In [1]:

```
import numpy as np
import pandas as pd
# ignore warnings
import warnings
warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt

import seaborn as sns

import os
os.getcwd
# the location of current working directory (CWD).
```

Out[1]:

```
<function nt.getcwd()>
```

In [2]:
```
df = pd.read_excel(r'D:\Madhav\Project\Data visualization - Empowering the Busi
df
```

Out[2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

541909 rows × 8 columns

In [3]:
```
df.shape
```

Out[3]:
```
(541909, 8)
```

In [4]:
```
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 [5]:
```
df.describe()
```

Out[5]:

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

In [6]:
```
df.isna().sum()
```

Out[6]:
```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

In [7]:
```
df.CustomerID.nunique()
```

Out[7]:
```
4372
```

In [8]:

```
# Calculate the value counts for the 'CustomerID' column
df['CustomerID'].value_counts()
```

Out[8]:

```
17841.0    7983
14911.0    5903
14096.0    5128
12748.0    4642
14606.0    2782
           ...
15070.0       1
15753.0       1
17065.0       1
16881.0       1
16995.0       1
Name: CustomerID, Length: 4372, dtype: int64
```

In [9]:

```
#grouped_data = df.groupby('CustomerID').agg({'Country':'sum'}).reset_index()
#grouped_data
```

In [10]:

```
df.Country.nunique()
```

Out[10]:

```
38
```

In [11]:

```
df.Country.unique()
```

Out[11]:

```
array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',
       'Norway', 'EIRE', 'Switzerland', 'Spain', 'Poland', 'Portugal',
       'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',
       'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',
       'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',
       'Lebanon', 'United Arab Emirates', 'Saudi Arabia',
       'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',
       'European Community', 'Malta', 'RSA'], dtype=object)
```

In [12]:
```
df['CustomerID']=df['CustomerID'].fillna(np.mean(pd.to_numeric(df['CustomerID']
df
```

Out[12]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

541909 rows × 8 columns

In [13]:
```
df['Description']=df['Description'].fillna(df['Description'].mode()[0])
df
```

Out[13]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

541909 rows × 8 columns

In [14]:

```
df.isna().sum()
```

Out[14]:

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

In [15]:
```
df
```

Out[15]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

541909 rows × 8 columns

Create a check that the quantity should not be below 1 unit

Create a check that the Unit price should not be below

In [16]:
```
df.InvoiceNo.nunique()
```

Out[16]:
```
25900
```

In [17]:
```
df.Quantity.nunique()
```

Out[17]:
```
722
```

In [18]:
```
df.UnitPrice.nunique()
```

Out[18]:
```
1630
```

In [19]:
```
#it shows the count that how many no.of values are less than 1 in quantity
# Create a boolean mask to filter rows where 'unit' is less than 1
mask = df['Quantity'] < 1

# Use the mask to select the rows where 'unit' is less than 1
result = df[mask]

# Display the result
print(result)
```

```
       InvoiceNo StockCode                         Description  Quantity  \
141      C536379         D                            Discount        -1   
154      C536383    35004C     SET OF 3 COLOURED  FLYING DUCKS        -1   
235      C536391     22556      PLASTERS IN TIN CIRCUS PARADE        -12   
236      C536391     21984    PACK OF 12 PINK PAISLEY TISSUES        -24   
237      C536391     21983    PACK OF 12 BLUE PAISLEY TISSUES        -24   
...          ...       ...                                 ...       ...   
540449   C581490     23144     ZINC T-LIGHT HOLDER STARS SMALL       -11   
541541   C581499         M                              Manual        -1   
541715   C581568     21258          VICTORIAN SEWING BOX LARGE        -5   
541716   C581569     84978    HANGING HEART JAR T-LIGHT HOLDER        -1   
541717   C581569     20979       36 PENCILS TUBE RED RETROSPOT        -5   

               InvoiceDate  UnitPrice  CustomerID         Country  
141    2010-12-01 09:41:00      27.50     14527.0  United Kingdom  
154    2010-12-01 09:49:00       4.65     15311.0  United Kingdom  
235    2010-12-01 10:24:00       1.65     17548.0  United Kingdom  
236    2010-12-01 10:24:00       0.29     17548.0  United Kingdom  
237    2010-12-01 10:24:00       0.29     17548.0  United Kingdom  
...                    ...        ...         ...             ...  
540449 2011-12-09 09:57:00       0.83     14397.0  United Kingdom  
541541 2011-12-09 10:28:00     224.69     15498.0  United Kingdom  
541715 2011-12-09 11:57:00      10.95     15311.0  United Kingdom  
541716 2011-12-09 11:58:00       1.25     17315.0  United Kingdom  
541717 2011-12-09 11:58:00       1.25     17315.0  United Kingdom  

[10624 rows x 8 columns]
```

In [20]:

```
#it shows the whole data set with out the 'quantity' value is lessthan 1
# Create a boolean mask to filter rows where 'quantity' is greater than or equa
mask = df['Quantity'] >= 1

# Use the mask to select the rows where 'quantity' is greater than or equal to
filtered_df = df[mask]

# Display the filtered DataFrame
filtered_df
```

Out[20]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531285 rows × 8 columns

In [21]:
```
filtered_df
```

Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531285 rows × 8 columns

In [22]:

```
#it shows the count that how many no.of values are less than 0 in UnitPrice
# Create a boolean mask to filter rows where 'unitprice' is less than 1
mask = filtered_df['UnitPrice'] < 0

# Use the mask to select the rows where 'unitprice' is less than 1
result = filtered_df[mask]

# Display the result
result
```

Out[22]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Count
299983	A563186	B	Adjust bad debt	1	2011-08-12 14:51:00	-11062.06	15287.69057	Unite Kingdo
299984	A563187	B	Adjust bad debt	1	2011-08-12 14:52:00	-11062.06	15287.69057	Unite Kingdo

In [23]:

```
#it shows the whole data set with out the 'UnitPrice' value is lessthan 0
# Create a boolean mask to filter rows where 'UnitPrice' is greater than or equ
mask = filtered_df['UnitPrice'] >= 0

# Use the mask to select the rows where 'UnitPrice' is greater than or equal to
finaldata = filtered_df[mask]

# Display the filtered DataFrame
finaldata
```

Out[23]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 8 columns

In [24]:
```
finaldata
```

Out[24]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 8 columns

In [25]:

```
finaldata.isna().sum()
```

Out[25]:

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

In [26]:

```
mask = finaldata['UnitPrice'] >= 0
result = finaldata[mask]
result
```

Out[26]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 8 columns

In [27]:

```
mask = finaldata['Quantity'] >= 0
result = finaldata[mask]
result
```

Out[27]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 8 columns

In [28]:

```
finaldata.shape
```

Out[28]:

```
(531283, 8)
```

In [29]:
```
#monthly_revenue = df_2011.groupby(df_2011['InvoiceDate'].dt.month)['Revenue'].
```

In [30]:
```

finaldata['Revenue'] = finaldata['Quantity'] * finaldata['UnitPrice']
```

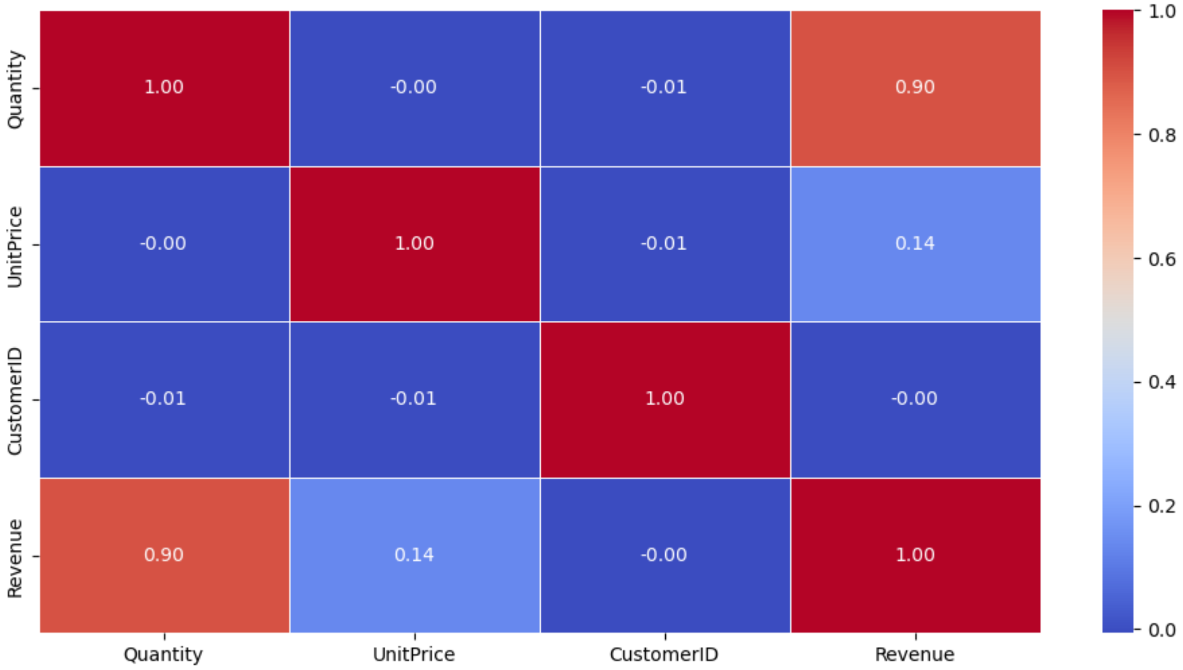
In [31]:
```
finaldata
```

Out[31]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 9 columns

In [32]:
```
correlation_data = finaldata.corr()
plt.figure(figsize=(12,6))
sns.heatmap(correlation_data, annot = True, cmap = 'coolwarm',fmt ='.2f', linew
plt.show()
```

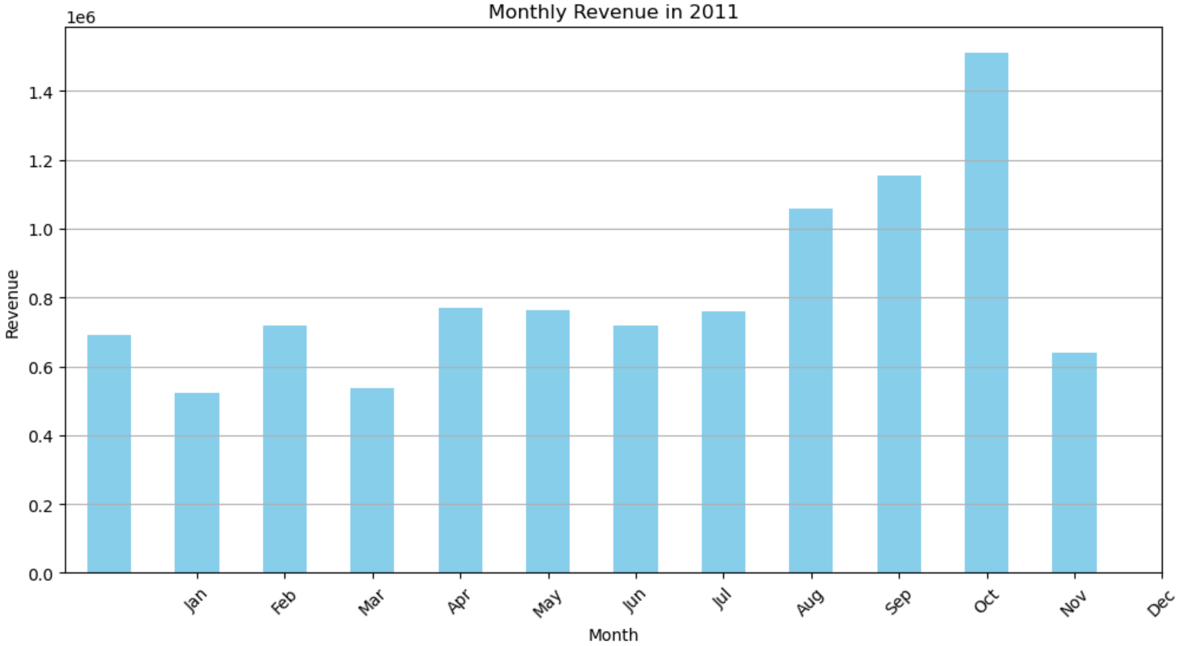


In [33]:
```
df_2011 = finaldata[finaldata['InvoiceDate'].dt.year == 2011]

# Aggregate data by month and calculate total revenue
monthly_revenue = df_2011.groupby(df_2011['InvoiceDate'].dt.month)['Revenue'].s

# Data Visualization (Bar Graph)
plt.figure(figsize=(12, 6))
monthly_revenue.plot(kind='bar', color='skyblue')
plt.title('Monthly Revenue in 2011')
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug
plt.grid(axis='y')

# Show the plot
plt.show()
```



In [34]:
```
# Filter data to exclude the United Kingdom
finaldata_filtered = finaldata[finaldata['Country'] != 'United Kingdom']

# Group data by country and calculate total revenue and total quantity sold
country_stats = finaldata_filtered.groupby('Country').agg({'Revenue': 'sum', 'Q
```

In [35]:

```
country_stats
```

Out[35]:

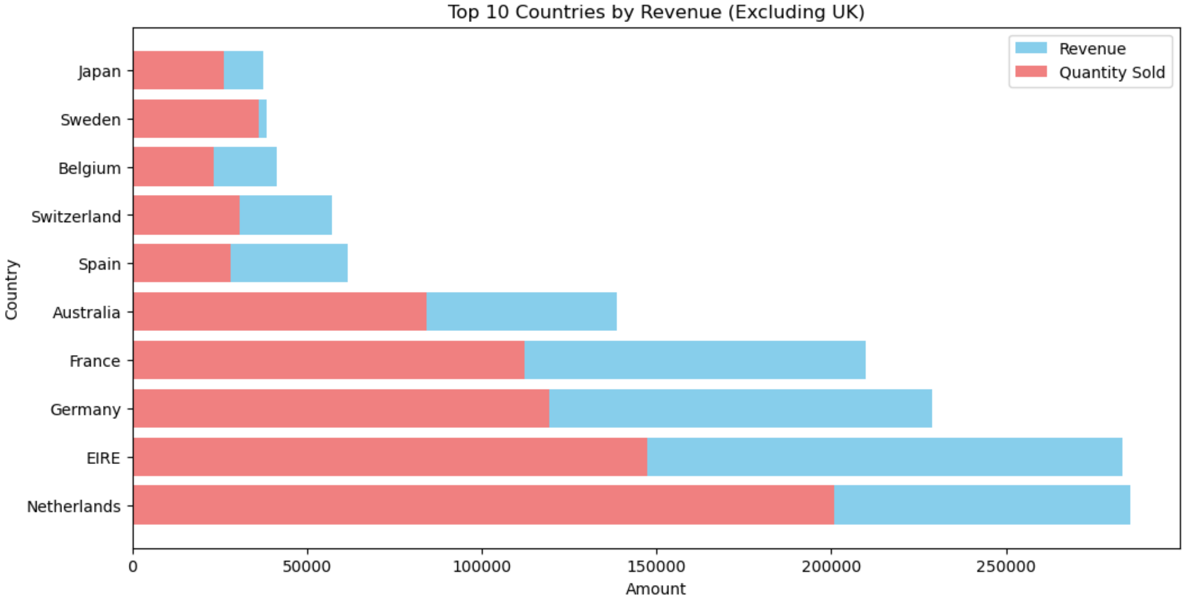
	Country	Revenue	Quantity
0	Australia	138521.31	84209
1	Austria	10198.68	4881
2	Bahrain	754.14	314
3	Belgium	41196.34	23237
4	Brazil	1143.60	356
5	Canada	3666.38	2763
6	Channel Islands	20450.44	9491
7	Cyprus	13590.38	6361
8	Czech Republic	826.74	671
9	Denmark	18955.34	8235
10	EIRE	283453.96	147447
11	European Community	1300.25	499
12	Finland	22546.08	10704
13	France	209715.11	112104
14	Germany	228867.14	119263
15	Greece	4760.52	1557
16	Hong Kong	15691.80	4773
17	Iceland	4310.00	2458
18	Israel	8135.26	4409
19	Italy	17483.24	8112
20	Japan	37416.37	26016
21	Lebanon	1693.88	386
22	Lithuania	1661.06	652
23	Malta	2725.59	970
24	Netherlands	285446.34	200937
25	Norway	36165.44	19338
26	Poland	7334.65	3684
27	Portugal	33747.10	16258
28	RSA	1002.31	352
29	Saudi Arabia	145.92	80
30	Singapore	21279.29	5241
31	Spain	61577.11	27951
32	Sweden	38378.33	36083
33	Switzerland	57089.90	30630
34	USA	3580.39	2458
35	United Arab Emirates	1902.28	982

	Country	Revenue	Quantity
36	Unspecified	4749.79	3300

In [36]:
```
# Sort by revenue in descending order to get the top 10 countries
top_10_countries = country_stats.sort_values(by='Revenue', ascending=False).hea

# Data Visualization (Bar Chart)
plt.figure(figsize=(12, 6))
plt.barh(top_10_countries['Country'], top_10_countries['Revenue'], color='skybl
plt.barh(top_10_countries['Country'], top_10_countries['Quantity'], color='ligh
plt.title('Top 10 Countries by Revenue (Excluding UK)')
plt.xlabel('Amount')
plt.ylabel('Country')
plt.legend()


# Show the plot
plt.show()
```



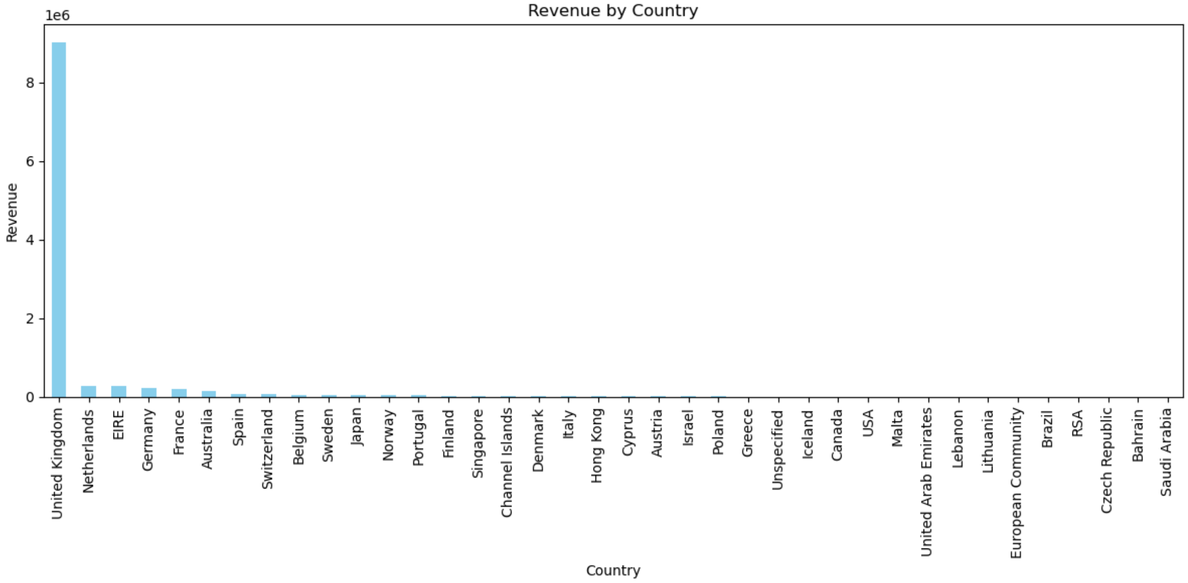
In [37]:
```
finaldata.Country.nunique()
```

Out[37]:
```
38
```

In [38]:
```
# Group data by country and calculate total revenue
country_revenue = finaldata.groupby('Country')['Revenue'].sum().sort_values(asc

# Data Visualization (Bar Chart)
plt.figure(figsize=(12, 6))
country_revenue.plot(kind='bar', color='skyblue')
plt.title('Revenue by Country')
plt.xlabel('Country')
plt.ylabel('Revenue')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability

# Show the plot
plt.tight_layout()
plt.show()
```



In [39]:

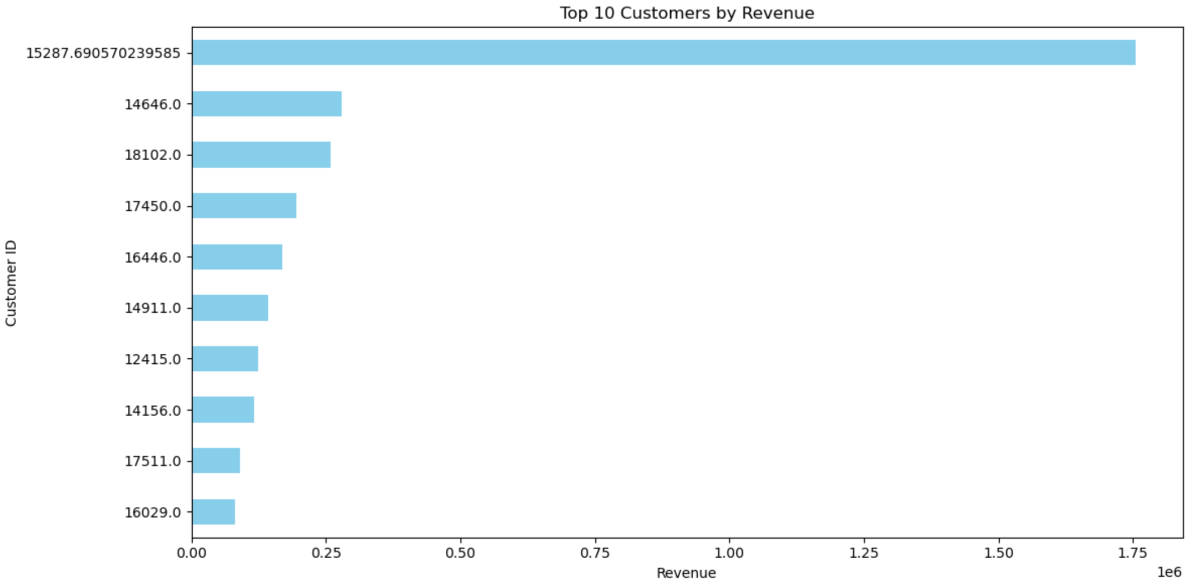
```
# Group data by customer and calculate total revenue
customer_revenue = finaldata.groupby('CustomerID')['Revenue'].sum().sort_values

# Select the top 10 customers
top_10_customers = customer_revenue.head(10)

# Reverse the order for visualization
top_10_customers = top_10_customers[::-1]

# Data Visualization (Horizontal Bar Chart)
plt.figure(figsize=(12, 6))
top_10_customers.plot(kind='barh', color='skyblue')
plt.title('Top 10 Customers by Revenue')
plt.xlabel('Revenue')
plt.ylabel('Customer ID')

# Show the plot
plt.tight_layout()
plt.show()
```



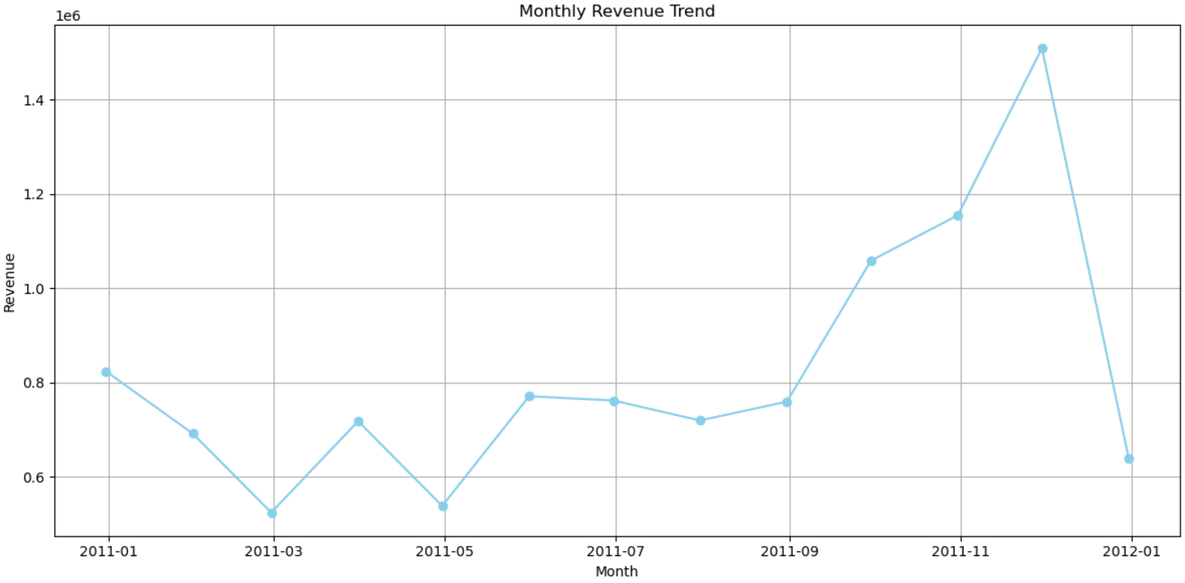
In [40]:

```
# Convert 'InvoiceDate' to datetime if it's not already
finaldata['InvoiceDate'] = pd.to_datetime(finaldata['InvoiceDate'])

# Group data by month and calculate total revenue
monthly_revenue = finaldata.groupby(pd.Grouper(key='InvoiceDate', freq='M'))['R

# Data Visualization (Line Plot)
plt.figure(figsize=(12, 6))
plt.plot(monthly_revenue.index, monthly_revenue.values, marker='o', color='skyb
plt.title('Monthly Revenue Trend')
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.grid(True)

# Show the plot
plt.tight_layout()
plt.show()
```



```
In [41]: finaldata
```

Out[41]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Uni Kingd
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	Frar
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	Frar
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	Frar
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	Frar

531283 rows × 9 columns

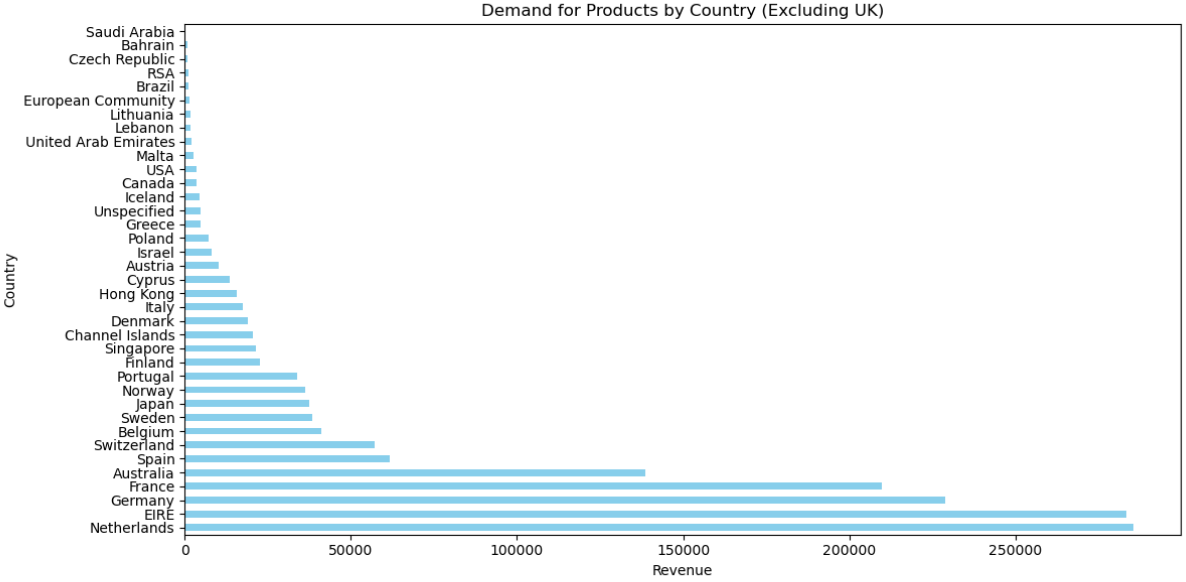
In [44]:

```
# Exclude the United Kingdom from the data
finaldata_filtered = finaldata[finaldata['Country'] != 'United Kingdom']

# Group data by country and calculate total revenue
country_revenue = finaldata_filtered.groupby('Country')['Revenue'].sum().sort_v

# Data Visualization (Horizontal Bar Chart)
plt.figure(figsize=(12, 6))
country_revenue.plot(kind='barh', color='skyblue')
plt.title('Demand for Products by Country (Excluding UK)')
plt.xlabel('Revenue')
plt.ylabel('Country')

# Show the plot
plt.tight_layout()
plt.show()
```



In [ ]: