```
import pandas as pd
```

```
df =pd.read_csv("/content/Online Retail.csv")
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	(
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	K
1	536365	71053	WHITE METAL	6	12/1/2010 8:26	3.39	17850.0	K

```
df.tail()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	Custc
292218	562542	22712	CARD DOLLY GIRL	12	8/5/2011 15:40	0.42	1'
292219	562542	22028	PENNY FARTHING BIRTHDAY CARD	12	8/5/2011 15:40	0.42	1'

```
df.describe()
```

	Quantity	UnitPrice	CustomerID
count	292223.000000	292222.000000	210045.000000
mean	9.485225	4.993732	15278.340870
std	203.128975	111.705698	1732.743976
min	-74215.000000	0.000000	12346.000000
25%	1.000000	1.250000	13838.000000
50%	3.000000	2.100000	15143.000000
75%	10.000000	4.130000	16823.000000
max	74215.000000	38970.000000	18287.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 292223 entries, 0 to 292222
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    292223 non-null  object 
 1   StockCode    292223 non-null  object 
 2   Description  291128 non-null  object 
 3   Quantity     292223 non-null  int64  
 4   InvoiceDate  292223 non-null  object 
 5   UnitPrice    292222 non-null  float64
 6   CustomerID   210045 non-null  float64
 7   Country      292222 non-null  object 
dtypes: float64(2), int64(1), object(5)
memory usage: 17.8+ MB
```

```
df.dtypes
```

```
InvoiceNo       object
StockCode       object
Description     object
Quantity         int64
InvoiceDate     object
UnitPrice      float64
CustomerID     float64
Country         object
dtype: object
```

```
df.isnull().sum()
```

```
InvoiceNo          0
StockCode          0
Description     1095
Quantity           0
InvoiceDate        1
UnitPrice          1
CustomerID     82178
Country            1
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.isnull().sum()
```

```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

```
unique_values = df['StockCode'].unique()
```

```
print(unique_values)
```

```
['85123A' '71053' '84406B' ... '23350' '23352' '23351']
```

```
duplicate_rows = df[df.duplicated()]
```

```
print(duplicate_rows)
```

```
       InvoiceNo StockCode                        Description  Quantity  \
517       536409     21866        UNION JACK FLAG LUGGAGE TAG         1   
527       536409     22866      HAND WARMER SCOTTY DOG DESIGN         1   
537       536409     22900    SET 2 TEA TOWELS I LOVE LONDON          1   
539       536409     22111       SCOTTIE DOG HOT WATER BOTTLE         1   
555       536412     22327  ROUND SNACK BOXES SET OF 4 SKULLS         1   
...          ...       ...                                ...       ...   
292127    562540     21162           TOXIC AREA  DOOR HANGER          3   
292134    562540     23254      CHILDRENS CUTLERY DOLLY GIRL          1   
292135    562540     23255    CHILDRENS CUTLERY CIRCUS PARADE         1   
292145    562540     22891               TEA FOR ONE POLKADOT         1   
292180    562540     23073               GEORGIAN TRINKET BOX         1   

            InvoiceDate  UnitPrice  CustomerID         Country  
517     12/1/2010 11:45       1.25     17908.0  United Kingdom  
527     12/1/2010 11:45       2.10     17908.0  United Kingdom  
537     12/1/2010 11:45       2.95     17908.0  United Kingdom  
539     12/1/2010 11:45       4.95     17908.0  United Kingdom  
555     12/1/2010 11:49       2.95     17920.0  United Kingdom  
...                 ...        ...         ...             ...  
292127   8/5/2011 15:38       0.39     17400.0  United Kingdom  
292134   8/5/2011 15:38       4.15     17400.0  United Kingdom  
292135   8/5/2011 15:38       4.15     17400.0  United Kingdom  
292145   8/5/2011 15:38       4.25     17400.0  United Kingdom  
292180   8/5/2011 15:38      12.50     17400.0  United Kingdom  

[2269 rows x 8 columns]
```

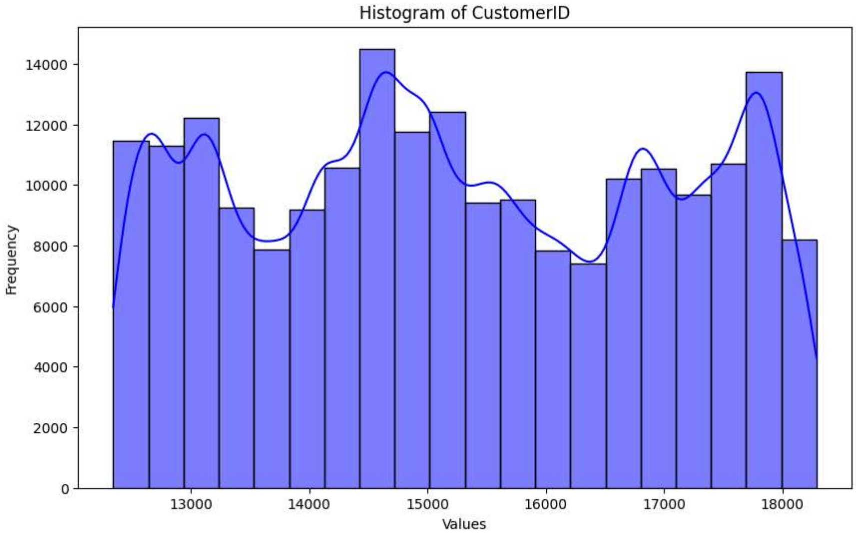
```
num_duplicates = df.duplicated().sum()
```

```
print(num_duplicates)
```

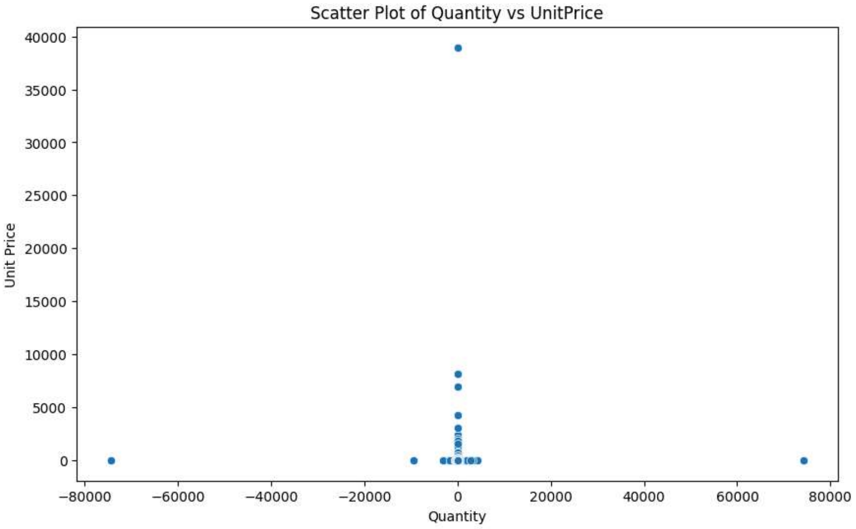
```
2269
```

```
df.drop_duplicates(inplace=True)
```

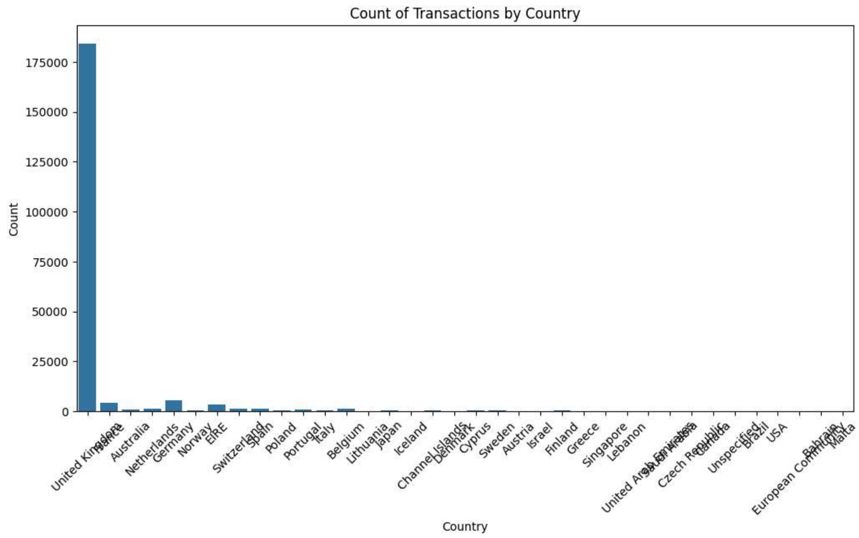
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.histplot(df['CustomerID'], bins=20, kde=True, color='blue')
plt.title('Histogram of CustomerID')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()
```



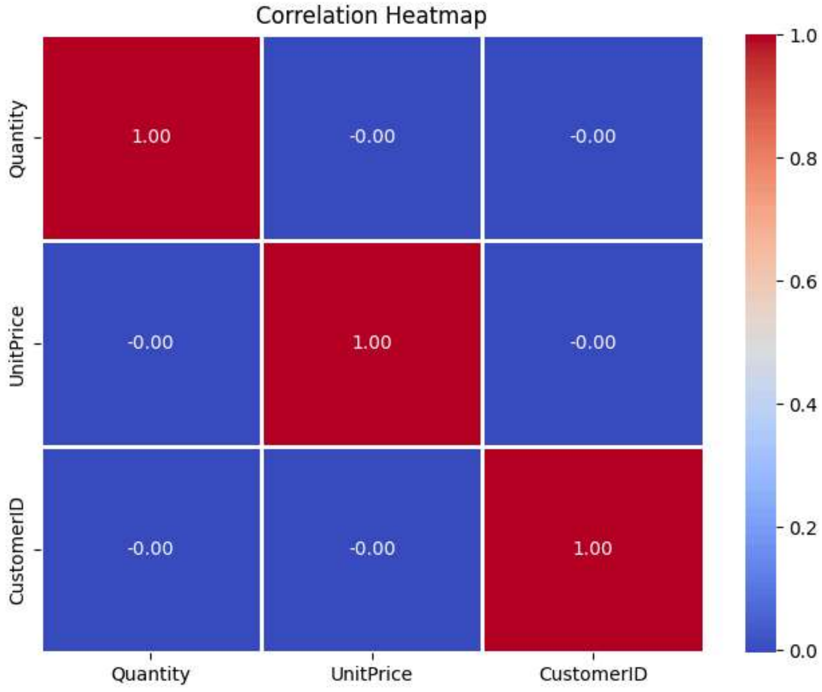
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Quantity', y='UnitPrice', data=df)
plt.title('Scatter Plot of Quantity vs UnitPrice')
plt.xlabel('Quantity')
plt.ylabel('Unit Price')
plt.show()
```



```
plt.figure(figsize=(12, 6))
sns.countplot(x='Country', data=df)
plt.title('Count of Transactions by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.heatmap(df[['Quantity', 'UnitPrice', 'CustomerID']].corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=1)
plt.title('Correlation Heatmap')
plt.show()
```

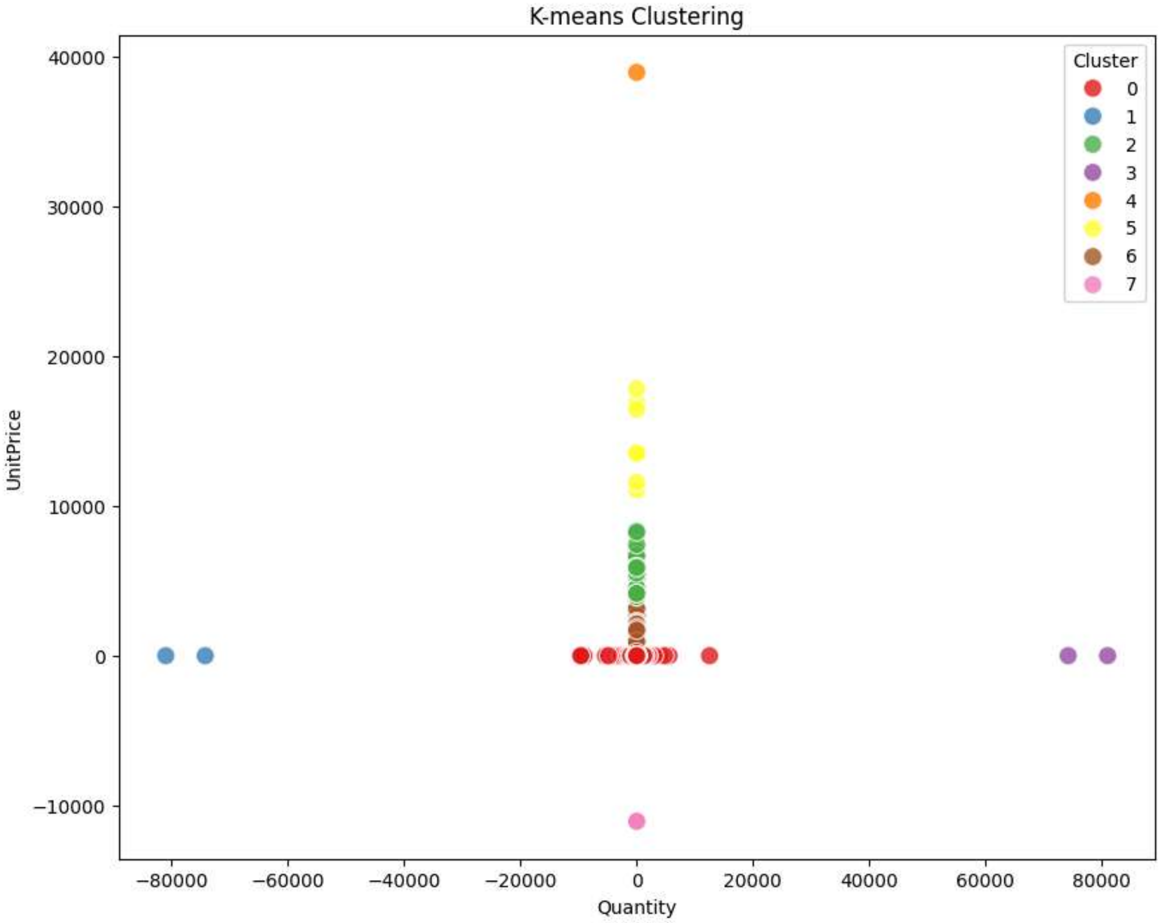


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('/content/Online Retail.csv')

X = df[['Quantity', 'UnitPrice']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
kmeans = KMeans(n_clusters=8, random_state=48)
kmeans.fit(X_scaled)
df['Cluster'] = kmeans.labels_
print("Cluster Centers (original scale):")
print(scaler.inverse_transform(kmeans.cluster_centers_))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
Cluster Centers (original scale):
[[ 9.55605400e+00  3.62849948e+00]
 [-7.76050000e+04  1.56000000e+00]
 [-6.66666667e-01  5.64337576e+03]
 [ 7.76050000e+04  1.56000000e+00]
 [-1.00000000e+00  3.89700000e+04]
 [-5.55555556e-01  1.42139478e+04]
 [ 7.00564972e-01  1.14292904e+03]
```

```
plt.figure(figsize=(10, 8))
sns.scatterplot(x='Quantity', y='UnitPrice', hue='Cluster', data=df, palette='Set1', s=100, alpha=0.8)
plt.title('K-means Clustering')
plt.xlabel('Quantity')
plt.ylabel('UnitPrice')
plt.legend(title='Cluster')
plt.show()
```


K-means Clustering
Cluster
● 0
● 1
● 2
● 3
● 4
● 5
● 6
● 7
UnitPrice
Quantity