In [18]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [20]:
```
df=pd.read_excel(r"C:\Users\deepa\OneDrive\Desktop\Online_Retail_Data_Set.xlsx")
```

In [21]:
```
df.head()
```

Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

In [46]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    541909 non-null  object 
 1   StockCode    541909 non-null  object 
 2   Description  540455 non-null  object 
 3   Quantity     541909 non-null  int64  
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object 
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

In [55]:
```
df.isna().sum()
```

Out[55]:

```
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID    135080
Country            0
dtype: int64
```

In [56]:
```
df_neg_qmdf[df["Quantity"]<1]
```

In [77]:
```
df_pos_qmdf[df["Quantity"]>0]
```

In [81]:
```
df_pos_q.shape
```

Out[81]:
```
(531285, 8)
```

In [59]:
```
df_neg_q.shape
```

Out[59]:
```
(10624, 8)
```

In [109]:
```
df_neg_pmdf_pos_q[df["UnitPrice"]<0]
```

```
C:\Users\deepa\AppData\Local\Temp\ipykernel_14880\2430195404.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  df_neg_p=df_pos_q[df["UnitPrice"]<0]
```

In [111]:
```
df_neg_p.shape
```

Out[111]:
```
(2, 8)
```

In [122]:
```
df_pos_pmdf_pos_q[df_pos_q["UnitPrice"]>0]
df_pos_p.shape
```

Out[122]:
```
(530104, 8)
```

In [133]:
```
df=df_pos_p.copy()
```

In [146]:
```
df.shape
```

Out[146]:
```
(530104, 8)
```

In [155]:
```
for i,j in zip(df.columns,df.isna().sum()):
    print(i,j,"-",round(j/df.shape[0],4)*100,"%")
```

```
InvoiceNo 0 - 0.0 %
StockCode 0 - 0.0 %
Description 0 - 0.0 %
Quantity 0 - 0.0 %
InvoiceDate 0 - 0.0 %
UnitPrice 0 - 0.0 %
CustomerID 132220 - 24.94 %
Country 0 - 0.0 %
```

In [166]:
```
# Missing value imputation

df["Description"].fillna(df["Description"].mode()[0],inplace=True)
```

```
C:\Users\deepa\AppData\Local\Temp\ipykernel_14880\2867247727.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df["Description"].fillna(df["Description"].mode()[0],inplace=True)
```

In [157]:
```
for i,j in zip(df.columns,df.isna().sum()):
    print(i,j,"-",round(j/df.shape[0],4)*100,"%")
```

```
InvoiceNo 0 - 0.0 %
StockCode 0 - 0.0 %
Description 0 - 0.0 %
Quantity 0 - 0.0 %
InvoiceDate 0 - 0.0 %
UnitPrice 0 - 0.0 %
CustomerID 132220 - 24.94 %
Country 0 - 0.0 %
```

In [158]:
```
df["CustomerID"].nunique()
```

Out[158]:
```
4338
```

In [159]:
```
# Missing value imputation by bfill or ffill or mode in Customer ID column

df["CustomerID"]=df["CustomerID"].fillna(method="ffill")

#  df["CustomerID"]=df["CustomerID"].fillna(method="bfill")
```

```
C:\Users\deepa\AppData\Local\Temp\ipykernel_14880\2283629695.py:3: FutureWarning: Series.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  df["CustomerID"]=df["CustomerID"].fillna(method="ffill")
```

In [220]:
```
df["CustomerID"].isna().sum()
```

Out[220]:
```
0
```

In [221]:
```
for i,j in zip(df.columns,df.isna().sum()):
    print(i,j,"-",round(j/df.shape[0],4)*100,"%")
```

```
InvoiceNo 0 - 0.0 %
StockCode 0 - 0.0 %
Description 0 - 0.0 %
Quantity 0 - 0.0 %
InvoiceDate 0 - 0.0 %
UnitPrice 0 - 0.0 %
CustomerID 0 - 0.0 %
Country 0 - 0.0 %
```

In [222]:
```
# Basic summary statistics

df.describe()
```

Out[222]:

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	530104.000000	530104	530104.000000	530104.000000
mean	10.542037	2011-07-04 20:16:05.225087744	3.907625	15287.610047
min	1.000000	2010-12-01 08:26:00	0.001000	12346.000000
25%	1.000000	2011-03-28 12:22:00	1.250000	13804.000000
50%	3.000000	2011-07-20 12:58:00	2.080000	15179.000000
75%	10.000000	2011-10-19 12:39:00	4.130000	16813.000000
max	80995.000000	2011-12-09 12:50:00	13541.330000	18287.000000
std	155.524124	NaN	35.915681	1735.660857

In [223]:
```
# creating new columns as total_amt

df["total_amt"]=df["Quantity"]*df["UnitPrice"]
```

In [224]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 530104 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    530104 non-null  object 
 1   StockCode    530104 non-null  object 
 2   Description  530104 non-null  object 
 3   Quantity     530104 non-null  int64  
 4   InvoiceDate  530104 non-null  datetime64[ns]
 5   UnitPrice    530104 non-null  float64
 6   CustomerID   530104 non-null  float64
 7   Country      530104 non-null  object 
 8   total_amt    530104 non-null  float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 40.4+ MB
```

In [225]:
```
df.describe()
```

Out[225]:

	Quantity	InvoiceDate	UnitPrice	CustomerID	total_amt
count	530104.000000	530104	530104.000000	530104.000000	530104.000000
mean	10.542037	2011-07-04 20:16:05.225087744	3.907625	15287.610047	20.121871
min	1.000000	2010-12-01 08:26:00	0.001000	12346.000000	0.001000
25%	1.000000	2011-03-28 12:22:00	1.250000	13804.000000	3.750000
50%	3.000000	2011-07-20 12:58:00	2.080000	15179.000000	9.900000
75%	10.000000	2011-10-19 12:39:00	4.130000	16813.000000	17.700000
max	80995.000000	2011-12-09 12:50:00	13541.330000	18287.000000	168469.600000
std	155.524124	NaN	35.915681	1735.660857	270.356743

In [226]:
```
df["InvoiceDate"]=df["InvoiceDate"].astype("str")
```

In [227]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 530104 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    530104 non-null  object 
 1   StockCode    530104 non-null  object 
 2   Description  530104 non-null  object 
 3   Quantity     530104 non-null  int64  
 4   InvoiceDate  530104 non-null  object 
 5   UnitPrice    530104 non-null  float64
 6   CustomerID   530104 non-null  float64
 7   Country      530104 non-null  object 
 8   total_amt    530104 non-null  float64
dtypes: float64(3), int64(1), object(5)
memory usage: 40.4+ MB
```

In [228]:
```
df["Date"]=df["InvoiceDate"].str.split(" ")
```

In [229]:
```
#df.drop(columns="Year",axis=1,inplace=True)
```

In [230]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 530104 entries, 0 to 541908
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    530104 non-null  object 
 1   StockCode    530104 non-null  object 
 2   Description  530104 non-null  object 
 3   Quantity     530104 non-null  int64  
 4   InvoiceDate  530104 non-null  object 
 5   UnitPrice    530104 non-null  float64
 6   CustomerID   530104 non-null  float64
 7   Country      530104 non-null  object 
 8   total_amt    530104 non-null  float64
 9   Date         530104 non-null  object 
dtypes: float64(3), int64(1), object(6)
memory usage: 44.5+ MB
```

In [231]:
```
df["Date"]=df["Date"][0][0]
```

In [232]:
```
df["Year"]=df["Date"].str.split("-")[0][0]
```

In [233]:
```
df["Month"]=df["Date"].str.split("-")[0][1]
```

In [234]:
```
df["Year"]=df["Year"].astype("int")
df["Month"]=df["Month"].astype("int")
```

In [235]:
```
df["Year"].isna().sum()
```

Out[235]:
```
0
```

In [261]:
```
df["Month"].isna().sum()
```

Out[261]:
```
0
```

In [237]:
```
df["Date"]=pd.to_datetime(df["Date"])
```

In [238]:
```
df.describe(include="O")
```

Out[238]:

	InvoiceNo	StockCode	Description	InvoiceDate	Country
count	530104	530104	530104	530104	530104
unique	19960	3922	4026	18499	38
top	573585	85123A	WHITE HANGING HEART T-LIGHT HOLDER	2011-10-31 14:41:00	United Kingdom
freq	1114	2265	2323	1114	485123

In [239]:
```
df.describe()
```

Out[239]:

	Quantity	UnitPrice	CustomerID	total_amt	Date	Year	Month
count	530104.000000	530104.000000	530104.000000	530104.000000	530104	530104.0	530104.0
mean	10.542037	3.907625	15287.610047	20.121871	2010-12-01 00:00:00	2010.0	12.0
min	1.000000	0.001000	12346.000000	0.001000	2010-12-01 00:00:00	2010.0	12.0
25%	1.000000	1.250000	13804.000000	3.750000	2010-12-01 00:00:00	2010.0	12.0
50%	3.000000	2.080000	15179.000000	9.900000	2010-12-01 00:00:00	2010.0	12.0
75%	10.000000	4.130000	16813.000000	17.700000	2010-12-01 00:00:00	2010.0	12.0
max	80995.000000	13541.330000	18287.000000	168469.600000	2010-12-01 00:00:00	2010.0	12.0
std	155.524124	35.915681	1735.660857	270.356743	NaN	0.0	0.0

In [240]:
```
df.to_csv(r"C:\Users\deepa\OneDrive\Desktop\Online_Retail_Data_Set1.csv",index=False)
```

In [ ]:
```

```

In [ ]:
```

```