Workshop 5 - Market Basket Analysis using Apriori Algorithm

STUDENT NO: 2302546

This notebook focuses on generating association rules for market basket analysis using apriori algorithm.

Importing necessary libraries

In [1]:

```
import matplotlib.pyplot as plt; plt.rcdefaults()
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('Workshop-5-dataset.zip', sep='\t',dtype=np.str)
```

Q1. Display the first five rows of the dataset

Answer:

In [3]:

```
df.head()
```

Out[3]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4
0	536365	01/12/2010	08:26	WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	KNITTED UNION FLAG HOT WATER BOTTLE	RED WOOLLY HOTTIE WHITE HEART
1	536366	01/12/2010	08:28	HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT	NaN	NaN	NaN
2	536367	01/12/2010	08:34	ASSORTED COLOUR BIRD ORNAMENT	POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN	FELTCRAFT PRINCESS CHARLOTTE DOLL	IVORY KNITTED MUG COSY
3	536368	01/12/2010	08:34	JAM MAKING SET WITH JARS	RED COAT RACK PARIS FASHION	YELLOW COAT RACK PARIS FASHION	BLUE COAT RACK PARIS FASHION	NaN
4	536369	01/12/2010	08:35	BATH BUILDING BLOCK WORD	NaN	NaN	NaN	NaN

5 rows × 44 columns

Q2. How many rows and columns are there in the dataset?

Answer:

In [4]:
```
df.shape
```

Out[4]:
```
(31941, 44)
```

There are 31941 rows and 44 columns in this dataset

GENERATING UNIQUE DATASET

In [5]:
```
STUDENT_NAME = 'ArabambiAkinyemi'
STUDENT_NO = '2546'
```

In [6]:
```
np.random.seed(int(STUDENT_NO))
unique_id = int('2' + STUDENT_NO)
rows = np.random.choice(df.index.values, unique_id)
data = df.loc[rows]
```

In [7]:
```
file_name = STUDENT_NAME + "_" + STUDENT_NO + ".csv"
data.to_csv(file_name)
```

Q3. How many unique dates are there in the dataset? HINT: You can use the .nunique() function available in Pandas.

Answer:

In [8]:
```
# Get the number of unique dates in the "date" column
num_unique_dates = df["Date"].nunique()

# Print the result
print("Number of unique dates in the dataset:", num_unique_dates)
```

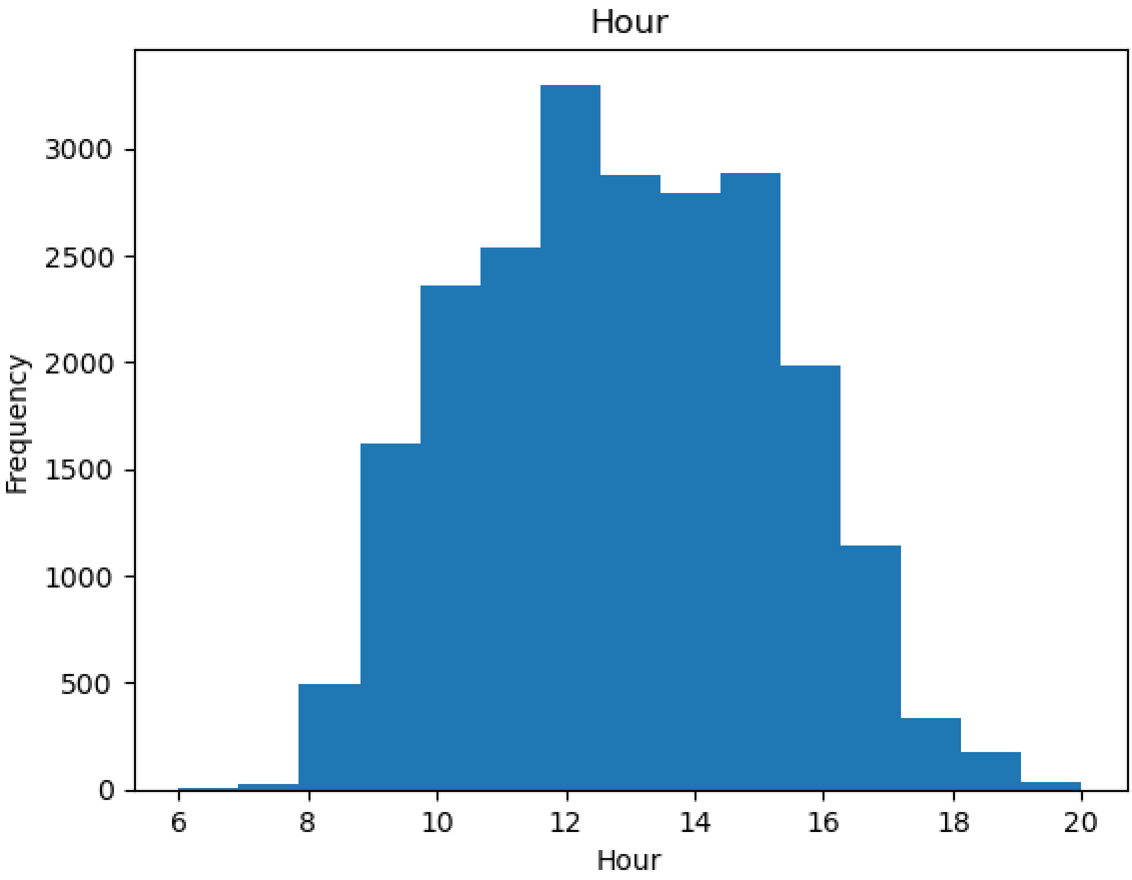
```
Number of unique dates in the dataset: 305
```

Data Visualisation

In [9]:
```
data['Hour'] = pd.to_datetime(data['Time'], format='%H:%M').dt.hour
```

In [10]:
```
hour_hist = data.hist(column="Hour", bins=15, grid=False)

for ax in hour_hist.flatten():
    ax.set_xlabel("Hour")
    ax.set_ylabel("Frequency")
```



APRIORI ALGORITHM

```
In [11]: # import apyori
         from apyori import apriori
```

```
In [12]: data.head(1)
```

Out[12]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4
19246	564166	23/08/2011	13:40	NATURAL SLATE HEART CHALKBOARD	CHILDRENS APRON APPLES DESIGN	WRAP VINTAGE DOILY	WRAP VINTAGE LEAF DESIGN	WRAP POPPIES DESIGN

1 rows × 45 columns

Data Preprocessing

```
In [13]: items_df=data[data.columns[3:44]]
```

```
In [14]: items_df.head()
```

Out[14]:

	item_0	item_1	item_2	item_3	item_4	item_5	item_6	
19246	NATURAL SLATE HEART CHALKBOARD	CHILDRENS APRON APPLES DESIGN	WRAP VINTAGE DOILY	WRAP VINTAGE LEAF DESIGN	WRAP POPPIES DESIGN	WRAP GINGHAM ROSE	WRAP ENGLISH ROSE	F
5135	WOODEN ROUNDERS GARDEN SET	WHITE BAMBOO RIBS LAMPSHADE	NaN	NaN	NaN	NaN	NaN	
15906	METAL SIGN TAKE IT OR LEAVE IT	METAL SIGN EMPIRE TEA	HAND OVER THE CHOCOLATE SIGN	METAL SIGN HER DINNER IS SERVED	DOORMAT UNION JACK GUNS AND ROSES	DOORMAT FANCY FONT HOME SWEET HOME	DOORMAT KEEP CALM AND COME IN	
10141	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
9075	JUMBO BAG WOODLAND ANIMALS	JUMBO STORAGE BAG SUKI	JUMBO BAG RED RETROSPOT	PACK OF 72 RETROSPOT CAKE CASES	RED RETROSPOT MINI CASES	WASH BAG VINTAGE ROSE PAISLEY	ASSORTED COLOURS SILK FAN	P V

5 rows × 41 columns

In [15]:

```python
baskets = items_df.T.apply(lambda x: x.dropna().tolist()).tolist()
```

In [16]:

```python
for i in baskets[:5]:
    print(i)
```

```
['NATURAL SLATE HEART CHALKBOARD', 'CHILDRENS APRON APPLES DESIGN', 'WRAP  VINTAGE
DOILY', 'WRAP VINTAGE LEAF DESIGN', 'WRAP POPPIES  DESIGN', 'WRAP GINGHAM ROSE',
'WRAP ENGLISH ROSE', 'WRAP PINK FAIRY CAKES', 'CARD DOLLY GIRL', 'PENNY FARTHING B
IRTHDAY CARD', 'BIRTHDAY CARD, RETRO SPOT', 'VINTAGE KID DOLLY CARD', 'FELTCRAFT 6
FLOWER FRIENDS', 'RED FLORAL FELTCRAFT SHOULDER BAG', 'BOX OF VINTAGE ALPHABET BLO
CKS', 'HOME BUILDING BLOCK WORD', 'CARD BILLBOARD FONT', 'COWBOYS AND INDIANS BIRT
HDAY CARD', 'SET OF 6 T-LIGHTS SANTA', 'SET OF 2 CHRISTMAS DECOUPAGE CANDLE', 'YUL
ETIDE IMAGES GIFT WRAP SET', 'RIBBON REEL MAKING SNOWMEN', 'RIBBON REEL CHRISTMAS
PRESENT', '3D VINTAGE CHRISTMAS STICKERS', 'ROTATING SILVER ANGELS T-LIGHT HLDR',
'SET OF 4 SANTA PLACE SETTINGS', 'SILVER STARS TABLE DECORATION', 'NOEL WOODEN BLO
CK LETTERS', 'GARLAND WITH STARS AND BELLS', 'CHRISTMAS TREE HANGING SILVER', 'CHE
RUB HEART DECORATION SILVER', '60 CAKE CASES VINTAGE CHRISTMAS', 'VINTAGE CHRISTMA
S TABLECLOTH']
['WOODEN ROUNDERS GARDEN SET', 'WHITE BAMBOO RIBS LAMPSHADE']
['METAL SIGN TAKE IT OR LEAVE IT', 'METAL SIGN EMPIRE TEA', 'HAND OVER THE CHOCOLA
TE   SIGN', 'METAL SIGN HER DINNER IS SERVED', 'DOORMAT UNION JACK GUNS AND ROSE
S', 'DOORMAT FANCY FONT HOME SWEET HOME', 'DOORMAT KEEP CALM AND COME IN', 'DOORMA
T WELCOME TO OUR HOME', 'DOORMAT FAIRY CAKE', 'DOORMAT WELCOME PUPPIES', 'DOORMAT
3 SMILEY CATS', 'DOORMAT RESPECTABLE HOUSE', 'DOORMAT NEIGHBOURHOOD WITCH', 'DOORM
AT AIRMAIL']
[]
['JUMBO BAG WOODLAND ANIMALS', 'JUMBO STORAGE BAG SUKI', 'JUMBO BAG RED RETROSPO
T', 'PACK OF 72 RETROSPOT CAKE CASES', 'RED RETROSPOT MINI CASES', 'WASH BAG VINTA
GE ROSE PAISLEY', 'ASSORTED COLOURS SILK FAN', 'PLASTERS IN TIN WOODLAND ANIMALS',
'PLASTERS IN TIN CIRCUS PARADE', 'ROMANTIC PINKS RIBBONS', 'BABY BOOM RIBBONS', 'C
HOCOLATE BOX RIBBONS', 'SCANDINAVIAN REDS RIBBONS', 'TEA PARTY BIRTHDAY CARD', 'RI
NG OF ROSES BIRTHDAY CARD', 'ELEPHANT, BIRTHDAY CARD', 'CARD CIRCUS PARADE', 'CARD
SUKI BIRTHDAY', 'SLEEPING CAT ERASERS', 'CUTE CATS TAPE', 'STARS GIFT TAPE', 'RED
RETROSPOT TAPE', 'WOODLAND PARTY BAG + STICKER SET', 'LUNCH BOX WITH CUTLERY RETRO
SPOT', 'SET/6 RED SPOTTY PAPER CUPS', 'CHARLOTTE BAG SUKI DESIGN', 'ALARM CLOCK BA
KELIKE GREEN', 'ALARM CLOCK BAKELIKE RED', 'ALARM CLOCK BAKELIKE PINK', 'GUMBALL C
OAT RACK', 'JUNGLE POPSICLES ICE LOLLY MOULDS', 'RECIPE BOX PANTRY YELLOW DESIGN',
'PACK OF 20 NAPKINS PANTRY DESIGN', 'RED RETROSPOT OVEN GLOVE', 'WOODLAND CHARLOTT
E BAG', 'LUNCH BAG WOODLAND', 'ROUND SNACK BOXES SET OF4 WOODLAND', 'ROUND SNACK B
OXES SET OF 4 FRUITS', 'STRAWBERRY LUNCH BOX WITH CUTLERY', 'RECIPE BOX BLUE SKETC
HBOOK DESIGN', 'RECIPE BOX RETROSPOT']
```

Algorithm Parameters

Minimum Support = 0.01 , Minimum Confidence = 0.2 , Minimum Lift = 3 , Minimum length = 2

In [17]:

```
association_rules = apriori(baskets, min_support=0.01, min_confidence=0.2,
                            min_lift=3, min_length=2)
association_results = list(association_rules)
```

Q. Have a look at the rules and find out how many association rules we were able to generate?

In [18]:

```
print('Rules generated: ', len(association_results))
```

```
Rules generated:  83
```

In [19]:

```
print(association_results[0])
```

```
RelationRecord(items=frozenset({'PACK OF 72 RETROSPOT CAKE CASES', '60 TEATIME FAI
RY CAKE CASES'}), support=0.010378781158520358, ordered_statistics=[OrderedStatist
ic(items_base=frozenset({'60 TEATIME FAIRY CAKE CASES'}), items_add=frozenset({'PA
CK OF 72 RETROSPOT CAKE CASES'}), confidence=0.3939393939393939, lift=9.3689425904
61577), OrderedStatistic(items_base=frozenset({'PACK OF 72 RETROSPOT CAKE CASE
S'}), items_add=frozenset({'60 TEATIME FAIRY CAKE CASES'}), confidence=0.246835443
03797467, lift=9.368942590461577)])
```

Change the index value [0] of 'association_results[0]' to see a different rule

```
In [20]: print(association_results[1])
```

```
RelationRecord(items=frozenset({'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIK
E IVORY'}), support=0.010112658564712143, ordered_statistics=[OrderedStatistic(ite
ms_base=frozenset({'ALARM CLOCK BAKELIKE GREEN'}), items_add=frozenset({'ALARM CLO
CK BAKELIKE IVORY'}), confidence=0.31754874651810583, lift=16.885504808955695), Or
deredStatistic(items_base=frozenset({'ALARM CLOCK BAKELIKE IVORY'}), items_add=fro
zenset({'ALARM CLOCK BAKELIKE GREEN'}), confidence=0.5377358490566038, lift=16.885
504808955695)])
```

```
In [21]: print(association_results[15])
```

```
RelationRecord(items=frozenset({'HEART OF WICKER LARGE', 'HEART OF WICKER SMAL
L'}), support=0.014814157721990597, ordered_statistics=[OrderedStatistic(items_bas
e=frozenset({'HEART OF WICKER LARGE'}), items_add=frozenset({'HEART OF WICKER SMAL
L'}), confidence=0.5107033639143731, lift=13.740236327939686), OrderedStatistic(it
ems_base=frozenset({'HEART OF WICKER SMALL'}), items_add=frozenset({'HEART OF WICK
ER LARGE'}), confidence=0.3985680190930788, lift=13.740236327939687)])
```

Analysing the Results

Analysing the results may include checking what kind of items are featured in some of the rules. Using the method in the cell below, you can display the results.

```
In [22]: def display_rules(association_results):
             for item in association_results:
                 pair = item[0]
                 items = [x for x in pair]
                 print("Rule: " + items[0] + " -> " + items[1])
                 print("Support: " + str(item[1]))
                 print("Confidence: " + str(item[2][0][2]))
                 print("Lift: " + str(item[2][0][3]))
                 print("=====================================")
```

Displaying the rules only for 10 results

```
In [23]: display_rules(association_results[:10])
```

```
Rule: PACK OF 72 RETROSPOT CAKE CASES -> 60 TEATIME FAIRY CAKE CASES
Support: 0.010378781158520358
Confidence: 0.3939393939393939
Lift: 9.368942590461577
=====================================
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE IVORY
Support: 0.010112658564712143
Confidence: 0.31754874651810583
Lift: 16.885504808955695
=====================================
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE PINK
Support: 0.012108577801827375
Confidence: 0.3802228412256267
Lift: 15.172573766854832
=====================================
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE RED
Support: 0.019693071941807 86
Confidence: 0.6183844011142062
Lift: 17.920430215322487
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE RED
Support: 0.011398917768118513
Confidence: 0.6061320754716981
Lift: 17.56536474753844
=====================================
Rule: ALARM CLOCK BAKELIKE PINK -> ALARM CLOCK BAKELIKE RED
Support: 0.014459327596912978
Confidence: 0.576991150442478
Lift: 16.72087722092045
=====================================
Rule: CHARLOTTE BAG SUKI DESIGN -> CHARLOTTE BAG PINK POLKADOT
Support: 0.011443271533753215
Confidence: 0.49710982658959535
Lift: 18.13566043736087
=====================================
Rule: HOT WATER BOTTLE TEA AND SYMPATHY -> CHOCOLATE HOT WATER BOTTLE
Support: 0.0111327951743103
Confidence: 0.40224358974358976
Lift: 19.629835442335445
=====================================
Rule: SPACEBOY LUNCH BOX -> DOLLY GIRL LUNCH BOX
Support: 0.015878648097223454
Confidence: 0.5830618892508143
Lift: 21.479923782759574
=====================================
Rule: GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA
Support: 0.016189124456666637
Confidence: 0.7115009746588694
Lift: 24.832044852413112
=====================================
```

In [24]:

```python
from collections import Counter

counter = Counter(baskets[0])
for i in baskets[1:]:
    if i != 'nan':
        counter.update(i)

del counter['nan']
counter.most_common(10)
```

Out[24]:
```
[('WHITE HANGING HEART T-LIGHT HOLDER', 1637),
 ('REGENCY CAKESTAND 3 TIER', 1566),
 ('JUMBO BAG RED RETROSPOT', 1423),
 ('PARTY BUNTING', 1167),
 ('LUNCH BAG RED RETROSPOT', 1126),
 ('SET OF 3 CAKE TINS PANTRY DESIGN', 1044),
 ('ASSORTED COLOUR BIRD ORNAMENT', 1014),
 ('PACK OF 72 RETROSPOT CAKE CASES', 969),
 ('LUNCH BAG  BLACK SKULL', 963),
 ("PAPER CHAIN KIT 50'S CHRISTMAS", 941)]
```

Q4 (a) How many of these items can you find in the rules you have just displayed?

In the association rule displayed above which is made up of the first 10 rows of the association rule, there is only one item from the most common top 10 rule (Out[27]) and the item is PACK OF 72 RETROSPOT CAKE CASES.

Q4 (b) Are all the top 10 items included? Provide an explanation as to why these items may be missing/present in the rules.

There is only one item from the top ten rule included in the most common.

This is because association rule mining works by identifying patterns of co-occurrence between items in transactions, and these patterns are influenced by factors such as the minimum support and confidence thresholds used for the analysis, the size and diversity of the dataset, and the choice of algorithm used for mining.

Therefore, it's possible that some of the top 10 frequent items may not have strong association rules with other items in the dataset, or their association rules may not have met the minimum support and confidence thresholds used in the analysis, resulting in their absence from the rules displayed. Additionally, it's also possible that some of the items in the rules may not be frequent items in the dataset but have strong association rules with other items, leading to their inclusion in the rules.

Q5. Run the apriori algorithm with the following three different settings:

Calculate the number of rules you get for each setting and how the quality of the rules differ in each setting.

- Setting 1: Min Support = 0.015, Min Confidence = 0.7, Min Lift = 3

Answer

In [25]:
```
association_rules1 = apriori(baskets, min_support=0.015, min_confidence=0.7,
                            min_lift=3, min_length=2)
association_results1 = list(association_rules1)
```

In [26]:
```
print('Rules generated: ', len(association_results1))
```

```
Rules generated:  5
```

In [27]:
```
print(association_results1[0])
```

```
RelationRecord(items=frozenset({'GARDENERS KNEELING PAD KEEP CALM', 'GARDENERS KNE
ELING PAD CUP OF TEA'}), support=0.01618912445666637, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'GARDENERS KNEELING PAD CUP OF TEA'}), items_add=fr
ozenset({'GARDENERS KNEELING PAD KEEP CALM'}), confidence=0.7115009746588694, lift
=24.832044852413112)])
```

In [28]:
```
def display_rules(association_results1):
 for item in association_results1:
     pair = item[0]
     items = [x for x in pair]
     print("Rule: " + items[0] + " -> " + items[1])
     print("Support: " + str(item[1]))
     print("Confidence: " + str(item[2][0][2]))
     print("Lift: " + str(item[2][0][3]))
     print("=====================================")
```

In [29]:
```
display_rules(association_results1[:5])
```

```
Rule: GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA
Support: 0.01618912445666637
Confidence: 0.7115009746588694
Lift: 24.832044852413112
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER
Support: 0.020535793488867204
Confidence: 0.7834179357021995
Lift: 22.189624093394208
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER
Support: 0.02523729264614566
Confidence: 0.714824120603015
Lift: 19.726345927926044
=====================================
Rule: ROSES REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER
Support: 0.019870487004346668
Confidence: 0.7580372250423011
Lift: 20.918858354716917
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER
Support: 0.017519737425707442
Confidence: 0.8531317494600432
Lift: 23.543094765393068
=====================================
```

Based on the rules displayed above, we can see that all of the rules have a relatively high support, indicating that the items in the rules are frequently purchased together. However, the confidence and lift values vary across the different rules.

For example, the rule "PINK REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER" has a high confidence value of 0.853, indicating that when a customer buys a pink regency teacup and saucer, they are very likely to also buy a roses regency teacup and saucer. This is further supported by a high lift value of 23.54, indicating that the rule is much stronger than what we would expect by chance alone.

In contrast, the rule "GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA" has a lower confidence value of 0.712 and a high lift value of 24.83. This indicates that the association between the two items is not as strong as the other rules, but when a customer does purchase the "GARDENERS KNEELING PAD KEEP CALM" item, they are much more likely to also purchase the "GARDENERS KNEELING PAD CUP OF TEA" item than what we would expect by chance.

Overall, the quality of the association rules depends on the specific context and goals of each analysis and data structure. However, based on the support, confidence, and lift values, we can see that these rules do have some predictive power and could be useful for guiding product recommendations or marketing strategies.

- Setting 2: Min Support = 0.009, Min Confidence = 0.5, Min Lift = 3

In [30]:

```
association_rules2 = apriori(baskets, min_support=0.009, min_confidence=0.5,
                            min_lift=3, min_length=2)
association_results2 = list(association_rules2)
```

In [31]:

```
print('Rules generated: ', len(association_results2))
```

```
Rules generated:  51
```

In [32]:

```
print(association_results2[0])
```

```
RelationRecord(items=frozenset({'ALARM CLOCK BAKELIKE CHOCOLATE', 'ALARM CLOCK BAK
ELIKE RED'}), support=0.009225583252018096, ordered_statistics=[OrderedStatistic(i
tems_base=frozenset({'ALARM CLOCK BAKELIKE CHOCOLATE'}), items_add=frozenset({'ALA
RM CLOCK BAKELIKE RED'}), confidence=0.6459627329192547, lift=18.719634674032797
7)])
```

In [33]:

```
def display_rules(association_results2):
 for item in association_results2:
     pair = item[0]
     items = [x for x in pair]
     print("Rule: " + items[0] + " -> " + items[1])
     print("Support: " + str(item[1]))
     print("Confidence: " + str(item[2][0][2]))
     print("Lift: " + str(item[2][0][3]))
     print("=====================================")
```

In [34]:

```
display_rules(association_results2[:10])
```

```
Rule: ALARM CLOCK BAKELIKE CHOCOLATE -> ALARM CLOCK BAKELIKE RED
Support: 0.009225583252018096
Confidence: 0.6459627329192547
Lift: 18.719634674032797
=====================================
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE IVORY
Support: 0.010112658564712143
Confidence: 0.5377358490566038
Lift: 16.885504808955695
=====================================
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE RED
Support: 0.01969307194180786
Confidence: 0.6183844011142062
Lift: 17.920430215322487
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE RED
Support: 0.011398917768118513
Confidence: 0.6061320754716981
Lift: 17.56536474753844
=====================================
Rule: ALARM CLOCK BAKELIKE ORANGE -> ALARM CLOCK BAKELIKE RED
Support: 0.009713474673999823
Confidence: 0.6930379746835443
Lift: 20.083848556831867
=====================================
Rule: ALARM CLOCK BAKELIKE PINK -> ALARM CLOCK BAKELIKE RED
Support: 0.014459327596912978
Confidence: 0.576991150442478
Lift: 16.72087722092045
=====================================
Rule: BAKING SET SPACEBOY DESIGN -> BAKING SET 9 PIECE RETROSPOT
Support: 0.009092521955113989
Confidence: 0.597667638483965
Lift: 19.14064570633448
=====================================
Rule: BATHROOM METAL SIGN -> TOILET METAL SIGN
Support: 0.00966912090836512
Confidence: 0.6707692307692309
Lift: 30.675787174286167
=====================================
Rule: BLUE HARMONICA IN BOX -> RED  HARMONICA IN BOX
Support: 0.00966912090836512
Confidence: 0.5034642032332564
Lift: 22.611760808958163
=====================================
Rule: HOT WATER BOTTLE TEA AND SYMPATHY -> CHOCOLATE HOT WATER BOTTLE
Support: 0.0111327951743103
Confidence: 0.5432900432900434
Lift: 19.629835442335445
=====================================
```

Based on the ruleset above, the quality of the rules can be evaluated in terms of support, confidence, and lift values. Overall, the ruleset shows moderate to high quality.

For example, the rule "ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE IVORY" has a support of 0.010 and a confidence of 0.538. This indicates that the items "ALARM CLOCK BAKELIKE GREEN" and "ALARM CLOCK BAKELIKE IVORY" appear frequently together in the transactions, and when "ALARM CLOCK BAKELIKE GREEN" is present, there is a 53.8% chance that "ALARM CLOCK BAKELIKE IVORY" will also be present. Furthermore, the lift value of 16.89 indicates that the presence of "ALARM CLOCK BAKELIKE GREEN" has a strong positive effect on the occurrence of "ALARM CLOCK BAKELIKE IVORY".

We can use the product to further illustrate the quality of the ruleset. For example, the "ALARM CLOCK BAKELIKE" item is a retro-style alarm clock with a colorful bakelite case. The rule "ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE IVORY" suggests that customers who buy the green version of the alarm clock are also likely to buy the ivory version. This information can be useful for product recommendation and marketing strategies.

Similarly, the rule "CHOCOLATE HOT WATER BOTTLE -> HOT WATER BOTTLE TEA AND SYMPATHY" has a support of 0.011, a confidence of 0.543, and a lift of 19.63. This suggests that customers who buy the "CHOCOLATE HOT WATER BOTTLE" are likely to also buy the "HOT WATER BOTTLE TEA AND SYMPATHY". This information can be useful for cross-selling and upselling strategies.

- Setting 3: Min Support = 0.015, Min Confidence = 0.5, Min Lift = 9

In [35]:

```
association_rules3 = apriori(baskets, min_support=0.015, min_confidence=0.5,
                            min_lift=9, min_length=3)
association_results3 = list(association_rules3)
```

In [36]:

```
print('Rules generated: ', len(association_results3))
```

```
Rules generated:  11
```

In [37]:

```
print(association_results3[0])
```

```
RelationRecord(items=frozenset({'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIK
E RED'}), support=0.01969307194180786, ordered_statistics=[OrderedStatistic(items_
base=frozenset({'ALARM CLOCK BAKELIKE GREEN'}), items_add=frozenset({'ALARM CLOCK
BAKELIKE RED'}), confidence=0.6183844011142062, lift=17.920430215322487), OrderedS
tatistic(items_base=frozenset({'ALARM CLOCK BAKELIKE RED'}), items_add=frozenset
({'ALARM CLOCK BAKELIKE GREEN'}), confidence=0.5706940874035991, lift=17.920430215
322487)])
```

In [38]:

```
def display_rules(association_results3):
 for item in association_results3:
     pair = item[0]
     items = [x for x in pair]
     print("Rule: " + items[0] + " -> " + items[1])
     print("Support: " + str(item[1]))
     print("Confidence: " + str(item[2][0][2]))
     print("Lift: " + str(item[2][0][3]))
     print("=====================================")
```

In [39]:

```
display_rules(association_results3[:10])
```

```
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE RED
Support: 0.01969307194180786
Confidence: 0.6183844011142062
Lift: 17.920430215322487
=====================================
Rule: SPACEBOY LUNCH BOX -> DOLLY GIRL LUNCH BOX
Support: 0.015878648097223454
Confidence: 0.5830618892508143
Lift: 21.479923782759574
=====================================
Rule: GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA
Support: 0.01618912445666637
Confidence: 0.7115009746588694
Lift: 24.832044852413112
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER
Support: 0.020535793488867204
Confidence: 0.5816582914572863
Lift: 22.189624093394208
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER
Support: 0.02523729264614566
Confidence: 0.714824120603015
Lift: 19.726345927926044
=====================================
Rule: JUMBO  BAG BAROQUE BLACK WHITE -> JUMBO BAG RED RETROSPOT
Support: 0.01636653951920518
Confidence: 0.5738724727838258
Lift: 9.208917275006502
=====================================
Rule: JUMBO BAG STRAWBERRY -> JUMBO BAG RED RETROSPOT
Support: 0.015834294331588752
Confidence: 0.6176470588235294
Lift: 9.911366966715512
=====================================
Rule: ROSES REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER
Support: 0.019870487004346668
Confidence: 0.7580372250423011
Lift: 20.918858354716917
=====================================
Rule: RED RETROSPOT CHARLOTTE BAG -> WOODLAND CHARLOTTE BAG
Support: 0.015124634081433514
Confidence: 0.5711892797319933
Lift: 17.9609951197176
=====================================
Rule: WOODEN FRAME ANTIQUE WHITE -> WOODEN PICTURE FRAME WHITE FINISH
Support: 0.015479464206511133
Confidence: 0.5344563552833078
Lift: 16.46154779537904
=====================================
```

Support refers to the proportion of transactions in the dataset that contain both the antecedent and consequent of a rule. In the given ruleset, the support values range from 0.0151 to 0.0252, indicating that the rules are not particularly common in the dataset.

Confidence, on the other hand, measures the proportion of transactions containing the antecedent that also contain the consequent. The confidence values in the ruleset range from 0.534 to 0.758, indicating that the rules have a relatively high level of confidence.

Lift measures the increase in the likelihood of the consequent given the antecedent, compared to the likelihood of the consequent in the absence of the antecedent. The lift values in the ruleset range from 9.21 to 24.83, indicating that the rules are strongly associated with the consequent.

Overall, the ruleset suggests that there are several strong associations between products in the dataset, with high levels of confidence and lift. The highest lift value of 24.83 is seen in the rule "GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA", indicating a very strong association between these two products. However, it is worth noting that the support values for all the rules are relatively low, indicating that these associations are not particularly common in the dataset.

Comparing the rules in the dataset, we can see that the rules with the highest lift values also tend to have high confidence values. For example, the rules "GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA" and "ROSES REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER" have the highest lift values of 24.83 and 20.92, respectively, and also have high confidence values of 0.711 and 0.758, respectively. This suggests that these rules are particularly strong associations between the products.

In summary, the ruleset contains several strong associations between products with high confidence and lift values. However, the support values suggest that these associations are not particularly common in the dataset.

Q6. Filter the transactions on the 'day' of the week or on the 'month' to perform analysis on either of them on two durations. Generate association rules to discover if there are significant differences in the buying behaviour between chosen durations, and, discuss if the rules are useful.

In [40]:

```
data['Date']=pd.to_datetime(data['Date'])
```

In [41]:

```
data['Month'] =data['Date'].dt.month
data.head()
```

Out[41]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	iter
19246	564166	2011-08-23	13:40	NATURAL SLATE HEART CHALKBOARD	CHILDRENS APRON APPLES DESIGN	WRAP VINTAGE DOILY	WRAP VINTAGE LEAF DESIGN	WI POPI DESI
5135	543751	2011-11-02	14:00	WOODEN ROUNDERS GARDEN SET	WHITE BAMBOO RIBS LAMPSHADE	NaN	NaN	N
15906	559340	2011-07-07	16:51	METAL SIGN TAKE IT OR LEAVE IT	METAL SIGN EMPIRE TEA	HAND OVER THE CHOCOLATE SIGN	METAL SIGN HER DINNER IS SERVED	DOORN UNI JACK GU AND RO
10141	550943	2011-04-21	15:51	NaN	NaN	NaN	NaN	N
9075	549316	2011-08-04	09:54	JUMBO BAG WOODLAND ANIMALS	JUMBO STORAGE BAG SUKI	JUMBO BAG RED RETROSPOT	PACK OF 72 RETROSPOT CAKE CASES	F RETROSI MINI CA

5 rows × 46 columns

In [42]:

```
data['Month'].value_counts()
```

Out[42]:

```
11    2864
10    2249
9     2079
6     1979
7     1946
5     1850
8     1832
3     1762
1     1645
4     1591
2     1444
12    1305
Name: Month, dtype: int64
```

November has the highest transaction (2864), with October and September in second and third place consecutively. However, December has the least number of transactions (1305)

In [43]:

```
data.shape
```

Out[43]:

```
(22546, 46)
```

I will make Comparism between the highest month (November) and the first month of the year (January) while generating the association rules and making discussions wheter the rule changes across these to months

Extracting data for transactions made in November only

In [44]:
```
Filtered_data = data[data['Date'].dt.month == 11]
```

In [45]:
```
Filtered_data.head()
```

Out[45]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_
5135	543751	2011-11-02	14:00	WOODEN ROUNDERS GARDEN SET	WHITE BAMBOO RIBS LAMPSHADE	NaN	NaN	Na
5150	C543766	2011-11-02	15:18	CHRISTMAS LIGHTS 10 VINTAGE BAUBLES	NaN	NaN	NaN	Na
6975	546402	2011-11-03	15:22	DOORMAT UNION FLAG	DOORMAT FANCY FONT HOME SWEET HOME	DOORMAT SPOTTY HOME SWEET HOME	POLKADOT PEN	WHI JEWELLE HEA DECORATIO
30476	579694	2011-11-30	14:11	CHARLIE + LOLA RED HOT WATER BOTTLE	SET OF 20 KIDS COOKIE CUTTERS	SUNSET CHECK HAMMOCK	HANGING HEART ZINC T-LIGHT HOLDER	SET OF PINK HEA PAPE DOILI
9263	549588	2011-11-04	10:04	HOMEMADE JAM SCENTED CANDLES	SET OF 4 JAM JAR MAGNETS	TUMBLER, NEW ENGLAND	PINK KNITTED EGG COSY	BLU KNITTE EGG COS

5 rows × 46 columns

Data Preparation for Apriori Algorithm

November transactions will be represented by items_dfnov

In [46]:
```
items_dfnov=Filtered_data[Filtered_data.columns[3:44]]
```

In [47]:
```
items_dfnov.head()
```

Out[47]:

	item_0	item_1	item_2	item_3	item_4	item_5	item_6
5135	WOODEN ROUNDERS GARDEN SET	WHITE BAMBOO RIBS LAMPSHADE	NaN	NaN	NaN	NaN	NaN
5150	CHRISTMAS LIGHTS 10 VINTAGE BAUBLES	NaN	NaN	NaN	NaN	NaN	NaN
6975	DOORMAT UNION FLAG	DOORMAT FANCY FONT HOME SWEET HOME	DOORMAT SPOTTY HOME SWEET HOME	POLKADOT PEN	WHITE JEWELLED HEART DECORATION	NaN	NaN
30476	CHARLIE + LOLA RED HOT WATER BOTTLE	SET OF 20 KIDS COOKIE CUTTERS	SUNSET CHECK HAMMOCK	HANGING HEART ZINC T-LIGHT HOLDER	SET OF 72 PINK HEART PAPER DOILIES	15CM CHRISTMAS GLASS BALL 20 LIGHTS	TRADITIONAL CHRISTMAS RIBBONS
9263	HOMEMADE JAM SCENTED CANDLES	SET OF 4 JAM JAR MAGNETS	TUMBLER, NEW ENGLAND	PINK KNITTED EGG COSY	BLUE KNITTED EGG COSY	ENVELOPE 50 BLOSSOM IMAGES	TEA TIME TEA SET IN GIFT BOX

5 rows × 41 columns

Then we assign each november transaction to a list and finally we will have a list of lists for november transactions. To do that, we first take the Transpose of the dataframe, using .T and then drop the 'nan' values using the dropna() function.

In [48]:

```
basketsnov = items_dfnov.T.apply(lambda x: x.dropna().tolist()).tolist()
```

In [50]:

```
print(len(basketsnov))
```

```
2864
```

There are 2864 transactions in November, still a confirmation of Out[42]

In [51]:

```
for i in basketsnov[:5]:
 print(i)
```

```
['WOODEN ROUNDERS GARDEN SET', 'WHITE BAMBOO RIBS LAMPSHADE']
['CHRISTMAS LIGHTS 10 VINTAGE BAUBLES']
['DOORMAT UNION FLAG', 'DOORMAT FANCY FONT HOME SWEET HOME', 'DOORMAT SPOTTY HOME
SWEET HOME', 'POLKADOT PEN', 'WHITE JEWELLED HEART DECORATION']
['CHARLIE + LOLA RED HOT WATER BOTTLE', 'SET OF 20 KIDS COOKIE CUTTERS', 'SUNSET C
HECK HAMMOCK', 'HANGING HEART ZINC T-LIGHT HOLDER', 'SET OF 72 PINK HEART PAPER DO
ILIES', '15CM CHRISTMAS GLASS BALL 20 LIGHTS', 'TRADITIONAL CHRISTMAS RIBBONS', 'S
CANDINAVIAN REDS RIBBONS', 'JUMBO BAG RED RETROSPOT', 'JUMBO  BAG BAROQUE BLACK WH
ITE', 'RED ENCHANTED FOREST PLACEMAT', 'WHITE HANGING HEART T-LIGHT HOLDER', 'DOTC
OM POSTAGE']
['HOMEMADE JAM SCENTED CANDLES', 'SET OF 4 JAM JAR MAGNETS', 'TUMBLER, NEW ENGLAN
D', 'PINK KNITTED EGG COSY', 'BLUE KNITTED EGG COSY', 'ENVELOPE 50 BLOSSOM IMAGE
S', 'TEA TIME TEA SET IN GIFT BOX', 'SET/6 FRUIT SALAD  PAPER PLATES', 'SET/6 FRUI
T SALAD PAPER CUPS', 'LARGE PINK GLASS SUNDAE DISH', 'PINK FAIRY CAKE CHILDRENS AP
RON', 'CHERRY BLOSSOM DECORATIVE FLASK', 'CHERRY BLOSSOM  DECORATIVE FLASK', 'TEA
PARTY BIRTHDAY CARD', 'PENNY FARTHING BIRTHDAY CARD', 'CARD DOLLY GIRL', 'CARD PAR
TY GAMES', 'STARS GIFT TAPE', 'PAPER BUNTING VINTAGE PAISLEY', 'PINK PARTY BAGS',
'POSTAGE']
```

In [52]:

```
from apyori import apriori
```

We will use a baseline parameter of

- Minimum Support = 0.01
- Minimum Confidence = 0.2
- Minimum Lift = 3
- Minimum Length = 2

In [53]:

```
association_rulesnov = apriori(basketsnov, min_support=0.01, min_confidence=0.2,
                               min_lift=3, min_length=2)
association_resultsnov = list(association_rulesnov)
```

In [54]:

```
print('Rules generated: ', len(association_resultsnov))
```

```
Rules generated:  194
```

In [55]:

```
print(association_resultsnov[0])
```

```
RelationRecord(items=frozenset({"PAPER CHAIN KIT 50'S CHRISTMAS", '3 HEARTS HANGIN
G DECORATION RUSTIC'}), support=0.010125698324022346, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'3 HEARTS HANGING DECORATION RUSTIC'}), items_add=f
rozenset({"PAPER CHAIN KIT 50'S CHRISTMAS"}), confidence=0.3918918918918919, lift=
4.879905992949471)])
```

In [56]:

```
print(association_resultsnov[120])
```

```
RelationRecord(items=frozenset({'LUNCH BAG SUKI DESIGN', 'LUNCH BAG RED RETROSPO
T'}), support=0.0111731843575419, ordered_statistics=[OrderedStatistic(items_base=
frozenset({'LUNCH BAG RED RETROSPOT'}), items_add=frozenset({'LUNCH BAG SUKI DESIG
N'}), confidence=0.25396825396825395, lift=8.081834215167548), OrderedStatistic(it
ems_base=frozenset({'LUNCH BAG SUKI DESIGN'}), items_add=frozenset({'LUNCH BAG RED
RETROSPOT'}), confidence=0.35555555555555557, lift=8.081834215167548)])
```

In [57]:

```
def display_rules(association_resultsnov):
 for item in association_resultsnov:
     pair = item[0]
     items = [x for x in pair]
     print("Rule: " + items[0] + " -> " + items[1])
     print("Support: " + str(item[1]))
     print("Confidence: " + str(item[2][0][2]))
     print("Lift: " + str(item[2][0][3]))
     print("=====================================")
```

In [58]:

```
display_rules(association_resultsnov[:10])
```

```
Rule: PAPER CHAIN KIT 50'S CHRISTMAS -> 3 HEARTS HANGING DECORATION RUSTIC
Support: 0.010125698324022346
Confidence: 0.3918918918918919
Lift: 4.879905992949471
=====================================
Rule: 6 GIFT TAGS VINTAGE CHRISTMAS -> 6 GIFT TAGS 50'S CHRISTMAS
Support: 0.01047486033519553
Confidence: 0.3896103896103896
Lift: 18.29252714498616
=====================================
Rule: ROLL WRAP 50'S CHRISTMAS -> 6 GIFT TAGS 50'S CHRISTMAS
Support: 0.015712290502793297
Confidence: 0.5844155844155845
Lift: 19.23869234214062
=====================================
Rule: ROLL WRAP 50'S RED CHRISTMAS -> 6 GIFT TAGS 50'S CHRISTMAS
Support: 0.0111731843575419
Confidence: 0.4155844155844156
Lift: 22.041366041366043
=====================================
Rule: PAPER CHAIN KIT 50'S CHRISTMAS -> 6 RIBBONS ELEGANT CHRISTMAS
Support: 0.010125698324022346
Confidence: 0.6041666666666666
Lift: 7.5231884057971
=====================================
Rule: PAPER CHAIN KIT 50'S CHRISTMAS -> 60 CAKE CASES VINTAGE CHRISTMAS
Support: 0.010824022346368716
Confidence: 0.28971962616822433
Lift: 3.6076391710686715
=====================================
Rule: PAPER CHAIN KIT VINTAGE CHRISTMAS -> 60 CAKE CASES VINTAGE CHRISTMAS
Support: 0.013966480446927373
Confidence: 0.37383177570093457
Lift: 6.907446487790171
=====================================
Rule: SET OF 20 VINTAGE CHRISTMAS NAPKINS -> 60 CAKE CASES VINTAGE CHRISTMAS
Support: 0.010125698324022346
Confidence: 0.27102803738317754
Lift: 7.322870745900193
=====================================
Rule: 60 TEATIME FAIRY CAKE CASES -> PACK OF 60 PINK PAISLEY CAKE CASES
Support: 0.010125698324022346
Confidence: 0.38666666666666666
Lift: 16.52855721393035
=====================================
Rule: ADVENT CALENDAR GINGHAM SACK -> WOODEN STAR CHRISTMAS SCANDINAVIAN
Support: 0.010125698324022346
Confidence: 0.48333333333333334
Lift: 12.037101449275362
=====================================
```

With minumum parameter of support and confidence as 1% and 20% respectively, 194 rules are generated within the parameter's threshold. It also shows the goods that are mostly bought together in November, also it shows that most of the products are Christmas items, confidence level is quite low as well with the highest been PAPER CHAIN KIT 50'S CHRISTMAS -> 6 RIBBONS ELEGANT CHRISTMAS at 60%

Using different parameter to check the association rules changes

Min Support = 0.02, Min Confidence = 0.2, Min Lift = 6

In [59]:
```
association_rulesnov = apriori(basketsnov, min_support=0.02, min_confidence=0.2,
                            min_lift=6, min_length=2)
association_resultsnov = list(association_rulesnov)
```

In [60]:
```
print('Rules generated: ', len(association_resultsnov))
```

```
Rules generated:  11
```

In [61]:
```
print(association_resultsnov[0])
```

```
RelationRecord(items=frozenset({'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIK
E RED'}), support=0.02164804469273743, ordered_statistics=[OrderedStatistic(items_
base=frozenset({'ALARM CLOCK BAKELIKE GREEN'}), items_add=frozenset({'ALARM CLOCK
BAKELIKE RED'}), confidence=0.6888888888888889, lift=18.970940170940167), OrderedS
tatistic(items_base=frozenset({'ALARM CLOCK BAKELIKE RED'}), items_add=frozenset
({'ALARM CLOCK BAKELIKE GREEN'}), confidence=0.5961538461538461, lift=18.970940170
94017)])
```

In [62]:
```
print(association_resultsnov[10])
```

```
RelationRecord(items=frozenset({'WOODEN HEART CHRISTMAS SCANDINAVIAN', 'WOODEN TRE
E CHRISTMAS SCANDINAVIAN'}), support=0.02164804469273743, ordered_statistics=[Orde
redStatistic(items_base=frozenset({'WOODEN HEART CHRISTMAS SCANDINAVIAN'}), items_
add=frozenset({'WOODEN TREE CHRISTMAS SCANDINAVIAN'}), confidence=0.50406504065040
66, lift=20.33298980877133), OrderedStatistic(items_base=frozenset({'WOODEN TREE C
HRISTMAS SCANDINAVIAN'}), items_add=frozenset({'WOODEN HEART CHRISTMAS SCANDINAVIA
N'}), confidence=0.8732394366197183, lift=20.332989808771327)])
```

In [63]:
```
def display_rules(association_resultsnov):
 for item in association_resultsnov:
     pair = item[0]
     items = [x for x in pair]
     print("Rule: " + items[0] + " -> " + items[1])
     print("Support: " + str(item[1]))
     print("Confidence: " + str(item[2][0][2]))
     print("Lift: " + str(item[2][0][3]))
     print("=====================================")
```

In [64]:
```
display_rules(association_resultsnov[:10])
```

```
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE RED
Support: 0.021648044692737743
Confidence: 0.6888888888888889
Lift: 18.970940170940167
=====================================
Rule: HOT WATER BOTTLE TEA AND SYMPATHY -> CHOCOLATE HOT WATER BOTTLE
Support: 0.023044692737430168
Confidence: 0.5739130434782608
Lift: 13.697391304347825
=====================================
Rule: CHRISTMAS CRAFT LITTLE FRIENDS -> CHRISTMAS CRAFT WHITE FAIRY
Support: 0.021298882681564244
Confidence: 0.5083333333333333
Lift: 15.008934707903778
=====================================
Rule: GARDENERS KNEELING PAD KEEP CALM -> GARDENERS KNEELING PAD CUP OF TEA
Support: 0.026536312849162011
Confidence: 0.6551724137931034
Lift: 13.402955665024631
=====================================
Rule: GREEN REGENCY TEACUP AND SAUCER -> PINK REGENCY TEACUP AND SAUCER
Support: 0.020251396648044692
Confidence: 0.7341772151898734
Lift: 29.615261187377428
=====================================
Rule: HAND WARMER SCOTTY DOG DESIGN -> HAND WARMER BIRD DESIGN
Support: 0.021298882681564244
Confidence: 0.6703296703296703
Lift: 18.63906966819588
=====================================
Rule: LUNCH BAG  BLACK SKULL -> LUNCH BAG RED RETROSPOT
Support: 0.020600558659217876
Confidence: 0.5175438596491228
Lift: 11.76385407964355
=====================================
Rule: PAPER CHAIN KIT 50'S CHRISTMAS -> PAPER CHAIN KIT VINTAGE CHRISTMAS
Support: 0.028980446927374302
Confidence: 0.3608695652173913
Lift: 6.667938288920055
=====================================
Rule: RED HANGING HEART T-LIGHT HOLDER -> WHITE HANGING HEART T-LIGHT HOLDER
Support: 0.020600558659217876
Confidence: 0.6413043478260869
Lift: 10.93271221532091
=====================================
Rule: WOODEN HEART CHRISTMAS SCANDINAVIAN -> WOODEN STAR CHRISTMAS SCANDINAVIAN
Support: 0.03247206703910614
Confidence: 0.7560975609756097
Lift: 18.830116648992572
=====================================
```

Using a different parameter increasing minimum support from 1% to 2% leads to a massive drop in the rules. It goes from 194 to 11 association rules. Other rulesets that dont match the parameter's threshold has been discarded.

In [67]:
```
from collections import Counter
counter = Counter(basketsnov[0])
for i in basketsnov[1:]:
    if i != 'nan':
        counter.update(i)
del counter['nan']
counter.most_common(10)
```

Out[67]:

```
[('RABBIT NIGHT LIGHT', 250),
 ("PAPER CHAIN KIT 50'S CHRISTMAS", 235),
 ('WHITE HANGING HEART T-LIGHT HOLDER', 173),
 ('HOT WATER BOTTLE KEEP CALM', 161),
 ('PAPER CHAIN KIT VINTAGE CHRISTMAS', 156),
 ('BAKING SET 9 PIECE RETROSPOT', 156),
 ('JUMBO BAG RED RETROSPOT', 148),
 ('REGENCY CAKESTAND 3 TIER', 147),
 ('JUMBO BAG PAISLEY PARK', 143),
 ('GARDENERS KNEELING PAD KEEP CALM', 143)]
```

The list in Out[67] lists the top 10 most common items purchased in November and they are mostly christmas items.

Extracting for January

Extracting data for transactions made in January only using Filtered_data2.head()

In [68]:

```python
Filtered_data2 = data[data['Date'].dt.month == 1]
Filtered_data2.head()
```

Out[68]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item
4393	542633	2011-01-31	10:31	ZINC FOLKART SLEIGH BELLS	TEA TIME OVEN GLOVE	PARTY BUNTING	DOORMAT FRIENDSHIP	RED WOOL HOT WHI HEA
4051	542149	2011-01-26	10:17	PAPER CHAIN KIT EMPIRE	ENGLISH ROSE GARDEN SECATEURS	DINOSAURS WRITING SET	DINOSAUR PARTY BAG + STICKER SET	FOUR HO WHI LOVEBIR
3722	541696	2011-01-20	18:08	BLUE EGG SPOON	12 IVORY ROSE PEG PLACE SETTINGS	12 DAISY PEGS IN WOOD BOX	TOADSTOOL MONEY BOX	CHICK GR HOT WAT BOTT
3729	541696	2011-01-20	18:08	HANGING WOOD AND FELT FLOWER	SET/3 POLKADOT STACKING TINS	GREEN VINTAGE EARRINGS	SILVER M.O.P ORBIT DROP EARRINGS	TI CRYST BRACEL GRE
17796	562024	2011-01-08	16:07	BROWN CHECK CAT DOORSTOP	PAINTED PINK RABBIT	STRAWBERRY RAFFIA FOOD COVER	FANCY FONT BIRTHDAY CARD	STRAWBER LUNCH B WI CUTLE

5 rows × 46 columns

Data Preparation for Apriori Algorithm

Januray transactions will be represented by items_dfjan

In [69]:

```python
items_dfjan=Filtered_data2[Filtered_data2.columns[3:44]]
```

In [70]:
```
items_dfjan.head()
```

Out[70]:

	item_0	item_1	item_2	item_3	item_4	item_5	item_6
4393	ZINC FOLKART SLEIGH BELLS	TEA TIME OVEN GLOVE	PARTY BUNTING	DOORMAT FRIENDSHIP	RED WOOLLY HOTTIE WHITE HEART	KNITTED UNION FLAG HOT WATER BOTTLE	ENGLISH ROSE HOT WATER BOTTLE
4051	PAPER CHAIN KIT EMPIRE	ENGLISH ROSE GARDEN SECATEURS	DINOSAURS WRITING SET	DINOSAUR PARTY BAG + STICKER SET	FOUR HOOK WHITE LOVEBIRDS	HEART IVORY TRELLIS LARGE	PACK OF 6 SANDCASTLE FLAGS ASSORTED
3722	BLUE EGG SPOON	12 IVORY ROSE PEG PLACE SETTINGS	12 DAISY PEGS IN WOOD BOX	TOADSTOOL MONEY BOX	CHICK GREY HOT WATER BOTTLE	ROTATING LEAVES T-LIGHT HOLDER	SKULLS GREETING CARD
3729	HANGING WOOD AND FELT FLOWER	SET/3 POLKADOT STACKING TINS	GREEN VINTAGE EARRINGS	SILVER M.O.P ORBIT DROP EARRINGS	TINY CRYSTAL BRACELET GREEN	CARNIVAL BRACELET	FRUIT SALAD BAG CHARM
17796	BROWN CHECK CAT DOORSTOP	PAINTED PINK RABBIT	STRAWBERRY RAFFIA FOOD COVER	FANCY FONT BIRTHDAY CARD	STRAWBERRY LUNCH BOX WITH CUTLERY	VINTAGE UNION JACK BUNTING	MOCK LOBSTER FRIDGE MAGNET

5 rows × 41 columns

In [71]:
```
basketsjan = items_dfjan.T.apply(lambda x: x.dropna().tolist()).tolist()
```

In [72]:
```
print(len(basketsjan))
```

```
1645
```

There are 1645 transactions in January

In [74]:
```
for i in basketsjan[:5]:
    print(i)
```

```
['ZINC FOLKART SLEIGH BELLS', 'TEA TIME OVEN GLOVE', 'PARTY BUNTING', 'DOORMAT FRI
ENDSHIP', 'RED WOOLLY HOTTIE WHITE HEART', 'KNITTED UNION FLAG HOT WATER BOTTLE',
'ENGLISH ROSE HOT WATER BOTTLE', 'CHARLIE+LOLA RED HOT WATER BOTTLE', 'SET OF 3 BU
TTERFLY COOKIE CUTTERS', 'BOX OF 24 COCKTAIL PARASOLS', '60 TEATIME FAIRY CAKE CAS
ES', 'PINK 3 PIECE POLKADOT CUTLERY SET']
['PAPER CHAIN KIT EMPIRE', 'ENGLISH ROSE GARDEN SECATEURS', 'DINOSAURS  WRITING SE
T', 'DINOSAUR PARTY BAG + STICKER SET', 'FOUR HOOK  WHITE LOVEBIRDS', 'HEART IVORY
TRELLIS LARGE', 'PACK OF 6 SANDCASTLE FLAGS ASSORTED', '72 SWEETHEART FAIRY CAKE C
ASES', 'SMALL POPCORN HOLDER', 'HEART OF WICKER SMALL']
['BLUE EGG  SPOON', '12 IVORY ROSE PEG PLACE SETTINGS', '12 DAISY PEGS IN WOOD BO
X', 'TOADSTOOL MONEY BOX', 'CHICK GREY HOT WATER BOTTLE', 'ROTATING LEAVES T-LIGHT
HOLDER', 'SKULLS GREETING CARD', 'FANCY FONT BIRTHDAY CARD', 'RED RETROSPOT PUDDIN
G BOWL', 'CERAMIC STRAWBERRY TRINKET TRAY', 'CERAMIC STRAWBERRY MONEY BOX', 'SET O
F 6 FUNKY BEAKERS', '4 PEAR BOTANICAL DINNER CANDLES', 'FLOWERS  STICKERS', 'FOLDI
NG UMBRELLA RED/WHITE POLKADOT', 'FOLDING UMBRELLA CREAM POLKADOT', 'FOLDING UMBRE
LLA CHOCOLATE POLKADOT', 'LOVE BUILDING BLOCK WORD', 'DECORATIVE ROSE BATHROOM BOT
TLE', 'DECORATIVE CATS BATHROOM BOTTLE', 'CHRISTMAS TREE DECORATION WITH BELL', 'C
HRISTMAS TREE HEART DECORATION', 'CHRISTMAS TREE STAR DECORATION', 'CHOCOLATE CALC
ULATOR', 'PINK DIAMANTE PEN IN GIFT BOX', 'BLUE  DIAMANTE PEN IN GIFT BOX', 'LILAC
DIAMANTE PEN IN GIFT BOX', 'HEARTS GIFT TAPE', 'SKULLS TAPE', 'STARS GIFT TAPE',
'CAKES AND BOWS GIFT  TAPE', 'TRADITIONAL WOODEN CATCH CUP GAME', 'CHOCOLATE THIS
WAY METAL SIGN', 'VINTAGE SEASIDE JIGSAW PUZZLES', 'SET 12 RETRO WHITE CHALK STICK
S', 'JUMBO BAG PINK VINTAGE PAISLEY', 'JUMBO STORAGE BAG SUKI', 'SKULL SHOULDER BA
G', 'SUKI  SHOULDER BAG', 'FRYING PAN PINK POLKADOT', 'FRYING PAN BLUE POLKADOT']
['HANGING WOOD AND FELT FLOWER', 'SET/3 POLKADOT STACKING TINS', 'GREEN VINTAGE EA
RRINGS', 'SILVER M.O.P ORBIT DROP EARRINGS', 'TINY CRYSTAL BRACELET GREEN', 'CARNI
VAL BRACELET', 'FRUIT SALAD BAG CHARM', 'ANT COPPER LIME BOUDICCA BRACELET', 'AMET
HYST HOOP EARRING FLORAL LEAF', 'BLUE SWEETHEART BRACELET', '4 VANILLA BOTANICAL C
ANDLES', 'SET OF 4 ROSE BOTANICAL CANDLES', 'HIPPY CHIC DECORATIVE PARASOL', 'ASSO
RTED TUTTI FRUTTI BRACELET', 'GLASS  BEURRE DISH', 'BLUE STRIPE CERAMIC DRAWER KNO
B', 'WHITE SPOT BLUE CERAMIC DRAWER KNOB', 'BUTTERFLIES STICKERS', 'MINI FUNKY DES
IGN TAPES', 'COLOURING PENCILS BROWN TUBE', 'RATTLE SNAKE EGGS', 'RECYCLED PENCIL
WITH RABBIT ERASER', 'SLEEPING CAT ERASERS', 'ORIGAMI SANDLEWOOD INCENSE/CAND SE
T', 'BLUE POLKADOT LUGGAGE TAG', 'CHERRY BLOSSOM LUGGAGE TAG', 'MOUSEY LONG LEGS S
OFT TOY', 'JUMBO BAG WOODLAND ANIMALS', 'STRAWBERRY SHOPPER BAG', 'WOODLAND CHARLO
TTE BAG', 'STRAWBERRY CHARLOTTE BAG', 'RED RETROSPOT CHARLOTTE BAG', 'LUNCH BAG RE
D RETROSPOT', 'LARGE PINK GLASS SUNDAE DISH', 'BLUE PATCH PURSE PINK HEART', 'DENI
M PATCH PURSE PINK BUTTERFLY', 'PURPLE ENAMEL FLOWER HAIR TIE', 'WHITE ENAMEL FLOW
ER HAIR TIE', 'PINK ENAMEL FLOWER HAIR TIE', '"LETTER ""A"" BLING KEY RING"', 'DOT
COM POSTAGE']
['BROWN CHECK CAT DOORSTOP', 'PAINTED PINK RABBIT', 'STRAWBERRY RAFFIA FOOD COVE
R', 'FANCY FONT BIRTHDAY CARD', 'STRAWBERRY LUNCH BOX WITH CUTLERY', 'VINTAGE UNIO
N JACK BUNTING', 'MOCK LOBSTER FRIDGE MAGNET', 'GIRLS VINTAGE TIN SEASIDE BUCKET',
'BOYS VINTAGE TIN SEASIDE BUCKET', 'RED METAL BEACH SPADE', 'LARGE RED RETROSPOT W
INDMILL', 'GIN AND TONIC MUG', 'POTTERING MUG', 'BINGO SET', 'MORE BUTTER METAL SI
GN', 'BLUE HARMONICA IN BOX', 'SET 12 KIDS COLOUR  CHALK STICKS', 'JUMBO BAG SCAND
INAVIAN BLUE PAISLEY', 'JUMBO STORAGE BAG SKULLS', 'RED RETROSPOT PICNIC BAG', 'PA
CK OF 60 PINK PAISLEY CAKE CASES', 'PACK OF 6 SKULL PAPER CUPS', 'PACK OF 6 SKULL
PAPER PLATES', 'MODERN FLORAL STATIONERY SET', 'BOHEMIAN COLLAGE STATIONERY SET',
'VINTAGE PAISLEY STATIONERY SET', 'FLORAL FOLK STATIONERY SET', 'BOUDOIR SQUARE TI
SSUE BOX', 'METAL SIGN HIS DINNER IS SERVED', 'SET OF 2 TEA TOWELS PING MICROWAV
E', 'MINI LADLE LOVE HEART PINK', 'BAKING SET 9 PIECE RETROSPOT', 'HEART DECORATIO
N RUSTIC HANGING', 'METAL 4 HOOK HANGER FRENCH CHATEAU', 'CAKE STAND VICTORIAN FIL
IGREE SMALL', 'HANGING HEART MIRROR DECORATION', 'EGG CUP NATURAL CHICKEN', 'HEART
IVORY TRELLIS SMALL', '200 RED + WHITE BENDY STRAWS', 'HOME GARLAND PAINTED ZINC',
'TEA BAG PLATE RED RETROSPOT']
```

Using the same parameters for the algorithm, as used for November transactions

Minimum Support = 0.01 Minimum Confidence = 0.2 Minimum Lift = 3

In [77]:
```
association_rulesjan = apriori(basketsjan, min_support=0.01, min_confidence=0.2,
                            min_lift=3, min_length=2)
association_resultsjan = list(association_rulesjan)
```

In [78]:
```
print('Rules generated: ', len(association_resultsjan))
```

```
Rules generated:  91
```

In [79]:
```
print(association_resultsjan[0])
```

```
RelationRecord(items=frozenset({'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIK
E RED'}), support=0.013373860182370821, ordered_statistics=[OrderedStatistic(items
_base=frozenset({'ALARM CLOCK BAKELIKE GREEN'}), items_add=frozenset({'ALARM CLOCK
BAKELIKE RED'}), confidence=0.55, lift=25.131944444444446), OrderedStatistic(items
_base=frozenset({'ALARM CLOCK BAKELIKE RED'}), items_add=frozenset({'ALARM CLOCK B
AKELIKE GREEN'}), confidence=0.6111111111111112, lift=25.131944444444446)])
```

In [82]:
```
def display_rules(association_resultsjan):
    for item in association_resultsjan:
        pair = item[0]
        items = [x for x in pair]
        print("Rule: " + items[0] + " -> " + items[1])
        print("Support: " + str(item[1]))
        print("Confidence: " + str(item[2][0][2]))
        print("Lift: " + str(item[2][0][3]))
        print("=====================================")
```

In [83]:
```
display_rules(association_resultsjan[:10])
```

```
Rule: ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE RED
Support: 0.013373860182370821
Confidence: 0.55
Lift: 25.131944444444446
=====================================
Rule: REGENCY CAKESTAND 3 TIER -> ASSORTED COLOUR BIRD ORNAMENT
Support: 0.011550151975683891
Confidence: 0.27941176470588236
Lift: 4.255855119825708
=====================================
Rule: BAKING SET SPACEBOY DESIGN -> BAKING SET 9 PIECE RETROSPOT
Support: 0.0121580547112462
Confidence: 0.37037037037037035
Lift: 21.759259259259256
=====================================
Rule: BATHROOM METAL SIGN -> TOILET METAL SIGN
Support: 0.012765957446808851
Confidence: 0.4666666666666666
Lift: 25.58888888888889
=====================================
Rule: PINK 3 PIECE POLKADOT CUTLERY SET -> BLUE 3 PIECE POLKADOT CUTLERY SET
Support: 0.015197568389057751
Confidence: 0.78125
Lift: 40.1611328125
=====================================
Rule: PINK HAPPY BIRTHDAY BUNTING -> BLUE HAPPY BIRTHDAY BUNTING
Support: 0.0121580547112462
Confidence: 0.7692307692307692
Lift: 45.192307692307686
=====================================
Rule: BLUE POLKADOT BOWL -> RED RETROSPOT BOWL
Support: 0.01094224924012158
Confidence: 0.8999999999999999
Lift: 41.125
=====================================
Rule: RED HANGING HEART T-LIGHT HOLDER -> CANDLEHOLDER PINK HANGING HEART
Support: 0.01094224924012158
Confidence: 0.5
Lift: 13.940677966101694
=====================================
Rule: CANDLEHOLDER PINK HANGING HEART -> WHITE HANGING HEART T-LIGHT HOLDER
Support: 0.011550151975683891
Confidence: 0.5277777777777778
Lift: 7.234953703703704
=====================================
Rule: JUMBO BAG PINK POLKADOT -> CHARLOTTE BAG PINK POLKADOT
Support: 0.011550151975683891
Confidence: 0.48717948717948717
Lift: 11.961347110600842
=====================================
```

There are 91 rules satisfying the Minimunm support and Mininimum confidence used in the algorithm.

In [85]:

```
from collections import Counter
counter = Counter(basketsjan[0])
for i in basketsjan[1:]:
    if i != 'nan':
        counter.update(i)
del counter['nan']
counter.most_common(10)
```

Out[85]:

```
[('WHITE HANGING HEART T-LIGHT HOLDER', 120),
 ('REGENCY CAKESTAND 3 TIER', 110),
 ('HEART OF WICKER SMALL', 92),
 ('NATURAL SLATE HEART CHALKBOARD', 89),
 ('LUNCH BAG  BLACK SKULL', 87),
 ('JUMBO BAG RED RETROSPOT', 85),
 ('LUNCH BAG RED RETROSPOT', 83),
 ('JAM MAKING SET WITH JARS', 83),
 ('SET OF 3 CAKE TINS PANTRY DESIGN', 80),
 ('SET OF 3 HEART COOKIE CUTTERS', 74)]
```

These are the top 10 items in January

Using different parameter to check the association rules changes

Min Support = 0.02, Min Confidence = 0.2, Min Lift = 6

In [86]:

```
association_rulesjan = apriori(basketsjan, min_support=0.02, min_confidence=0.2,
                                min_lift=6, min_length=2)
association_resultsjan = list(association_rulesjan)
```

In [87]:

```
print('Rules generated: ', len(association_resultsjan))
```

```
Rules generated:  3
```

In [88]:

```
print(association_resultsjan[0])
```

```
RelationRecord(items=frozenset({'GREEN REGENCY TEACUP AND SAUCER', 'ROSES REGENCY
TEACUP AND SAUCER'}), support=0.02370820668693009, ordered_statistics=[OrderedStat
istic(items_base=frozenset({'GREEN REGENCY TEACUP AND SAUCER'}), items_add=frozens
et({'ROSES REGENCY TEACUP AND SAUCER'}), confidence=0.7222222222222222, lift=21.60
1010101010104), OrderedStatistic(items_base=frozenset({'ROSES REGENCY TEACUP AND S
AUCER'}), items_add=frozenset({'GREEN REGENCY TEACUP AND SAUCER'}), confidence=0.7
090909090909091, lift=21.6010101010101)])
```

In [89]:

```
def display_rules(association_resultsjan):
    for item in association_resultsjan:
        pair = item[0]
        items = [x for x in pair]
        print("Rule: " + items[0] + " -> " + items[1])
        print("Support: " + str(item[1]))
        print("Confidence: " + str(item[2][0][2]))
        print("Lift: " + str(item[2][0][3]))
        print("=====================================")
```

In [90]:

```
display_rules(association_resultsjan[:3])
```

```
Rule: GREEN REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER
Support: 0.02370820668693009
Confidence: 0.7222222222222222
Lift: 21.601010101010104
=====================================
Rule: HEART OF WICKER LARGE -> HEART OF WICKER SMALL
Support: 0.020668693009118544
Confidence: 0.5573770491803278
Lift: 9.96614397719173
=====================================
Rule: RED HANGING HEART T-LIGHT HOLDER -> WHITE HANGING HEART T-LIGHT HOLDER
Support: 0.022492401215805473
Confidence: 0.6271186440677966
Lift: 8.596751412429379
=====================================
```

There is a great drop in rules conforming to the parameter threshold as we are reduced to just 3 association rules which indicates a really strong ruleset. GREEN REGENCY TEACUP AND SAUCER -> ROSES REGENCY TEACUP AND SAUCER seems to be quite bought together as the confidence is above 72% with support around 0.023.

The changes of the association rules with different months

The analysis indicates that the association rules vary across different months. With a minimum support of 0.01, minimum confidence of 0.2, and minimum lift of 3, there were 194 association rules generated for the transactions in November and 91 association rules generated for the transactions in January. As November is a period where people tend to shop for winter items, Christmas decorations, and gift items, it is expected that certain items will be frequently bought together. Therefore, the algorithm identifies all frequent itemsets that meet the minimum support threshold in November and generates rules that satisfy the minimum support and minimum confidence criteria.

In contrast, the number of transactions in January is typically lower compared to other months. This could be attributed to various factors, such as the end of the holiday season, people focusing on paying off debts accumulated during the holiday season, or the colder weather making it less appealing to go out shopping. As a result, the association rules generated in January were larger itemsets that meet the minimum support and minimum confidence thresholds.

However, when the minimum support was increased to 0.02 and minimum confidence remained at 0.2, the number of rules generated decreased significantly, with only 3 rules generated in January and 11 in November. This is due to the fact that fewer large itemsets meet the minimum support threshold, resulting in a reduction in the number of rules generated.

REFERENCES

Han, J., Kamber, M. and Pei, J., 2012. Data Mining Concepts and Techinques. 3rd ed. Amsterdam: Elsevier, p.246.