Duplication & Typecasting

Instructions:

Please share your answers filled inline in the word document. Submit Python code and R code files wherever applicable.

Please ensure you update all the details:

Name: ULLI VENKATA SAI KUMAR

Batch Id: 04072024HYD10AM

Topic: Preliminaries for Data Analysis

Problem statement:

Data collected may have duplicate entries, that might be because the data collected were not at regular intervals or for any other reason. Building a proper solution on such data will be a tough ask. The common techniques are either removing duplicates completely or substituting those values with logical data. There are various techniques to treat these types of problems.

Q1. For the given dataset perform the type casting (convert the datatypes, ex. float to int)

```
import pandas as pd

df = pd.read_csv(r"Online Retail.csv", encoding='unicode_escape')

df['UnitPrice'] = df['UnitPrice'].astype(float)

print(df.dtypes)
```

```
In [3]: print(df.dtypes)
InvoiceNo        object
StockCode        object
Description      object
Quantity          int64
InvoiceDate      object
UnitPrice       float64
CustomerID      float64
Country          object
dtype: object
```

Q2. Check for duplicate values, and handle the duplicate values (ex. drop)

```
import pandas as pd

df = pd.read_csv(r"Online Retail.csv",

            encoding='unicode_escape')

duplicates = df.duplicated()
```

duplicate_count = duplicates.sum()

df_cleaned = df.drop_duplicates()

duplicate_count, df_cleaned

Name	Type	Size	Value
df	DataFrame	(541909, 8)	Column names: InvoiceNo, StockCode, Description, Quantity, InvoiceDate ...
df_cleaned	DataFrame	(536641, 8)	Column names: InvoiceNo, StockCode, Description, Quantity, InvoiceDate ...
duplicate_count	int64	1	5268
duplicates	Series	(541909,)	Series object of pandas.core.series module

Help | Variable Explorer | Plots | Files

Console 5/A

```
(5268,
        InvoiceNo StockCode  ... CustomerID         Country
0          536365    85123A  ...    17850.0  United Kingdom
1          536365     71053  ...    17850.0  United Kingdom
2          536365    84406B  ...    17850.0  United Kingdom
3          536365    84029G  ...    17850.0  United Kingdom
4          536365    84029E  ...    17850.0  United Kingdom
...           ...       ...  ...        ...             ...
541904     581587     22613  ...    12680.0          France
541905     581587     22899  ...    12680.0          France
541906     581587     23254  ...    12680.0          France
541907     581587     23255  ...    12680.0          France
541908     581587     22138  ...    12680.0          France

[536641 rows x 8 columns])
```

Q3. Do the data analysis (EDA)?

Such as histogram, boxplot, scatterplot, etc.

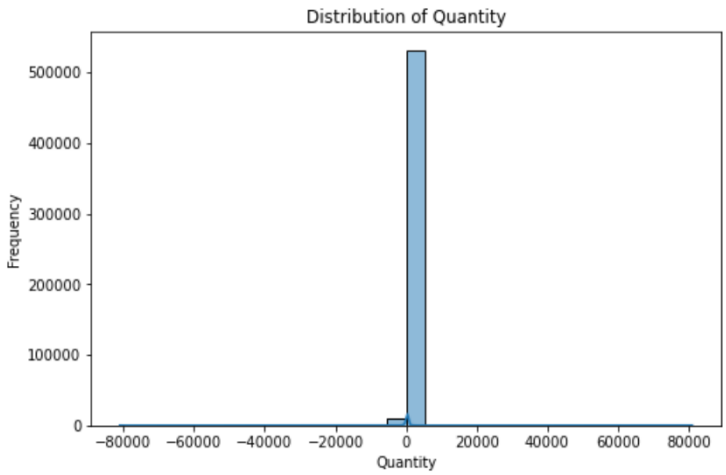
import pandas as pd

import matplotlib.pyplot as plt

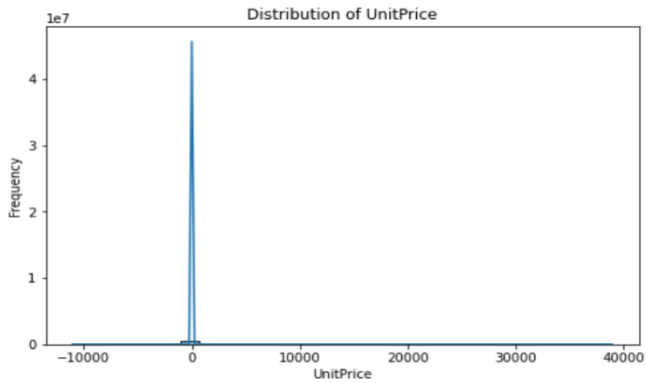
import seaborn as sns

df = pd.read_csv(r"Online Retail.csv",encoding='unicode_escape')

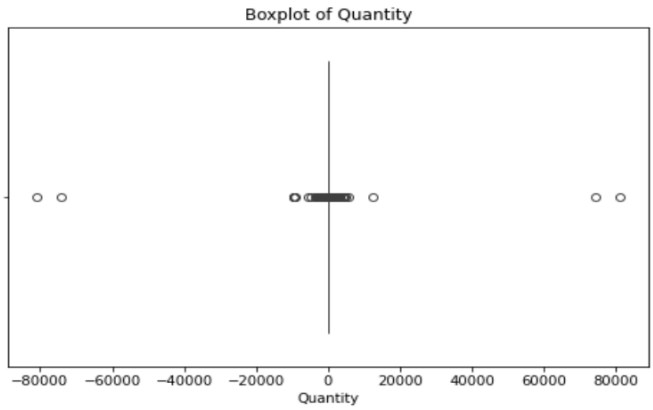
```
# Histogram for Quantity
plt.figure(figsize=(8, 5))
sns.histplot(df['Quantity'], bins=30, kde=True)
plt.title('Distribution of Quantity')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.show()
```



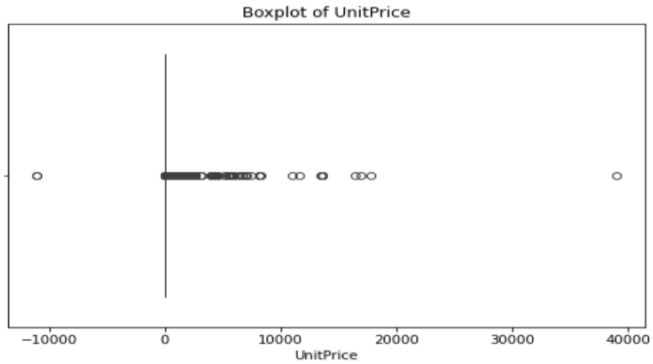

```
# Histogram for UnitPrice
plt.figure(figsize=(8, 5))
sns.histplot(df['UnitPrice'], bins=30, kde=True)
plt.title('Distribution of UnitPrice')
plt.xlabel('UnitPrice')
plt.ylabel('Frequency')
plt.show()
```



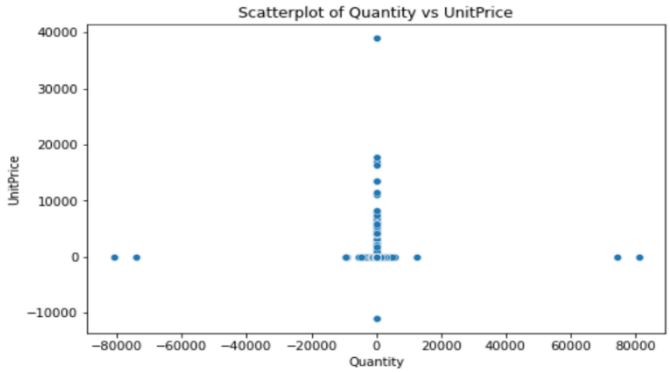

```
# Boxplot for Quantity
plt.figure(figsize=(8, 5))
sns.boxplot(x=df['Quantity'])
plt.title('Boxplot of Quantity')
plt.show()
```



```
# Boxplot for UnitPrice
plt.figure(figsize=(8, 5))
sns.boxplot(x=df['UnitPrice'])
plt.title('Boxplot of UnitPrice')
plt.show()
```




```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Quantity', y='UnitPrice',
data=df)
plt.title('Scatterplot of Quantity vs UnitPrice')
plt.xlabel('Quantity')
plt.ylabel('UnitPrice')
plt.show()
```




```
# Convert InvoiceDate to datetime
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
# Plotting the number of orders over time
plt.figure(figsize=(10, 6))
df['InvoiceDate'].groupby(df['InvoiceDate'].
dt.date).count().plot()
plt.title('Number of Orders Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Orders')
plt.show()
```