

```
In [23]: #Install mlxtend and apriori
#Install mlxtend using below command in conda, if it doesn't exist in jupyter or in spyder notebook
#conda install -c conda-forge mlxtend
# in conda prompt type there pip install mlxtend then press enter button there
```

```
In [24]: # import libraries
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

#ignore/ disable warnings
import warnings
warnings.filterwarnings("ignore")
```

```
In [25]: #df = pd.read_excel('D:/M U K E S H/T R A I N I N G/PYTHON/CODES/Association Rule/Online Retail.xlsx')
import pandas as pd
#df = pd.read_excel(r'C:\Users\Sanjay Lohar\Downloads\Online Retail\Online Retail.xlsx')
df= pd.read_excel(r"C:\Users\Sanjay Lohar\OneDrive\Desktop\Online Retail.xlsx")
df
```

Out[25]:

	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
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

541909 rows × 8 columns

```
In [26]: df.head()
```

Out[26]:

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

```
In [28]: #Check if an invoice number is missing
df.isnull().sum()
```

Out[28]:

```
InvoiceNo      0
StockCode      0
Description    1455
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID    135080
Country        0
dtype: int64
```

```
In [29]: #drop the rows that don't have invoice numbers
# i.e. if Invoice No is not available or null then drop that row for this filter only for 1 or more particular c
# use subset function for this advanced filtering to dropping rows where invoice no is not available or null
# subset means Define in which columns to look for missing values. e.g. subset=['name', 'toy']
# subset : column label or sequence of labels, optional
```

```
# Labels along other axis to consider, e.g. if you are dropping rows
# these would be a list of columns to include.
# Keep the DataFrame with valid entries in the same variable if inplace=True
# by default inplace takes False
df.dropna(axis=0, subset=['InvoiceNo'])
```

Out[29]:

	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
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

541909 rows × 8 columns

In [30]:

```
# to check dataframe again
# InvoiceNo column have all the entries and doesnt have any non null entry
# i.e.our df shape is 541909 rows × 8 columns and available entries in InvoiceNo column is 541909 that is equal
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      540454 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 [31]:

```
# to know the values in the country column
country_count = df["Country"].value_counts()
country_count
```

```
Out[31]: United Kingdom      495478
         Germany            9495
         France            8557
         EIRE              8196
         Spain             2533
         Netherlands       2371
         Belgium           2069
         Switzerland       2002
         Portugal          1519
         Australia         1259
         Norway            1086
         Italy              803
         Channel Islands    758
         Finland           695
         Cyprus            622
         Sweden            462
         Unspecified       446
         Austria           401
         Denmark           389
         Japan             358
         Poland            341
         Israel            297
         USA               291
         Hong Kong         288
         Singapore         229
         Iceland           182
         Canada            151
         Greece            146
         Malta             127
         United Arab Emirates 68
         European Community 61
         RSA               58
         Lebanon           45
         Lithuania         35
         Brazil            32
         Czech Republic    30
         Bahrain           19
         Saudi Arabia      10
Name: Country, dtype: int64
```

```
In [32]: #consolidate the items into 1 transaction per row with each product
         #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 [33]: # to check
         basket
```

Out[33]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	...
InvoiceNo											
536370	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536974	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537065	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537463	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
C579532	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
C579562	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
C580161	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
C580263	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
C581316	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

461 rows × 1564 columns

```
In [34]: # Check how does data look after transformation
         #basket.to_excel('C:\M U K E S H\T R A I N I N G\PYTHON\CODES\Association Rule\France_Encoded_Data.xlsx')
```

```
In [35]: # convert all positive values to 1 and everythig else to 0
         # Encode -ve or 0 value transaction to 0 and +ve one to 1
         def encode_units(x):
             if x <= 0:
                 return 0
```

```

if x >= 1:
    return 1

```

```

In [36]: # Apply Encoding
basket_sets = basket.applymap(encode_units)
basket_sets

```

Out[36]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	...
InvoiceNo											
536370	0	0	0	0	0	0	0	0	0	0	...
536852	0	0	0	0	0	0	0	0	0	0	...
536974	0	0	0	0	0	0	0	0	0	0	...
537065	0	0	0	0	0	0	0	0	0	0	...
537463	0	0	0	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
C579532	0	0	0	0	0	0	0	0	0	0	...
C579562	0	0	0	0	0	0	0	0	0	0	...
C580161	0	0	0	0	0	0	0	0	0	0	...
C580263	0	0	0	0	0	0	0	0	0	0	...
C581316	0	0	0	0	0	0	0	0	0	0	...

461 rows × 1564 columns

```

In [37]: #Delete POSTAGE item from the data. It is included in many bills to add postage charge
basket_sets=basket_sets.drop('POSTAGE', axis=1)
basket_sets

```

Out[37]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	...
InvoiceNo											
536370	0	0	0	0	0	0	0	0	0	0	...
536852	0	0	0	0	0	0	0	0	0	0	...
536974	0	0	0	0	0	0	0	0	0	0	...
537065	0	0	0	0	0	0	0	0	0	0	...
537463	0	0	0	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
C579532	0	0	0	0	0	0	0	0	0	0	...
C579562	0	0	0	0	0	0	0	0	0	0	...
C580161	0	0	0	0	0	0	0	0	0	0	...
C580263	0	0	0	0	0	0	0	0	0	0	...
C581316	0	0	0	0	0	0	0	0	0	0	...

461 rows × 1563 columns

```

In [38]: #generate frequent item sets that have a support of at least 7%
#(this number was chosen so that I could get enough useful examples)
# use_colnames=True means use my column names as item names if false then it will return column index no.
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
frequent_itemsets

```

Out [38]:	support	itemsets
	0 0.082430	(ALARM CLOCK BAKELIKE GREEN)
	1 0.086768	(ALARM CLOCK BAKELIKE PINK)
	2 0.080260	(ALARM CLOCK BAKELIKE RED)
	3 0.084599	(DOLLY GIRL LUNCH BOX)
	4 0.082430	(JUMBO BAG RED RETROSPOT)
	5 0.106291	(LUNCH BAG APPLE DESIGN)
	6 0.071584	(LUNCH BAG DOLLY GIRL DESIGN)
	7 0.130152	(LUNCH BAG RED RETROSPOT)
	8 0.101952	(LUNCH BAG SPACEBOY DESIGN)
	9 0.099783	(LUNCH BAG WOODLAND)
	10 0.121475	(LUNCH BOX WITH CUTLERY RETROSPOT)
	11 0.088937	(MINI PAINT SET VINTAGE)
	12 0.086768	(PACK OF 72 RETROSPOT CAKE CASES)
	13 0.143167	(PLASTERS IN TIN CIRCUS PARADE)
	14 0.117137	(PLASTERS IN TIN SPACEBOY)
	15 0.145336	(PLASTERS IN TIN WOODLAND ANIMALS)
	16 0.160521	(RABBIT NIGHT LIGHT)
	17 0.082430	(RED RETROSPOT CHARLOTTE BAG)
	18 0.117137	(RED RETROSPOT MINI CASES)
	19 0.154013	(RED TOADSTOOL LED NIGHT LIGHT)
	20 0.106291	(REGENCY CAKESTAND 3 TIER)
	21 0.073753	(RETROSPOT TEA SET CERAMIC 11 PC)
	22 0.091106	(ROUND SNACK BOXES SET OF 4 FRUITS)
	23 0.134490	(ROUND SNACK BOXES SET OF 4 WOODLAND)
	24 0.112798	(SET/20 RED RETROSPOT PAPER NAPKINS)
	25 0.117137	(SET/6 RED SPOTTY PAPER CUPS)
	26 0.108460	(SET/6 RED SPOTTY PAPER PLATES)
	27 0.106291	(SPACEBOY LUNCH BOX)
	28 0.104121	(STRAWBERRY LUNCH BOX WITH CUTLERY)
	29 0.080260	(TEA PARTY BIRTHDAY CARD)
	30 0.075922	(PLASTERS IN TIN SPACEBOY, PLASTERS IN TIN CIR...
	31 0.086768	(PLASTERS IN TIN WOODLAND ANIMALS, PLASTERS IN...
	32 0.088937	(PLASTERS IN TIN WOODLAND ANIMALS, PLASTERS IN...
	33 0.086768	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...
	34 0.086768	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...
	35 0.104121	(SET/6 RED SPOTTY PAPER PLATES, SET/6 RED SPOT...
	36 0.084599	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...

```
In [39]: #The final step is to generate the rules with their corresponding support, confidence and lift:
# take only frequent itemsets whose lift should be greater than 1
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules
```

Out[39]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN CIRCUS PARADE)	0.117137	0.143167	0.075922	0.648148	4.527217	0.059152	2.435209
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN SPACEBOY)	0.143167	0.117137	0.075922	0.530303	4.527217	0.059152	1.879645
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 PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.108460	0.117137	0.104121	0.960000	8.195556	0.091417	22.071584
11	(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
12	(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
13	(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
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
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 PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	0.108460	0.086768	0.084599	0.780000	8.989500	0.075188	4.151055
17	(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

In [40]: # rules.to\_excel(r'C:\Users\Admin\Desktop\France\_rules.xlsx')

In [41]: # Filtering rules based on conditions  
#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[41]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
10	(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
11	(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
12	(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
13	(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
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 [ ]:

In [42]: r1 = rules[rules["confidence"]>=0.3]

In [43]: r1

Out [43]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN CIRCUS PARADE)	0.117137	0.143167	0.075922	0.648148	4.527217	0.059152	2.435209
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN SPACEBOY)	0.143167	0.117137	0.075922	0.530303	4.527217	0.059152	1.879645
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 PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.108460	0.117137	0.104121	0.960000	8.195556	0.091417	22.071584
11	(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
12	(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
13	(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
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
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 PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	0.108460	0.086768	0.084599	0.780000	8.989500	0.075188	4.151055
17	(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

In [44]:

```
r2 = rules.iloc[:, 0:7]
r2 = pd.DataFrame(r2)
r2
```

Out [44]:

	antecedents		consequents	antecedent support	consequent support	support	confidence	lift
0	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN CIRCUS PARADE)		0.117137	0.143167	0.075922	0.648148	4.527217
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN SPACEBOY)		0.143167	0.117137	0.075922	0.530303	4.527217
2	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)		0.145336	0.143167	0.086768	0.597015	4.170059
3	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)		0.143167	0.145336	0.086768	0.606061	4.170059
4	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN SPACEBOY)		0.145336	0.117137	0.088937	0.611940	5.224157
5	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)		0.117137	0.145336	0.088937	0.759259	5.224157
6	(SET/20 RED RETROSPOT PAPER NAPKINS)	(SET/6 RED SPOTTY PAPER CUPS)		0.112798	0.117137	0.086768	0.769231	6.566952
7	(SET/6 RED SPOTTY PAPER CUPS)	(SET/20 RED RETROSPOT PAPER NAPKINS)		0.117137	0.112798	0.086768	0.740741	6.566952
8	(SET/20 RED RETROSPOT PAPER NAPKINS)	(SET/6 RED SPOTTY PAPER PLATES)		0.112798	0.108460	0.086768	0.769231	7.092308
9	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)		0.108460	0.112798	0.086768	0.800000	7.092308
10	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)		0.108460	0.117137	0.104121	0.960000	8.195556
11	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)		0.117137	0.108460	0.104121	0.888889	8.195556
12	(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
13	(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
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
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
16	(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
17	(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

In [ ]:

In [ ]: