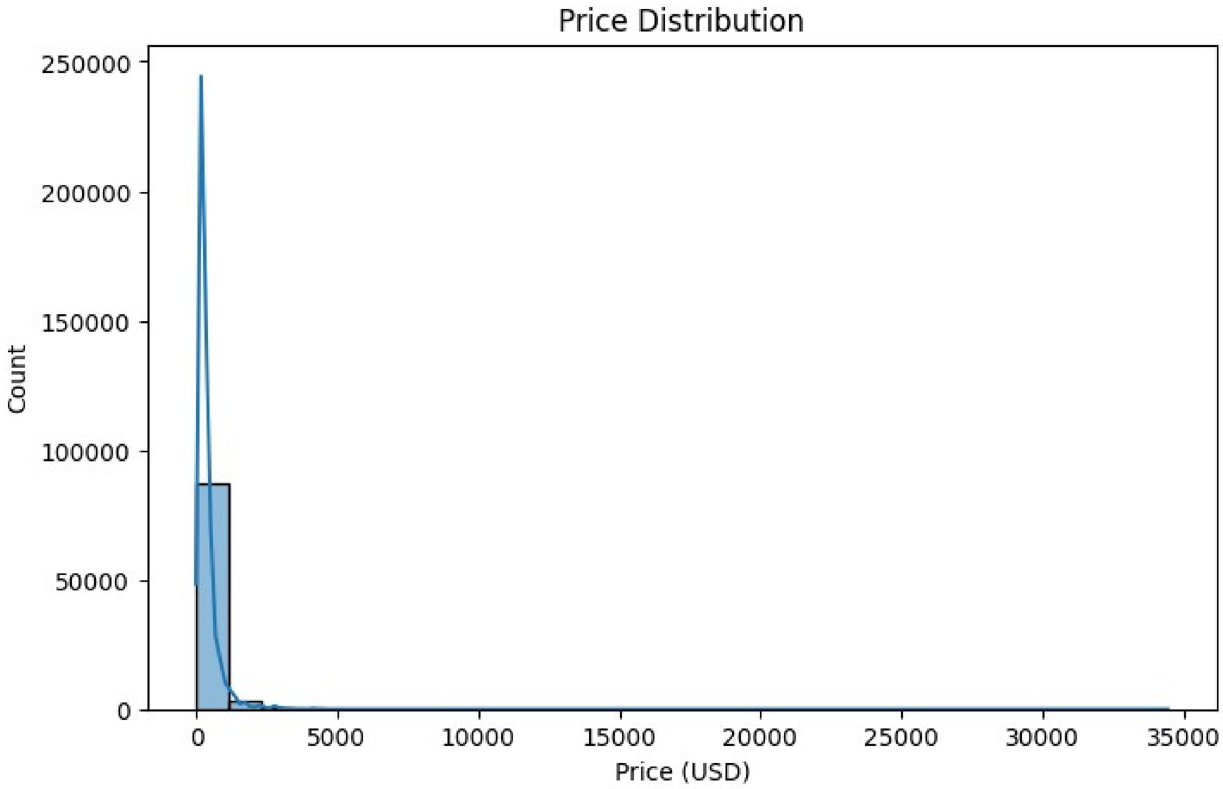
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset (no header in file)
df = pd.read_csv('/kaggle/input/ecommerce-purchase-history-from-
jewelry-store/jewelry.csv', header=None)

# Assign column names
df.columns = [
    'Order Datetime', 'Order ID', 'Product ID', 'Quantity',
    'Category ID', 'Category Alias', 'Brand ID', 'Price USD',
    'User ID', 'Gender', 'Color', 'Material', 'Gemstone'
]

# Convert datetime column
df['Order Datetime'] = pd.to_datetime(df['Order Datetime'],
errors='coerce')

# Check missing values
print("Missing values:\n", df.isnull().sum())

# Basic statistics
print("\nSummary Statistics:\n", df.describe(include='all'))

# Drop rows with missing crucial data
df.dropna(subset=['Price USD', 'Quantity'], inplace=True)

# Visualize Price distribution
plt.figure(figsize=(8,5))
sns.histplot(df['Price USD'], bins=30, kde=True)
plt.title('Price Distribution')
plt.xlabel('Price (USD)')
plt.show()

# Boxplot to detect outliers in Price
plt.figure(figsize=(8,4))
sns.boxplot(x=df['Price USD'])
plt.title('Price USD Outliers')
plt.show()

# Top Categories by count
plt.figure(figsize=(10,5))
df['Category Alias'].value_counts().head(10).plot(kind='bar')
plt.title('Top 10 Product Categories')
plt.xlabel('Category Alias')
plt.ylabel('Count')
plt.show()

# Correlation Heatmap
```

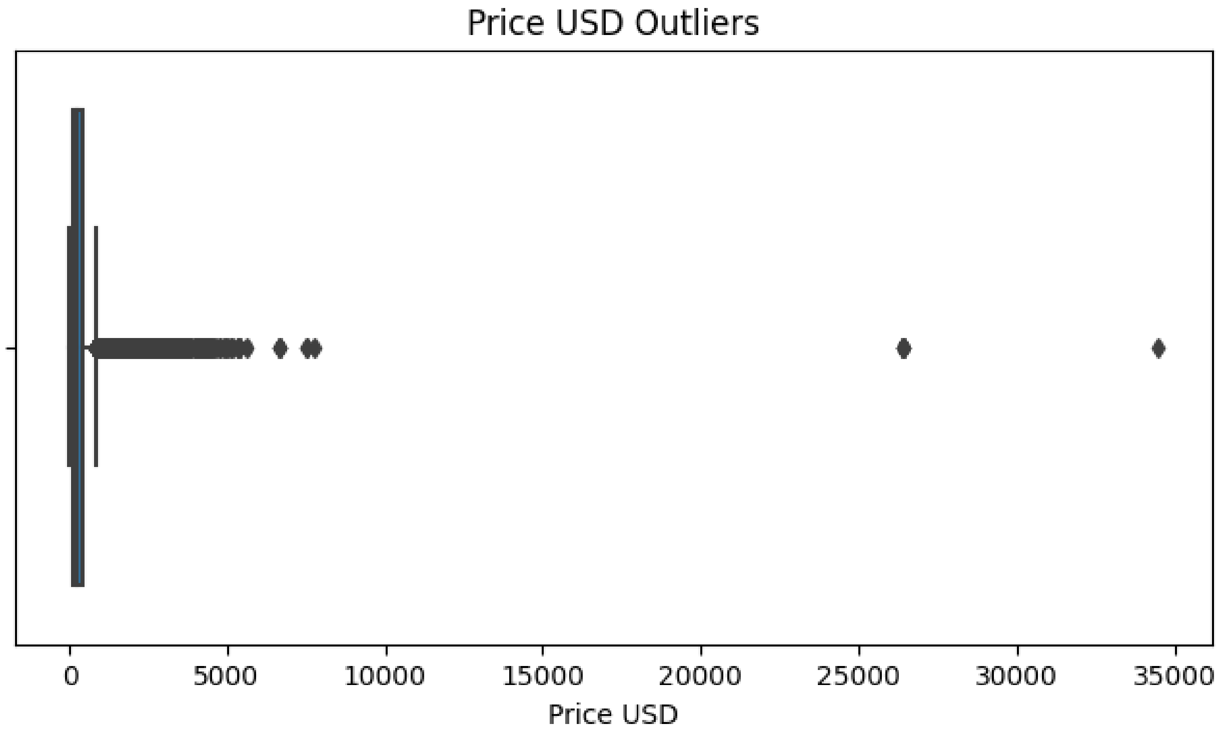
```
num_cols = ['Quantity', 'Brand ID', 'Price USD']
plt.figure(figsize=(6,4))
sns.heatmap(df[num_cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Between Numerical Features')
plt.show()

# Gender distribution
plt.figure(figsize=(6,4))
sns.countplot(x='Gender', data=df)
plt.title('Gender Distribution')
plt.show()

# Color Distribution
plt.figure(figsize=(10,4))
df['Color'].value_counts().head(10).plot(kind='bar')
plt.title('Top 10 Colors')
plt.xlabel('Color')
plt.ylabel('Frequency')
plt.show()
```

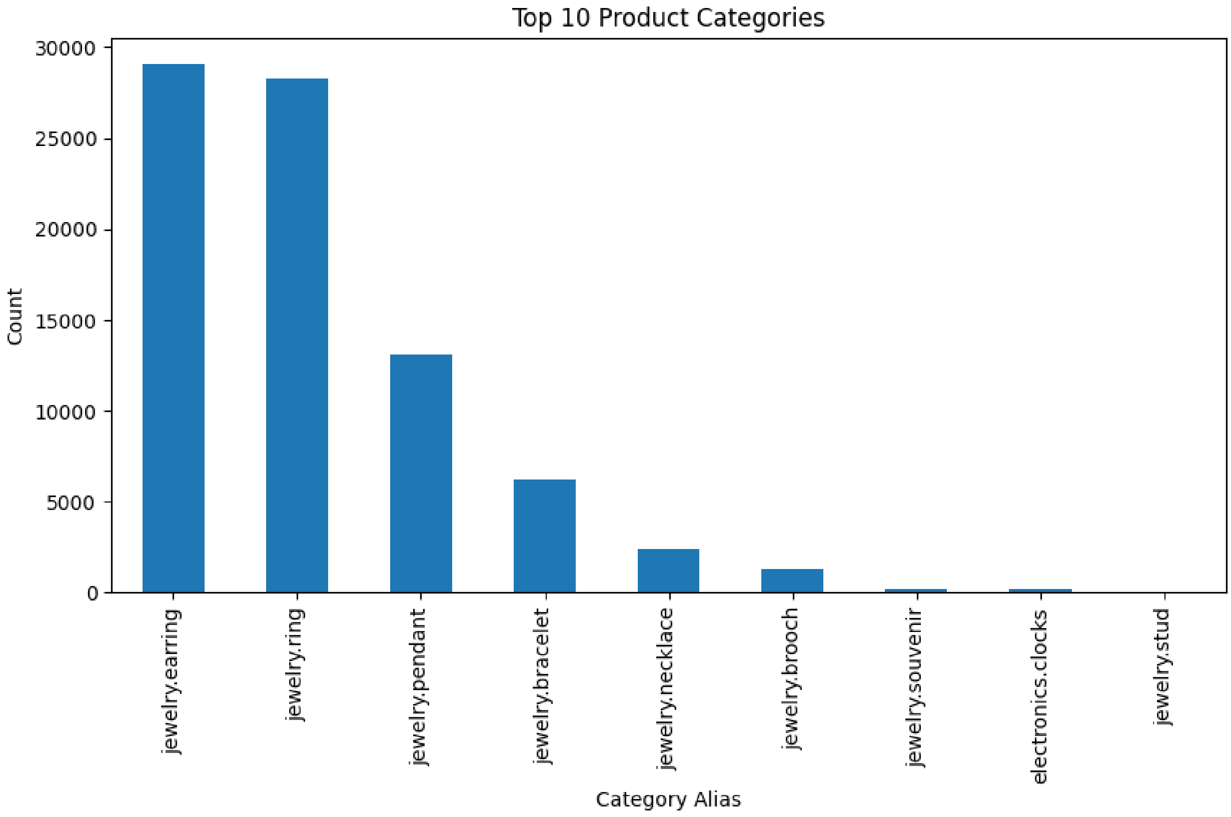
```
Missing values:
 Order Datetime        0
Order ID              0
Product ID            0
Quantity              0
Category ID        5352
Category Alias     9933
Brand ID           4785
Price USD          5352
User ID            5352
Gender            48168
Color              7660
Material           5462
Gemstone          34058
dtype: int64

Summary Statistics:
                              Order Datetime      Order ID    Product
ID  \
count                                  95911  9.591100e+04
9.591100e+04
unique                                   NaN           NaN
NaN
top                                      NaN           NaN
NaN
freq                                     NaN           NaN
NaN
mean    2021-01-12 18:45:26.986456320+00:00  2.485185e+18
1.815970e+18
min               2018-12-01 11:40:29+00:00  1.924719e+18
```

```
1.313551e+18
25%        2020-08-20 06:31:57.500000+00:00  2.379723e+18
1.515966e+18
50%               2021-03-07 17:12:47+00:00  2.524276e+18
1.956664e+18
75%               2021-08-20 09:11:33+00:00  2.644347e+18
1.956664e+18
max               2021-12-01 09:59:07+00:00  2.719022e+18
2.541962e+18
std                                     NaN  1.934825e+17
2.136803e+17

        Quantity   Category ID   Category Alias       Brand ID
Price USD  \
count    95911.0  9.055900e+04            85978  9.112600e+04
90559.000000
unique       NaN           NaN              218           NaN
NaN
top          NaN           NaN  jewelry.earring           NaN
NaN
freq         NaN           NaN            29052           NaN
NaN
mean         1.0  1.805947e+18              NaN  8.890938e+16
362.215217
min          1.0  1.313678e+18              NaN  0.000000e+00
0.990000
25%          1.0  1.806829e+18              NaN  0.000000e+00
145.620000
50%          1.0  1.806829e+18              NaN  1.000000e+00
258.770000
75%          1.0  1.806829e+18              NaN  1.000000e+00
431.370000
max          1.0  1.806829e+18              NaN  1.550613e+18
34448.600000
std          0.0  2.083942e+16              NaN  3.559633e+17
444.155706

             User ID Gender  Color Material Gemstone
count   9.055900e+04  47743  88251    90449    61853
unique           NaN      2      5        3       30
top              NaN      f    red     gold  diamond
freq             NaN  47379  69511    89082    29610
mean    1.512644e+18    NaN    NaN      NaN      NaN
min     1.313554e+18    NaN    NaN      NaN      NaN
25%     1.515916e+18    NaN    NaN      NaN      NaN
50%     1.515916e+18    NaN    NaN      NaN      NaN
75%     1.515916e+18    NaN    NaN      NaN      NaN
max     1.554297e+18    NaN    NaN      NaN      NaN
std     2.374763e+16    NaN    NaN      NaN      NaN
```

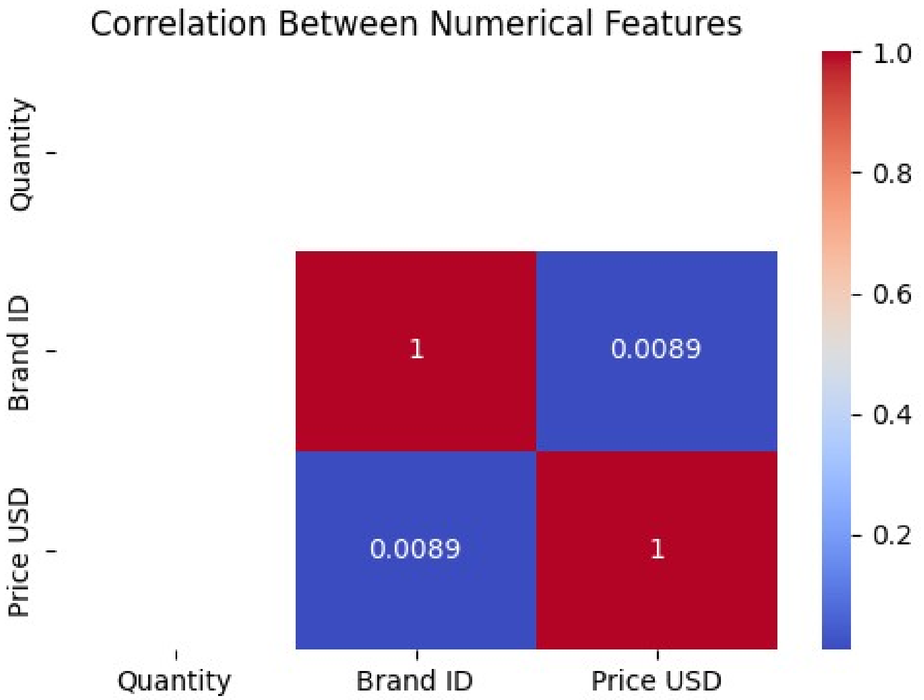
```
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/
format.py:1458: RuntimeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in greater
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

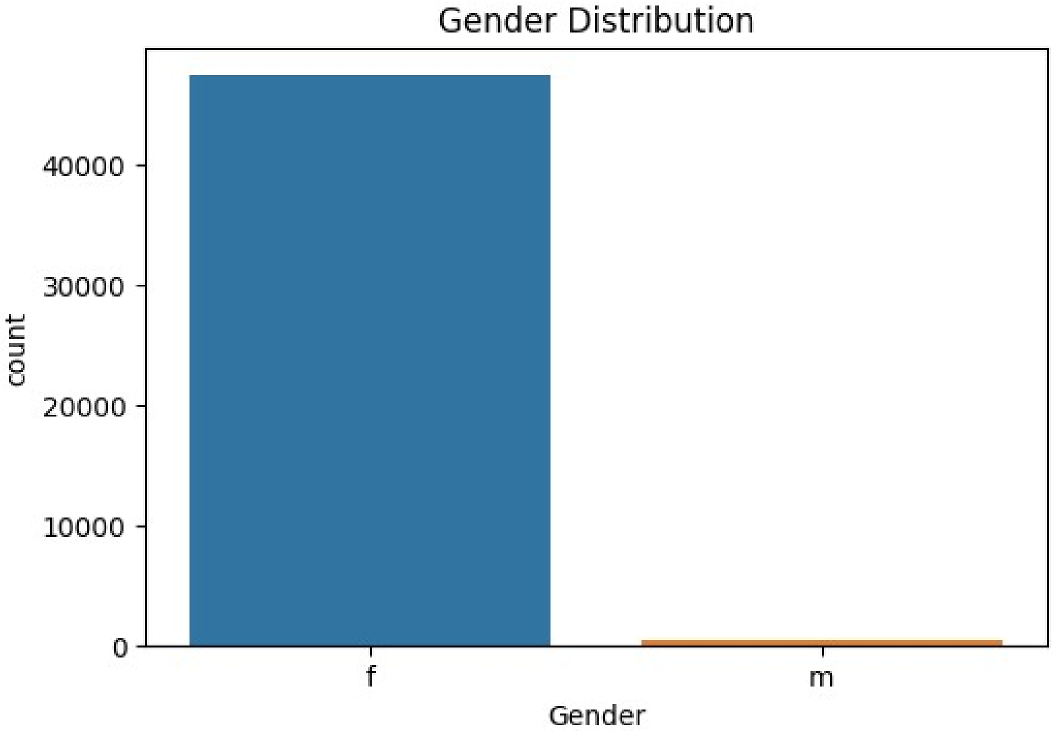


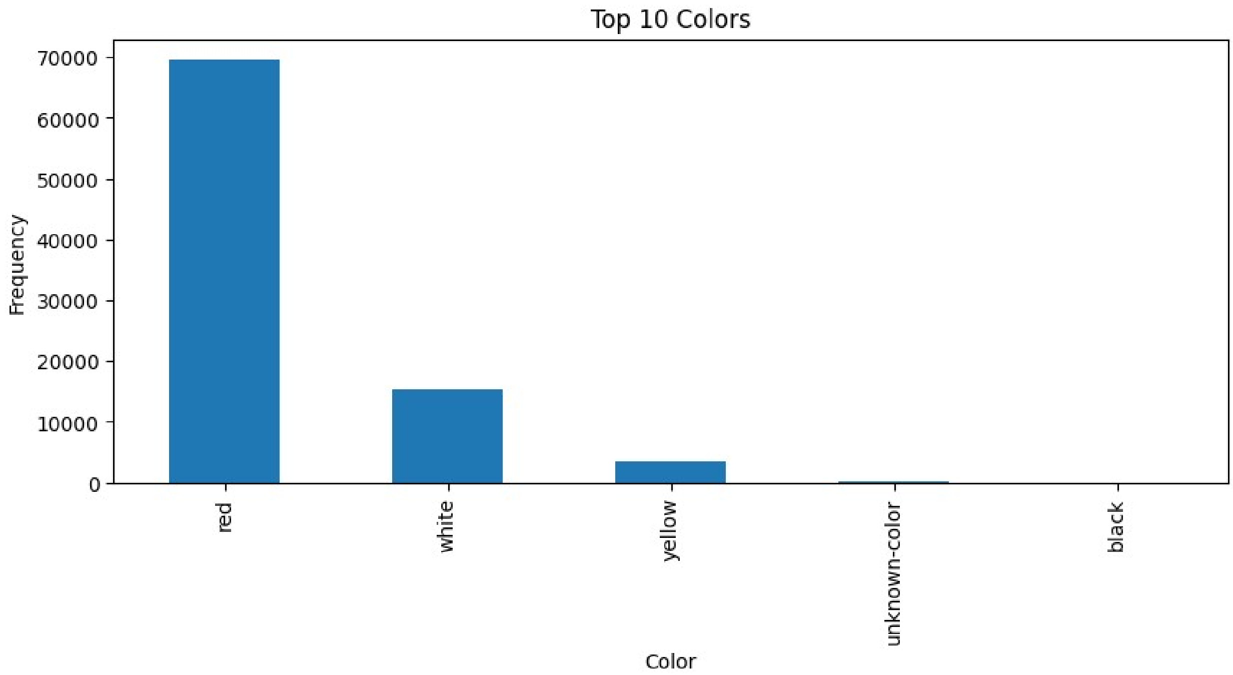

Price USD Outliers
0
5000
10000
15000
20000
25000
30000
35000
Price USD


Top 10 Product Categories
30000
25000
20000
15000
10000
5000
0
Count
jewelry.earring
jewelry.ring
jewelry.pendant
jewelry.bracelet
jewelry.necklace
jewelry.brooch
jewelry.souvenir
electronics.clocks
jewelry.stud
Category Alias

```
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721:
RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
```



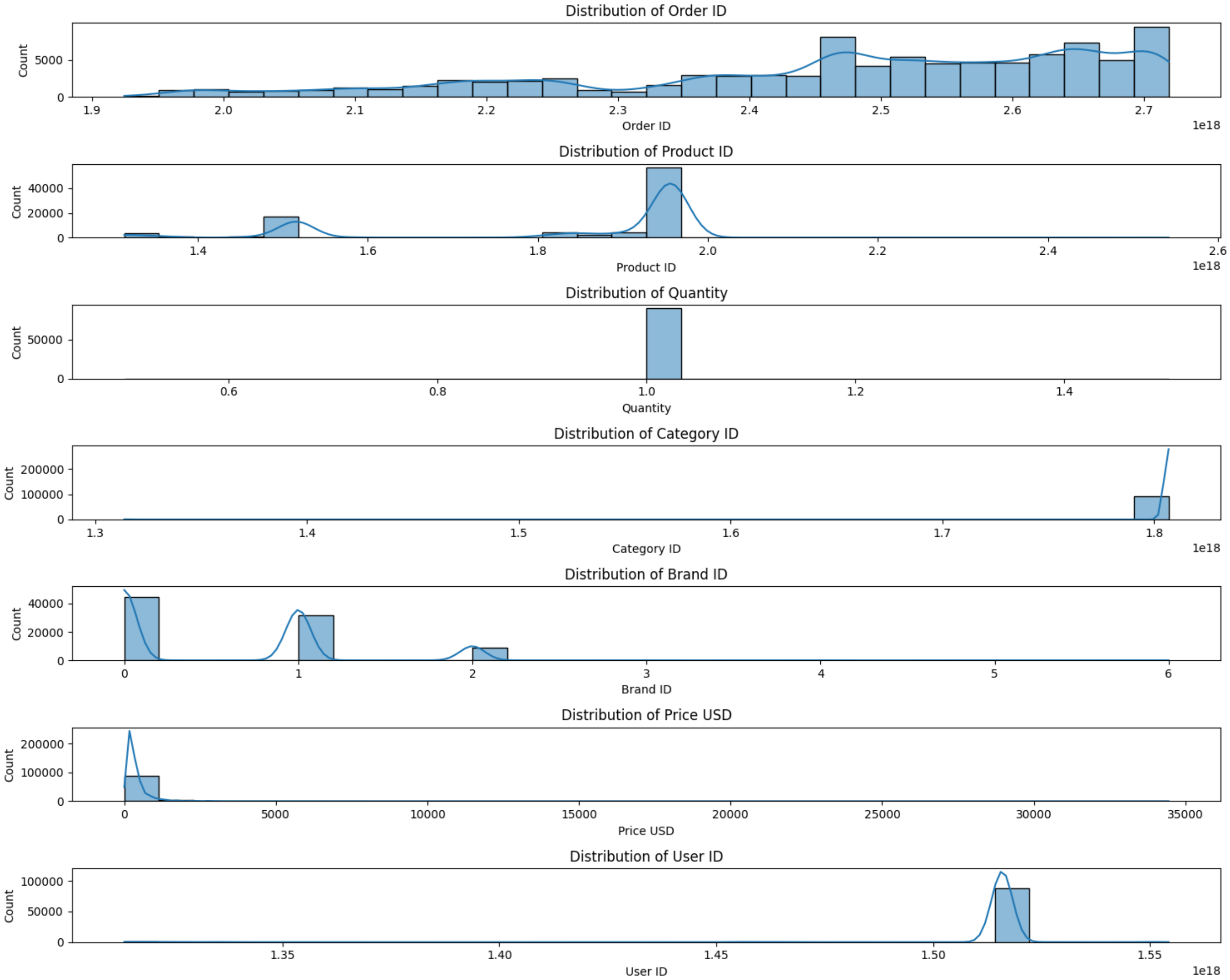




```python
import matplotlib.pyplot as plt
import seaborn as sns

# Select only numeric columns
numeric_cols = df.select_dtypes(include='number').columns

# Plot distribution for each numeric column
plt.figure(figsize=(15, 12))
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(len(numeric_cols), 1, i)
    sns.histplot(df[col], bins=30, kde=True)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.tight_layout()

plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
```

```
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/tmp/ipykernel_36/4198226707.py:14: UserWarning: The figure layout has
changed to tight
  plt.tight_layout()
```


Distribution of Order ID
Count
5000
0
1.9
2.0
2.1
2.2
2.3
2.4
2.5
2.6
2.7
1e18
Order ID
Distribution of Product ID
Count
40000
20000
0
1.4
1.6
1.8
2.0
2.2
2.4
2.6
1e18
Product ID
Distribution of Quantity
Count
50000
0
0.6
0.8
1.0
1.2
1.4
Quantity
Distribution of Category ID
Count
200000
100000
0
1.3
1.4
1.5
1.6
1.7
1.8
1e18
Category ID
Distribution of Brand ID
Count
40000
20000
0
0
1
2
3
4
5
6
Brand ID
Distribution of Price USD
Count
200000
100000
0
0
5000
10000
15000
20000
25000
30000
35000
Price USD
Distribution of User ID
Count
100000
50000
0
1.35
1.40
1.45
1.50
1.55
1e18
User ID