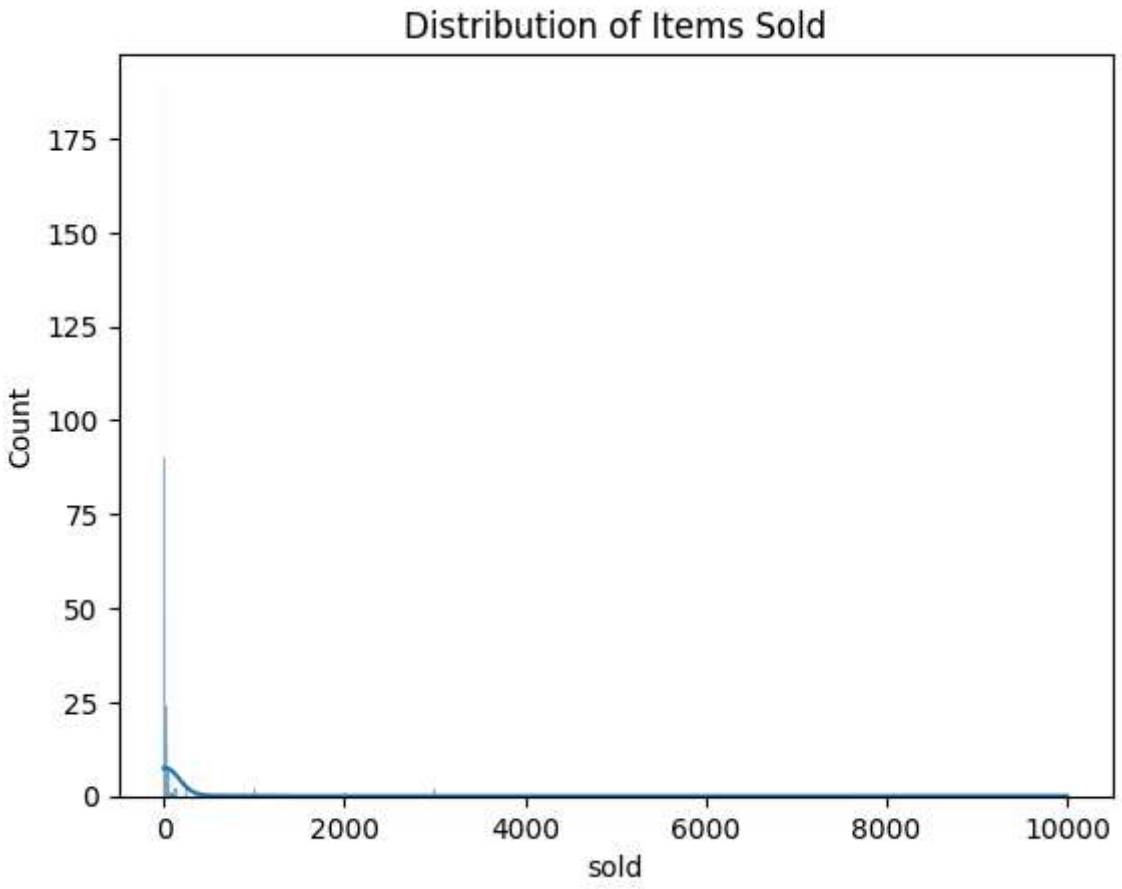
```
# Install necessary packages (if not already)
!pip install seaborn xgboost wordcloud

# Import standard libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Read the dataset
df = pd.read_csv('/content/ecommerce_furniture_dataset_2024.csv')
df.head()
```

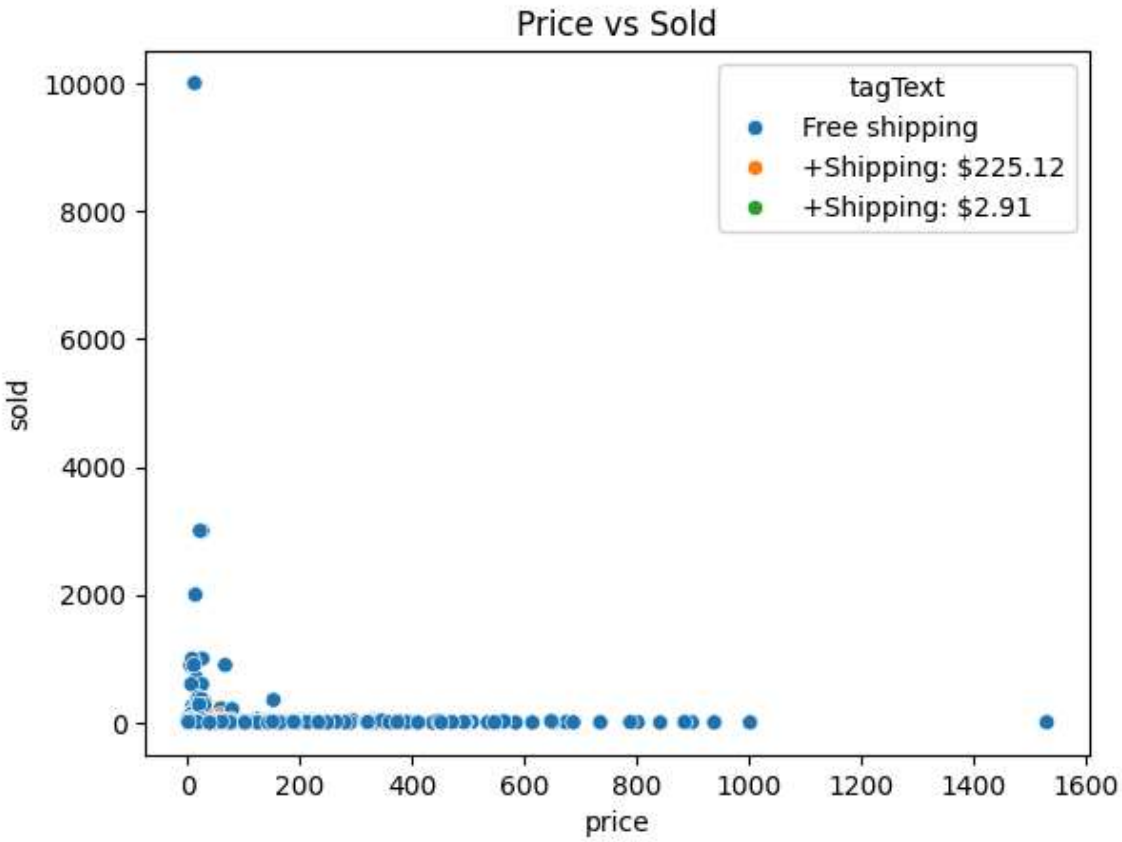
```
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: xgboost in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: wordcloud in /usr/local/lib/python3.11/dist-
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/pytho
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/py
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3.1
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
```

	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

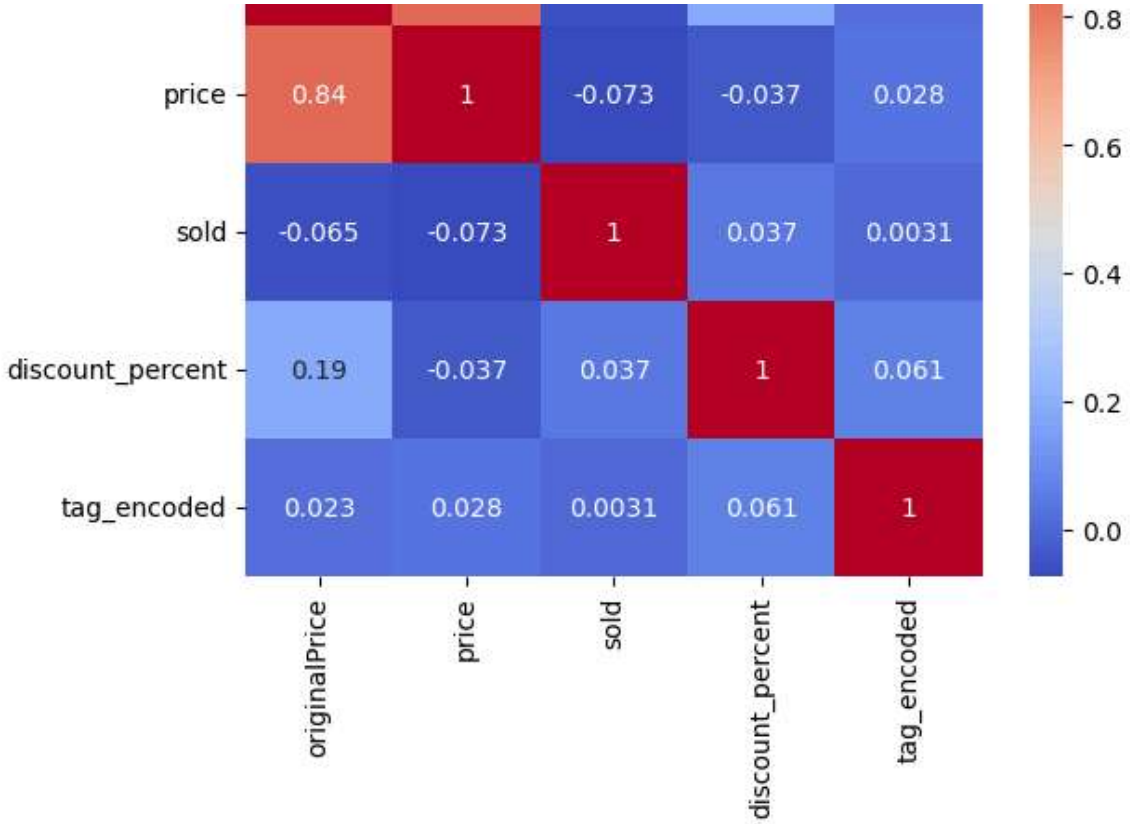
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Remove rows with missing essential values
df.dropna(subset=['price', 'sold'], inplace=True)

# Remove $ sign and convert to float
df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)
df['originalPrice'] = df['originalPrice'].replace('[\$,]', '', regex=True).asty

# Handle missing originalPrice by dropping or filling
df.dropna(subset=['originalPrice'], inplace=True)



# Discount percentage
df['discount_percent'] = ((df['originalPrice'] - df['price']) / df['originalPri

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



# Visualize sales distribution
sns.histplot(df['sold'], kde=True)
plt.title('Distribution of Items Sold')
plt.show()

# Price vs Sold
sns.scatterplot(x='price', y='sold', hue='tagText', data=df)
plt.title('Price vs Sold')
plt.show()

# Heatmap
numeric_df = df.select_dtypes(include=np.number) # Select only numeric columns
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```


Distribution of Items Sold
Count
sold
0
25
50
75
100
125
150
175
0
2000
4000
6000
8000
10000


Price vs Sold
tagText
Free shipping
+Shipping: $225.12
+Shipping: $2.91
sold
price
0
2000
4000
6000
8000
10000
0
200
400
600
800
1000
1200
1400
1600


Correlation Heatmap
originalPrice
1
0.84
-0.065
0.19
0.023
1.0




```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=50, stop_words='english')
title_features = tfidf.fit_transform(df['productTitle'].astype(str))
title_df = pd.DataFrame(title_features.toarray(), columns=tfidf.get_feature_nam

# Combine with original DataFrame
df = pd.concat([df.reset_index(drop=True), title_df], axis=1)
df.drop('productTitle', axis=1, inplace=True)
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Features and target
X = df.drop(['sold', 'tagText'], axis=1)
y = df['sold']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random

# Train model
model = RandomForestRegressor()
```

```
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)



import numpy as np

print("R² Score:", r2_score(y_test, y_pred))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))

# Plot predictions vs actual
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Sold")
plt.ylabel("Predicted Sold")
plt.title("Actual vs Predicted")
plt.show()
```

```
R² Score: 0.4740065015763718
RMSE: 269.05358787605786
```