In [ ]:

```
#conda install -c conda-forge mlxtend
```

In [1]:

```
#Install mlxtend and apriori
#Install mlxtend using belwo comand in conda, if it doesn't exist in spyder
#conda install -c conda-forge mlxtend

import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [2]:

```
df = pd.read_excel(r"C:\Users\GIRISH\Desktop\INTROTALLENT\PYTHON\ML PROJECT\104380_Pytho
```

In [3]:

```
#check number of rows and columns
df.shape
```

Out[3]:

```
(541909, 8)
```

In [4]:

```
#check data
df.head()
```

Out[4]:

	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 [5]:

```
#print unique country names from country coluns
df['Country'].unique()
```

Out[5]:

```
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 [ ]:

```
#Since buyer behavior differs from one geography to other and hence we
#will take one country at a time for this study
```

In [6]:

```
#some of the descriptions have spaces that need to be removed
df['Description'] = df['Description'].str.strip()
```

In [7]:

```
#Check if an invoice number is missing
df.isnull().sum()
```

Out[7]:

```
InvoiceNo          0
StockCode          0
Description     1455
Quantity           0
```

```
InvoiceDate          0
UnitPrice            0
CustomerID      135080
Country              0
dtype: int64
```

In [8]:

```
#drop the rows that don’t have invoice numbers
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
```

In [9]:

```
#Looking at sales for France only for ease
basket = (df[df['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity'].sum()
          .unstack()
          .reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

In [10]:

```
basket.shape
```

Out[10]:

```
(461, 1564)
```

In [ ]:

```
#basket.to_excel(r"C:\Users\GIRISH\Desktop\INTROTALLENT\PYTHON\New folder.xlsx")
```

In [11]:

```
# Encode -ve or 0 value transaction to 0 and +ve one to 1
def replace_quantity(x):
    if x >= 1:
        return 1
    else:
        return 0

# Apply Encoding
basket_sets = basket.applymap(replace_quantity)
```

In [ ]:

```
#basket_sets.to_excel(r"C:\Users\Mukesh\Desktop\basket_data.xlsx")
```

In [13]:

```
#Delete POSTAGE item from the data. It is included in many bills toadd postage charge
basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

In [14]:

```
#generate frequent item sets that have a support of at least 7%
#(this number was chosen so that I could get enough useful examples)
frequent_itemsets = apriori(basket_sets, min_support=0.07,
                            use_colnames=True)
```

```
C:\Users\GIRISH\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:110: D
eprecationWarning: DataFrames with non-bool types result in worse computationalperforman
ce and their support might be discontinued in the future.Please use a DataFrame with boo
l type
  warnings.warn(
```

In [15]:

```
#The final step is to generate the rules with their corresponding support, confidence an
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules
```

Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zha
0	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN SPACEBOY)	0.143167	0.117137	0.075922	0.530303	4.527217	0.059152	1.879645	
1	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN	0.117137	0.143167	0.075922	0.648148	4.527217	0.059152	2.435209	

		CIRCUS PARADE)							
2	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)	0.145336	0.143167	0.086768	0.597015	4.170059	0.065961	2.126215
3	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.143167	0.145336	0.086768	0.606061	4.170059	0.065961	2.169531
4	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN SPACEBOY)	0.145336	0.117137	0.088937	0.611940	5.224157	0.071913	2.275071
5	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.117137	0.145336	0.088937	0.759259	5.224157	0.071913	3.550142
6	(SET/20 RED RETROSPOT PAPER NAPKINS)	(SET/6 RED SPOTTY PAPER CUPS)	0.112798	0.117137	0.086768	0.769231	6.566952	0.073555	3.825741
7	(SET/6 RED SPOTTY PAPER CUPS)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.117137	0.112798	0.086768	0.740741	6.566952	0.073555	3.422064
8	(SET/20 RED RETROSPOT PAPER NAPKINS)	(SET/6 RED SPOTTY PAPER PLATES)	0.112798	0.108460	0.086768	0.769231	7.092308	0.074534	3.863341
9	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.108460	0.112798	0.086768	0.800000	7.092308	0.074534	4.436009
10	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.117137	0.108460	0.104121	0.888889	8.195556	0.091417	8.023861
11	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.108460	0.117137	0.104121	0.960000	8.195556	0.091417	22.071584
12	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER PLATES)	0.086768	0.108460	0.084599	0.975000	8.989500	0.075188	35.661605
13	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER CUPS)	0.086768	0.117137	0.084599	0.975000	8.323611	0.074435	35.314534
14	(SET/6 RED SPOTTY PAPER CUPS,	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.104121	0.112798	0.084599	0.812500	7.203125	0.072854	4.731743

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zha
	SET/6 RED SPOTTY...									
15	(SET/20 RED RETROSPOT PAPER NAPKINS)	(SET/6 RED SPOTTY PAPER CUPS, SET/6 RED SPOTTY...	0.112798	0.104121	0.084599	0.750000	7.203125	0.072854	3.583514	
16	(SET/6 RED SPOTTY PAPER CUPS)	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	0.117137	0.086768	0.084599	0.722222	8.323611	0.074435	3.287636	
17	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	0.108460	0.086768	0.084599	0.780000	8.989500	0.075188	4.151055	

In [16]:

```
#We can filter the dataframe using standard pandas code.
#In this case, look for a large lift (6) and high confidence (.8):
rules[ (rules['lift'] >= 6) & (rules['confidence'] >= 0.8) ]
```

Out[16]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zha
10	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.117137	0.108460	0.104121	0.888889	8.195556	0.091417	8.023861	
11	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.108460	0.117137	0.104121	0.960000	8.195556	0.091417	22.071584	
12	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER PLATES)	0.086768	0.108460	0.084599	0.975000	8.989500	0.075188	35.661605	
13	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER CUPS)	0.086768	0.117137	0.084599	0.975000	8.323611	0.074435	35.314534	
14	(SET/6 RED SPOTTY PAPER CUPS, SET/6 RED SPOTTY...	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.104121	0.112798	0.084599	0.812500	7.203125	0.072854	4.731743	

In [ ]:

```
#export association rules to excel
#rules.to_excel(r"C:\Users\Mukesh\Desktop\output.xlsx")
```