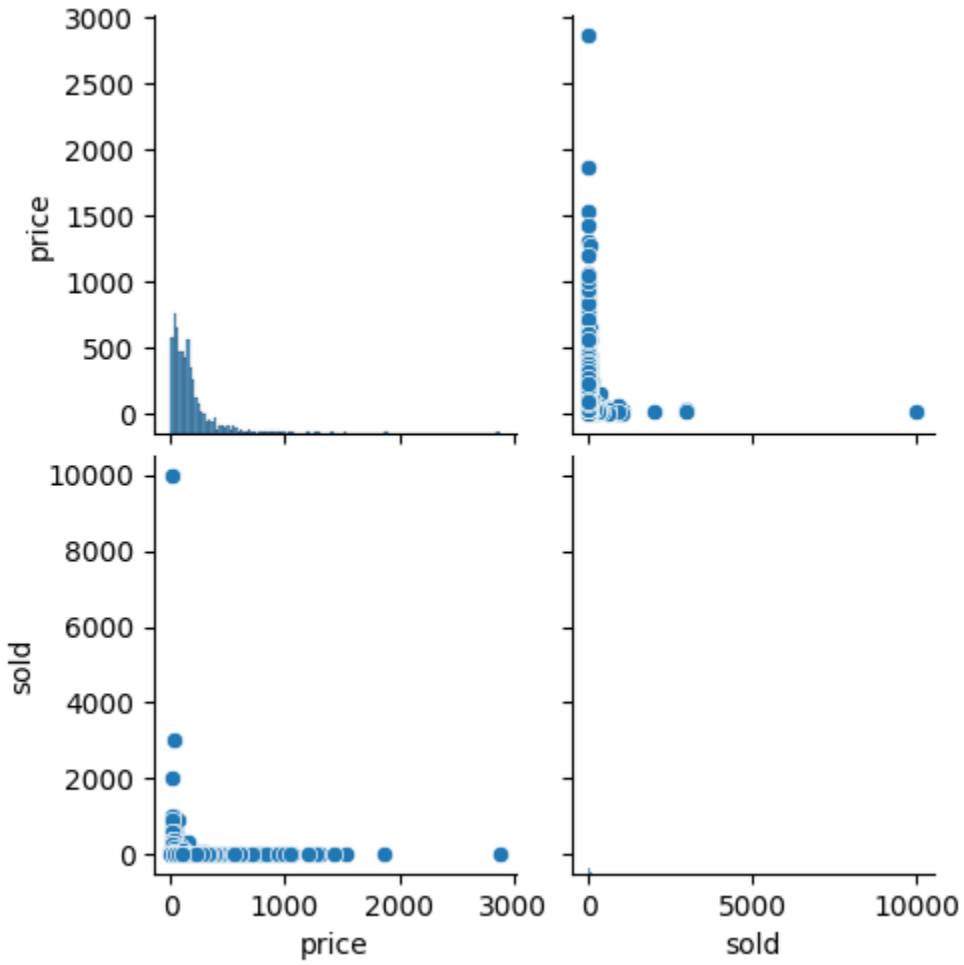
Relationship Between Price and Units Sold (Free Shipping Products)

In [ ]:

```
# Create a pairplot including the 'sold' column and other relevant columns
sns.pairplot(filtered_df[['price', 'sold']])
```

Out[ ]:

```
<seaborn.axisgrid.PairGrid at 0x7859c485c590>
```



In [ ]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['tagText']=le.fit_transform(df['tagText'])
```

In [ ]:

```
df.head()
```

Out[ ]:

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	NaN	46.79	600	1
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	NaN	169.72	0	1
2	Desser For Bedroom With 7 Fabric Drawers Organ...	$78.4	39.46	7	1
3	Modern Accent Boucle Chair,Upholstered Tufted ...	NaN	111.99	0	1
4	Small Unit Simple Computer Desk Household Wood...	$48.82	21.37	1	1

In [ ]:

```
df['tagText'].value_counts()
```

Out[ ]:

tagText	count
1	1880
2	111
0	9

dtype: int64