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('C:\\Users\\User\\Downloads\\Workshop-5-dataset.zip', sep='\t',dtype=np.str)
```

Q1. Print the first five rows of the dataset.

```
In [3]: df.head()
```

Out[3]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4	item_5	item_6	...	item_31	item_32
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	SET 7 BABUSHKA NESTING BOXES	GLASS STAR FROSTED T-LIGHT HOLDER	...	NaN	NaN
1	536366	01/12/2010	08:28	HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT	NaN	NaN	NaN	NaN	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	BOX OF 6 ASSORTED COLOUR TEASPOONS	BOX OF VINTAGE JIGSAW BLOCKS	...	NaN	NaN
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	NaN	NaN	...	NaN	NaN
4	536369	01/12/2010	08:35	BATH BUILDING BLOCK WORD	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

5 rows × 44 columns

In [4]:

```
df.head()
```

Out[4]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4	item_5	item_6	...	item_31	item_32
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	SET 7 BABUSHKA NESTING BOXES	GLASS STAR FROSTED T-LIGHT HOLDER	...	NaN	NaN
1	536366	01/12/2010	08:28	HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT	NaN	NaN	NaN	NaN	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	BOX OF 6 ASSORTED COLOUR TEASPOONS	BOX OF VINTAGE JIGSAW BLOCKS	...	NaN	NaN
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	NaN	NaN	...	NaN	NaN
4	536369	01/12/2010	08:35	BATH BUILDING BLOCK WORD	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

5 rows × 44 columns

In [5]:

```
df.shape
```

Out[5]:

```
(31941, 44)
```

Q2. How many rows and columns are there in the dataset? From output generated by the code in line 5, the data has 31941 rows and 44 columns.

In [6]:

```
STUDENT_NAME = 'MoradekeAdeleye'
STUDENT_NO = '9810'
```

In [7]:

```
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 [8]:

```
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. From the output in line 9 below, there are 305 unique dates in the data set.

In [9]:

```
data['Date'].nunique()
```

Out[9]:

```
305
```

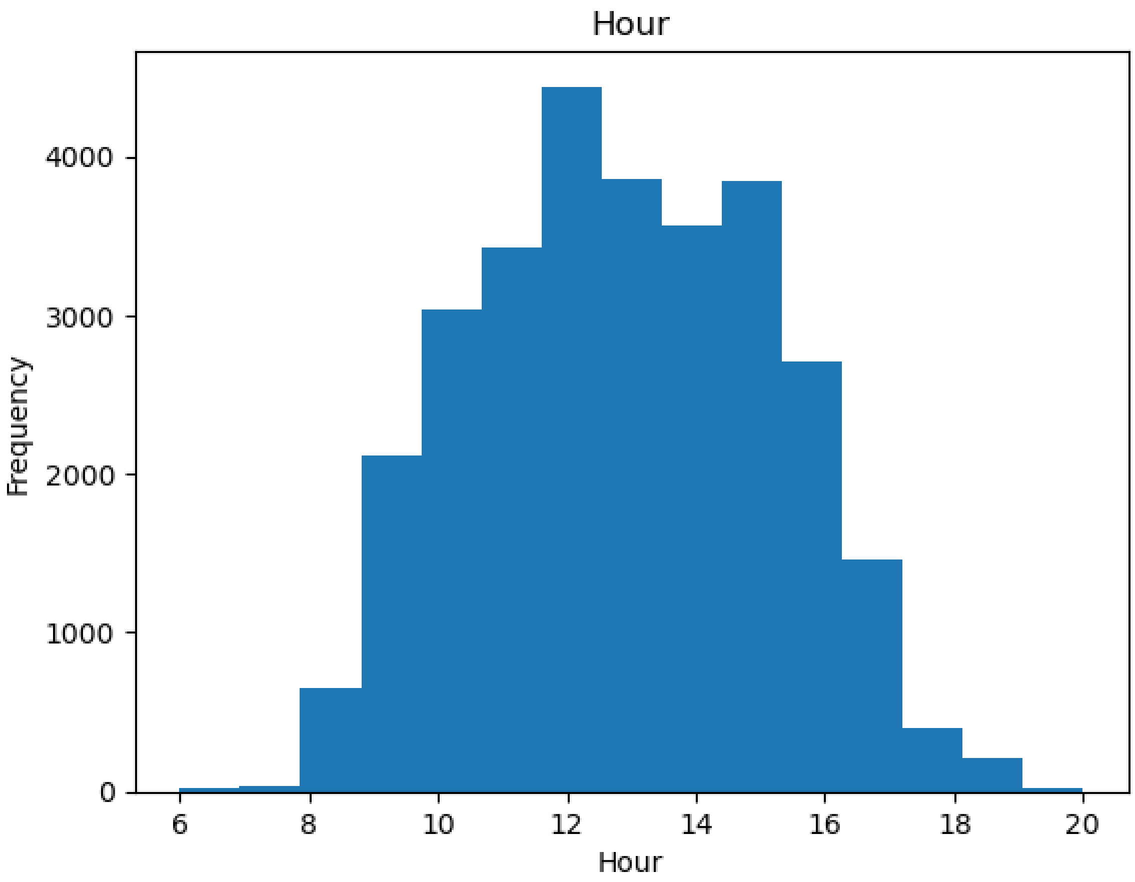
In [10]:

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

In [11]:

```
hour_hist = data.hist(column="Hour", bins=15, grid=False)

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



In [12]:

```
!pip install apyori
```

```
Requirement already satisfied: apyori in c:\users\user\anaconda3\lib\site-packages (1.1.2)
```

In [13]:

```python
from apyori import apriori
```

In [14]:

```python
data.head(1)
```

Out[14]:

	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4	item_5	item_6	...	item_32	item_33	i
30381	579520	29/11/2011	18:14	ROUND CAKE TIN VINTAGE RED	VINTAGE SNAP CARDS	NURSERY A,B,C PAINTED LETTERS	GOLD MUG BONE CHINA TREE OF LIFE	3 ROSE MORRIS BOXED CANDLES	3 WHITE CHOC MORRIS BOXED CANDLES	FIVE HEART HANGING DECORATION	...	PINK OWL SOFT TOY	SPOTTY PINK DUCK DOORSTOP	W AN

1 rows × 45 columns

In [15]:

```python
items_df=data[data.columns[3:44]]
```

In [16]:
```
items_df.head()
```

Out[16]:

	item_0	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	iter
30381	ROUND CAKE TIN VINTAGE RED	VINTAGE SNAP CARDS	NURSERY A,B,C PAINTED LETTERS	GOLD MUG BONE CHINA TREE OF LIFE	3 ROSE MORRIS BOXED CANDLES	3 WHITE CHOC MORRIS BOXED CANDLES	FIVE HEART HANGING DECORATION	GIANT 50'S CHRISTMAS CRACKER	GUMBALL COAT RACK	FREN CARRIA LANTE
16787	Manual	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
20582	ENGLISH ROSE NOTEBOOK A7 SIZE	HANGING WOOD AND FELT BUTTERFLY	PACK OF 12 VINTAGE DOILY TISSUES	RED RETROSPOT CHARLOTTE BAG	CHARLOTTE BAG VINTAGE ALPHABET	ASS FLORAL PRINT MULTI SCREWDRIVER	PACK OF 12 RED APPLE TISSUES	PACK OF 12 LONDON TISSUES	PACK OF 12 SKULL TISSUES	MAGNE PACK C SWALLO
15891	CAKE PLATE LOVEBIRD WHITE	HANGING HEART MIRROR DECORATION	BUNNY WOODEN PAINTED WITH BIRD	JIGSAW TREE WITH WATERING CAN	JIGSAW TREE WITH BIRDHOUSE	JIGSAW RABBIT AND BIRDHOUSE	3 HOOK HANGER MAGIC GARDEN	BUNNY DECORATION MAGIC GARDEN	DECORATION BUTTERFLY MAGIC GARDEN	BIRDHOU DECORAT MA GARD
23459	VICTORIAN GLASS HANGING T-LIGHT	T-LIGHT GLASS FLUTED ANTIQUE	FILIGREE HEART BUTTERFLY WHITE	METAL 4 HOOK HANGER FRENCH CHATEAU	NaN	NaN	NaN	NaN	NaN	N

5 rows × 41 columns

In [17]:
```
baskets = items_df.T.apply(lambda x: x.dropna().tolist()).tolist()
```

```
In [18]: for i in baskets[:5]:
             print(i)
```

```
['ROUND CAKE TIN VINTAGE RED', 'VINTAGE SNAP CARDS', 'NURSERY A,B,C PAINTED LETTERS', 'GOLD MUG BONE CHINA TREE OF LI
FE', '3 ROSE MORRIS BOXED CANDLES', '3 WHITE CHOC MORRIS BOXED CANDLES', 'FIVE HEART HANGING DECORATION', "GIANT 50'S
CHRISTMAS CRACKER", 'GUMBALL COAT RACK', 'FRENCH CARRIAGE LANTERN', 'THREE MINI HANGING FRAMES', 'BLUE HARMONICA IN B
OX', 'RABBIT NIGHT LIGHT', "POPPY'S PLAYHOUSE BEDROOM", "POPPY'S PLAYHOUSE KITCHEN", 'NOEL GARLAND PAINTED ZINC', 'SE
T OF 3 WOODEN HEART DECORATIONS', 'TRAVEL CARD WALLET RETROSPOT', 'FRENCH BLUE METAL DOOR SIGN 3', 'FRENCH BLUE METAL
DOOR SIGN 8', 'PANTRY MAGNETIC  SHOPPING LIST', 'DOORMAT FANCY FONT HOME SWEET HOME', 'CREAM CUPID HEARTS COAT HANGE
R', 'TOOL BOX SOFT TOY', "DOCTOR'S BAG SOFT TOY", 'TOMATO CHARLIE+LOLA COASTER SET', 'RETRO LONGBOARD IRONING BOARD C
OVER', 'JINGLE BELL HEART DECORATION', 'GARDENERS KNEELING PAD CUP OF TEA', 'GARDENERS KNEELING PAD KEEP CALM', 'RED
RETROSPOT CAKE STAND', "POPPY'S PLAYHOUSE KITCHEN", 'PINK OWL SOFT TOY', 'SPOTTY PINK DUCK DOORSTOP', 'WOODEN FRAME A
NTIQUE WHITE', 'FOUR HOOK  WHITE LOVEBIRDS', 'HANGING ENGRAVED METAL HEART', 'SET OF 6 SOLDIER SKITTLES', 'PINK BABY
BUNTING', "PAPER CHAIN KIT 50'S CHRISTMAS", 'NOEL WOODEN BLOCK LETTERS']
['Manual']
['ENGLISH ROSE NOTEBOOK A7 SIZE', 'HANGING WOOD AND FELT BUTTERFLY', 'PACK OF 12 VINTAGE DOILY TISSUES', 'RED RETROSP
OT CHARLOTTE BAG', 'CHARLOTTE BAG VINTAGE ALPHABET', 'ASS FLORAL PRINT MULTI SCREWDRIVER', 'PACK OF 12 RED APPLE TISS
UES', 'PACK OF 12 LONDON TISSUES', 'PACK OF 12 SKULL TISSUES', 'MAGNETS PACK OF 4 SWALLOWS', 'MAGNETS PACK OF 4 VINTA
GE COLLAGE', 'PARTY BUNTING', 'PACK OF 12 SPACEBOY TISSUES', 'COLOUR GLASS T-LIGHT HOLDER HANGING', 'ENGLISH ROSE GAR
DEN SECATEURS', 'ASS FLORAL PRINT MULTI SCREWDRIVER', 'REX CASH+CARRY JUMBO SHOPPER', 'PACK OF 12 VINTAGE LEAF TISSUE
S', 'PACK OF 12 PINK PAISLEY TISSUES', 'PACK OF 12 BLUE PAISLEY TISSUES', 'PACK OF 12 RED RETROSPOT TISSUES', 'DOORST
OP RETROSPOT HEART', 'SET OF 6 HALLOWEEN GHOST T-LIGHTS', 'CERAMIC HEART FAIRY CAKE MONEY BANK', 'CERAMIC CHERRY CAKE
MONEY BANK', 'WOODEN ADVENT CALENDAR CREAM', 'BLUE CHARLIE+LOLA PERSONAL DOORSIGN', '"CHARLIE+LOLA""EXTREMELY BUSY""
SIGN"', 'RED CHARLIE+LOLA PERSONAL DOORSIGN', 'CHARLIE+LOLA MY ROOM DOOR SIGN', 'SET OF 4 KNICK KNACK TINS LEAF', 'SE
T OF 4 KNICK KNACK TINS DOILY', 'GLASS JAR DAISY FRESH COTTON WOOL', 'STRAWBERRY CHARLOTTE BAG', 'SMALL HANGING IVOR
Y/RED WOOD BIRD', 'HANGING WOOD AND FELT HEART', 'FAIRY CAKES NOTEBOOK A7 SIZE']
['CAKE PLATE LOVEBIRD WHITE', 'HANGING HEART MIRROR DECORATION', 'BUNNY WOODEN PAINTED WITH BIRD', 'JIGSAW TREE WITH
WATERING CAN', 'JIGSAW TREE WITH BIRDHOUSE', 'JIGSAW RABBIT AND BIRDHOUSE', '3 HOOK HANGER MAGIC GARDEN', 'BUNNY DECO
RATION MAGIC GARDEN', 'DECORATION  BUTTERFLY  MAGIC GARDEN', 'BIRDHOUSE DECORATION MAGIC GARDEN', 'DECORATION SITTING
BUNNY', 'OVERNIGHT BAG VINTAGE ROSE PAISLEY', 'POCKET BAG BLUE PAISLEY RED SPOT', 'POCKET BAG PINK PAISELY BROWN SPO
T', 'HANGING METAL CHICKEN DECORATION', '200 RED + WHITE BENDY STRAWS', 'FIVE CATS HANGING DECORATION', 'FIVE HEART H
ANGING DECORATION', 'BIRD DECORATION GREEN POLKADOT', 'ROUND SNACK BOXES SET OF4 WOODLAND', 'ROUND SNACK BOXES SET OF
4 SKULLS', 'ROUND CONTAINER SET OF 5 RETROSPOT', 'WOODLAND PARTY BAG + STICKER SET', 'HOME GARLAND PAINTED ZINC', 'LU
NCH BOX WITH CUTLERY RETROSPOT', 'CHARLOTTE BAG SUKI DESIGN', 'KINGS CHOICE TEA CADDY', 'DOORMAT AIRMAIL', 'AIRLINE B
AG VINTAGE TOKYO 78', 'BOTTLE BAG RETROSPOT', 'WALL TIDY RETROSPOT', 'TOY TIDY SPACEBOY', 'TOY TIDY PINK POLKADOT',
'JUMBO BAG SPACEBOY DESIGN', 'WOODEN BOX OF DOMINOES', 'KEY FOB , SHED', 'JUMBO SHOPPER VINTAGE RED PAISLEY', 'LIPSTI
CK PEN RED', 'REGENCY CAKESTAND 3 TIER', 'WOODEN REGATTA BUNTING', 'WOODEN UNION JACK BUNTING']
['VICTORIAN GLASS HANGING T-LIGHT', 'T-LIGHT GLASS FLUTED ANTIQUE', 'FILIGREE HEART BUTTERFLY WHITE', 'METAL 4 HOOK H
ANGER FRENCH CHATEAU']
```

In [19]:
```
association_rules = apriori(baskets, min_support=0.01, min_confidence=0.2,
                            min_lift=3, min_length=2)
association_results = list(association_rules)
```

In [20]:
```
print('Rules generated: ', len(association_results))
```

```
Rules generated:  85
```

In [21]:
```
print(association_results[0])
```

```
RelationRecord(items=frozenset({'60 TEATIME FAIRY CAKE CASES', 'PACK OF 72 RETROSPOT CAKE CASES'}), support=0.0109023
81751090238, ordered_statistics=[OrderedStatistic(items_base=frozenset({'60 TEATIME FAIRY CAKE CASES'}), items_add=fr
ozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), confidence=0.4209844559585492, lift=9.999638750696695), OrderedStatisti
c(items_base=frozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), items_add=frozenset({'60 TEATIME FAIRY CAKE CASES'}), co
nfidence=0.2589641434262948, lift=9.999638750696695)])
```

In [22]:
```
print(association_results[8])
```

```
RelationRecord(items=frozenset({'RED RETROSPOT CHARLOTTE BAG', 'CHARLOTTE BAG PINK POLKADOT'}), support=0.01103656491
1103657, ordered_statistics=[OrderedStatistic(items_base=frozenset({'CHARLOTTE BAG PINK POLKADOT'}), items_add=frozen
set({'RED RETROSPOT CHARLOTTE BAG'}), confidence=0.46207865168539325, lift=14.012781899024997), OrderedStatistic(item
s_base=frozenset({'RED RETROSPOT CHARLOTTE BAG'}), items_add=frozenset({'CHARLOTTE BAG PINK POLKADOT'}), confidence=
0.3346897253306205, lift=14.012781899024997)])
```

this command shows us the 8th association rule.

In [23]:
```
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("======================================")
```

In [24]:
```
display_rules(association_results[:5])
```

```
Rule: 60 TEATIME FAIRY CAKE CASES -> PACK OF 72 RETROSPOT CAKE CASES
Support: 0.010902381751090238
Confidence: 0.4209844559585492
Lift: 9.999638750696695
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE GREEN
Support: 0.010097282791009729
Confidence: 0.3195329087048832
Lift: 16.948889694826633
=====================================
Rule: ALARM CLOCK BAKELIKE PINK -> ALARM CLOCK BAKELIKE GREEN
Support: 0.013485407581348854
Confidence: 0.4267515923566878
Lift: 16.5213830755232
=====================================
Rule: ALARM CLOCK BAKELIKE RED -> ALARM CLOCK BAKELIKE GREEN
Support: 0.020496477692049648
Confidence: 0.648619957537155
Lift: 18.64547823932747
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE RED
Support: 0.011774572291177458
Confidence: 0.6245551601423487
Lift: 17.953702337361054
=====================================
```

In [25]:

```
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[25]:

```
[('WHITE HANGING HEART T-LIGHT HOLDER', 2220),
 ('REGENCY CAKESTAND 3 TIER', 2032),
 ('JUMBO BAG RED RETROSPOT', 1955),
 ('PARTY BUNTING', 1668),
 ('LUNCH BAG RED RETROSPOT', 1536),
 ('ASSORTED COLOUR BIRD ORNAMENT', 1430),
 ('SET OF 3 CAKE TINS PANTRY DESIGN', 1357),
 ('PACK OF 72 RETROSPOT CAKE CASES', 1280),
 ('LUNCH BAG  BLACK SKULL', 1257),
 ('NATURAL SLATE HEART CHALKBOARD', 1197)]
```

Q4 (a) How many of these items can you find in the rules you have just displayed? From the 5 association rules dsplayed, just 1 item from the frequent list is part of it - ('PACK OF 72 RETROSPOT CAKE CASES', 1280)

Q4 (b) Are all the top 10 items included? Provide an explanation as to why these items may be missing/present in the rules. only 1 of the 10 top items is included in the list. There are several factors that may be responsible for this. One reason could be that we have generated 85 association rules, but we have only displayed 5 of the association rules, the frequent items displayed are 10 items. If we display the 85 association rules, we would most likely see that the items listed as being frequent occur in the association rules. It could also be that the minimum support level the minimum confidence level for the association rule generated is too high and strong.

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

In [26]:

```
association_rules = apriori(baskets, min_support=0.015, min_confidence=0.7,
                            min_lift=3, min_length=2)
association_results = list(association_rules)
```

In [27]:

```
# no of association rules generated using Min Support = 0.015, Min Confidence = 0.7, Min Lift = 3
print('Rules generated: ', len(association_results))
```

```
Rules generated:  4
```

In [28]:

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

In [29]:

```
# no of association rules generated using Min Support = 0.009, Min Confidence = 0.5, Min Lift = 3
print('Rules generated: ', len(association_results))
```

```
Rules generated:  47
```

In [30]:

```
association_rules = apriori(baskets, min_support=0.015, min_confidence=0.5,
                            min_lift=9, min_length=2)
association_results = list(association_rules)
```

In [31]:

```
# no of association rules generated using Min Support = 0.015, Min Confidence = 0.5, Min Lift = 9
print('Rules generated: ', len(association_results))
```

```
Rules generated:  12
```

Setting 1: Rules generated = 4 Setting 2: Rules generated = 47 Setting 3: Rules generated = 12 The value of the minimum confidence and support affects the number of rules generated. A high confidence and support level will reduce the number of rules generated, even though the rules would be more reliable and stronger, while a low confidence and support level will increase the number of rules generated. In Setting 1, a minimum support of 0.015 and minimum confidence f 0.7 generated 4 association rules, while in setting 3, the same minimum support, but a lesser confidence level and greater minimum lift generated more association rules.

```
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.
To do this, we need to run the codes in line 31-37 to get the 'Date' in dd-mm-yyyy format converted to days of the
week. An additional column is added.
```

In [32]:

```
import datetime
```

```
In [33]: from dateutil import parser
```

```
In [34]: data = pd.read_csv('MoradekeAdeleye_9810.csv', parse_dates=['Date'])
```

```
In [35]: data['day_of_the_week'] = data['Date'].dt.day_name()
```

```
In [36]: data.head()
```

Out[36]:

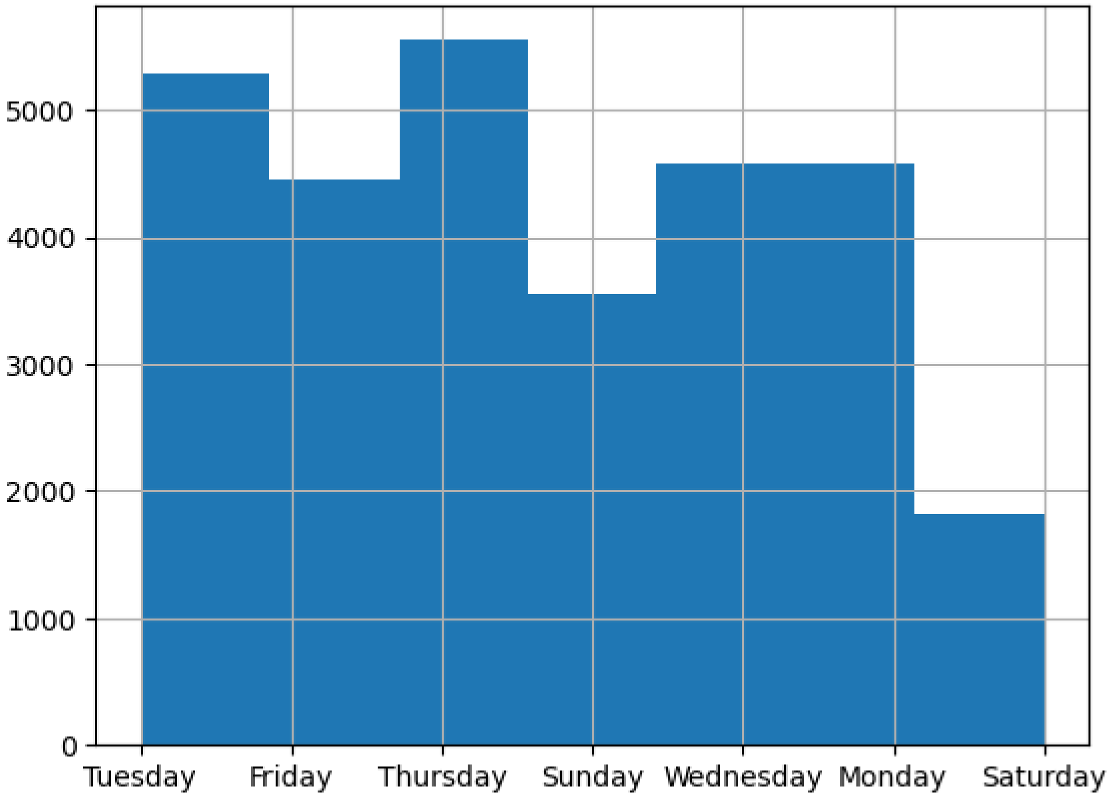
item_4	item_5	...	item_32	item_33	item_34	item_35	item_36	item_37	item_38	item_39	item_40	day_
ROSE ORRIS 3OXED NDLES	3 WHITE CHOC MORRIS BOXED CANDLES	...	PINK OWL SOFT TOY	SPOTTY PINK DUCK DOORSTOP	WOODEN FRAME ANTIQUE WHITE	FOUR HOOK WHITE LOVEBIRDS	HANGING ENGRAVED METAL HEART	SET OF 6 SOLDIER SKITTLES	PINK BABY BUNTING	PAPER CHAIN KIT 50'S CHRISTMAS	NOEL WOODEN BLOCK LETTERS	
NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
LOTTE BAG NTAGE HABET	ASS FLORAL PRINT MULTI SCREWDRIVER	...	GLASS JAR DAISY FRESH COTTON WOOL	STRAWBERRY CHARLOTTE BAG	SMALL HANGING IVORY/RED WOOD BIRD	HANGING WOOD AND FELT HEART	FAIRY CAKES NOTEBOOK A7 SIZE	NaN	NaN	NaN	NaN	
IGSAW E WITH HOUSE	JIGSAW RABBIT AND BIRDHOUSE	...	TOY TIDY PINK POLKADOT	JUMBO BAG SPACEBOY DESIGN	WOODEN BOX OF DOMINOES	KEY FOB , SHED	JUMBO SHOPPER VINTAGE RED PAISLEY	LIPSTICK PEN RED	REGENCY CAKESTAND 3 TIER	WOODEN REGATTA BUNTING	WOODEN UNION JACK BUNTING	
NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

so now, we have an additional column added to become 46 columns as against the 45 columns we had initially. We need to draw an histogram to check the frequency of the days of the week. But first of all, we need to group the data by days of the week.

In [37]:
```
day_counts = data['day_of_the_week'].value_counts()
```

In [38]:
```
data['day_of_the_week'].hist(bins=7)
ax.set_xlabel("day_of_the_week")
ax.set_ylabel("Frequency")
```

Out[38]:
```
Text(24.0, 0.5, 'Frequency')
```



From this graph, we notice that more people shop on Tuesdays and Thursdays. The association rules will be checked for Wednesday(a weekday) and Sunday(a weekend) to identify the buying patterns and see if there is any difference between items bought together during the weekend and weekday.

In [39]:

```
#we need to extract the transactions done on Wednesday
day_to_extract = 'Wednesday'
new_data = data[data['day_of_the_week'] == day_to_extract]
```

In [40]:

```
#we use this to check the first five rows of the dataset
new_data.head()
```

Out[40]:

	Unnamed: 0	transaction_ID	Date	Time	item_0	item_1	item_2	item_3	item_4	item_5	...	item_32
8	8288	548289	2011-03-30	11:48	COFFEE MUG PEARS DESIGN	COFFEE MUG APPLES DESIGN	DOG BOWL CHASING BALL DESIGN	JUMBO SHOPPER VINTAGE RED PAISLEY	HOOK, 1 HANGER ,MAGIC GARDEN	ENGLISH ROSE NOTEBOOK A7 SIZE	...	WRAP ENGLISH ROSE
19	21002	566726	2011-09-14	13:12	HOME SWEET HOME BLACKBOARD	LOVE HEART POCKET WARMER	HOME BUILDING BLOCK WORD	DOG BOWL CHASING BALL DESIGN	ILLUSTRATED CAT BOWL	JUMBO BAG ALPHABET	...	EDWARDIAN PARASOL RED
20	8808	548997	2011-05-04	14:33	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
26	20702	566290	2011-11-09	15:24	TRIPLE HOOK ANTIQUE IVORY ROSE	HEART IVORY TRELLIS LARGE	HEART IVORY TRELLIS SMALL	MULTICOLOUR SPRING FLOWER MUG	ENGLISH ROSE HOT WATER BOTTLE	PINK PAISLEY SQUARE TISSUE BOX	...	12 RED ROSE PEG PLACE SETTINGS
27	19318	564248	2011-08-24	10:37	BATHROOM METAL SIGN	RED HANGING HEART T-LIGHT HOLDER	RED RETROSPOT BOWL	DOORMAT RED RETROSPOT	WHITE HANGING HEART T-LIGHT HOLDER	CERAMIC CAKE BOWL + HANGING CAKES	...	NaN

5 rows × 46 columns

In [41]:
```
items_df=new_data.drop(["Unnamed: 0","Date", "Time", "day_of_the_week"], axis=1)
```

#We need to drop unnecessary columns that are not useful to the transaction. We are dropping "Unamed = 0", because we don't know what it is, "Date", "Time" and "day_of_the_week", because they are not necessary. Day_of_the_week is not necessary since we already know that we are analysing for Wednesday. So we need only the transacton id and the items bought.

In [42]:
```
items_df.head()
```

Out[42]:

	transaction_ID	item_0	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	...	it
8	548289	COFFEE MUG PEARS DESIGN	COFFEE MUG APPLES DESIGN	DOG BOWL CHASING BALL DESIGN	JUMBO SHOPPER VINTAGE RED PAISLEY	HOOK, 1 HANGER ,MAGIC GARDEN	ENGLISH ROSE NOTEBOOK A7 SIZE	SET OF 2 TEA TOWELS APPLE AND PEARS	POTTING SHED TWINE	WHITE WOOD GARDEN PLANT LADDER	...	SE EN PLACE
19	566726	HOME SWEET HOME BLACKBOARD	LOVE HEART POCKET WARMER	HOME BUILDING BLOCK WORD	DOG BOWL CHASING BALL DESIGN	ILLUSTRATED CAT BOWL	JUMBO BAG ALPHABET	JUMBO STORAGE BAG SUKI	JUMBO STORAGE BAG SKULLS	JUMBO BAG OWLS	...	\ E BOAR \ F
20	548997	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
26	566290	TRIPLE HOOK ANTIQUE IVORY ROSE	HEART IVORY TRELLIS LARGE	HEART IVORY TRELLIS SMALL	MULTICOLOUR SPRING FLOWER MUG	ENGLISH ROSE HOT WATER BOTTLE	PINK PAISLEY SQUARE TISSUE BOX	POCKET BAG BLUE PAISLEY RED SPOT	POCKET BAG PINK PAISELY BROWN SPOT	WASH BAG VINTAGE ROSE PAISLEY	...	BE CR H L
27	564248	BATHROOM METAL SIGN	RED HANGING HEART T-LIGHT HOLDER	RED RETROSPOT BOWL	DOORMAT RED RETROSPOT	WHITE HANGING HEART T-LIGHT HOLDER	CERAMIC CAKE BOWL + HANGING CAKES	NaN	NaN	NaN	...	

5 rows × 42 columns

```
In [43]: baskets = items_df.T.apply(lambda x: x.dropna().tolist()).tolist()
```

```
In [44]: for i in baskets[:5]:
             print(i)
```

```
['548289', 'COFFEE MUG PEARS  DESIGN', 'COFFEE MUG APPLES DESIGN', 'DOG BOWL CHASING BALL DESIGN', 'JUMBO SHOPPER VIN
TAGE RED PAISLEY', 'HOOK, 1 HANGER ,MAGIC GARDEN', 'ENGLISH ROSE NOTEBOOK A7 SIZE', 'SET OF 2 TEA TOWELS APPLE AND PE
ARS', 'POTTING SHED TWINE', 'WHITE WOOD GARDEN PLANT LADDER', 'HERB MARKER THYME', 'HERB MARKER ROSEMARY', 'HERB MARK
ER BASIL', 'HERB MARKER PARSLEY', 'HERB MARKER CHIVES', 'HOMEMADE JAM SCENTED CANDLES', 'HYACINTH BULB T-LIGHT CANDLE
S', 'PAINTED METAL PEARS ASSORTED', 'ASSORTED COLOUR BIRD ORNAMENT', 'ENAMEL FLOWER JUG CREAM', 'PEG BAG APPLES DESIG
N', 'SET/20 STRAWBERRY PAPER NAPKINS', 'STRAWBERRY CERAMIC TRINKET BOX', 'ROUND SNACK BOXES SET OF 4 FRUITS', 'ASSORT
ED COLOUR MINI CASES', 'SET OF 2 TINS JARDIN DE PROVENCE', 'SET OF 3 CAKE TINS SKETCHBOOK', 'RECIPE BOX BLUE SKETCHBO
OK DESIGN', 'SET OF 6 SPICE TINS PANTRY DESIGN', 'SET 2 PANTRY DESIGN TEA TOWELS', 'CUBIC MUG PINK POLKADOT', 'SET OF
4 ENGLISH ROSE COASTERS', 'SET OF 4 ENGLISH ROSE PLACEMATS', 'WRAP ENGLISH ROSE', 'IVORY HANGING DECORATION  BIRD',
'IVORY HANGING DECORATION  HEART', 'WRAP DAISY CARPET', 'PACK OF 20 NAPKINS RED APPLES', 'VINTAGE CREAM 3 BASKET CAKE
STAND', 'RED TEA TOWEL CLASSIC DESIGN', 'POTTING SHED TEA MUG', 'SET OF 2 WOODEN MARKET CRATES']
['566726', 'HOME SWEET HOME BLACKBOARD', 'LOVE HEART POCKET WARMER', 'HOME BUILDING BLOCK WORD', 'DOG BOWL CHASING BA
LL DESIGN', 'ILLUSTRATED CAT BOWL', 'JUMBO BAG ALPHABET', 'JUMBO STORAGE BAG SUKI', 'JUMBO STORAGE BAG SKULLS', 'JUMB
O BAG OWLS', 'JUMBO  BAG BAROQUE BLACK WHITE', 'SMALL ZINC HEART WALL ORGANISER', 'SMALL IVORY HEART WALL ORGANISER',
'LARGE IVORY HEART WALL ORGANISER', 'SCOTTIE DOG HOT WATER BOTTLE', 'KNITTED UNION FLAG HOT WATER BOTTLE', 'WHITE SKU
LL HOT WATER BOTTLE', 'LOVE HOT WATER BOTTLE', 'VINTAGE UNION JACK DOORSTOP', 'HAND WARMER RED RETROSPOT', 'HAND WARM
ER OWL DESIGN', 'HAND WARMER RED LOVE HEART', 'HAND WARMER SCOTTY DOG DESIGN', 'HAND WARMER RED RETROSPOT', 'HAND WAR
MER RED RETROSPOT', 'HAND WARMER BIRD DESIGN', 'HAND WARMER BABUSHKA DESIGN', 'HAND WARMER OWL DESIGN', 'WOODEN FRAME
ANTIQUE WHITE', 'WOODEN PICTURE FRAME WHITE FINISH', 'RED FLORAL FELTCRAFT SHOULDER BAG', 'FELTCRAFT CUSHION BUTTERFL
Y', 'WOOD BLACK BOARD ANT WHITE FINISH', 'EDWARDIAN PARASOL RED', 'JUMBO BAG VINTAGE CHRISTMAS', 'IVORY STRING CURTAI
N WITH POLE', 'MAGIC TREE -PAPER FLOWERS', 'WOOD 2 DRAWER CABINET WHITE FINISH']
['548997']
['566290', 'TRIPLE HOOK ANTIQUE IVORY ROSE', 'HEART IVORY TRELLIS LARGE', 'HEART IVORY TRELLIS SMALL', 'MULTICOLOUR S
PRING FLOWER MUG', 'ENGLISH ROSE HOT WATER BOTTLE', 'PINK PAISLEY SQUARE TISSUE BOX', 'POCKET BAG BLUE PAISLEY RED SP
OT', 'POCKET BAG PINK PAISELY BROWN SPOT', 'WASH BAG VINTAGE ROSE PAISLEY', 'PLASTERS IN TIN VINTAGE PAISLEY', 'PACK
OF 12 PINK PAISLEY TISSUES', 'LAVENDER SCENTED FABRIC HEART', '3 DRAWER ANTIQUE WHITE WOOD CABINET', 'TOY TIDY PINK P
OLKADOT', 'PINK VINTAGE PAISLEY PICNIC BAG', 'JUMBO BAG SCANDINAVIAN BLUE PAISLEY', 'SCANDINAVIAN PAISLEY PICNIC BA
G', 'JUMBO BAG PINK POLKADOT', 'JUMBO BAG PINK VINTAGE PAISLEY', 'EGG FRYING PAN IVORY', 'LIPSTICK PEN BABY PINK', 'F
EATHER PEN,LIGHT PINK', 'ENGLISH ROSE NOTEBOOK A7 SIZE', 'HOME SWEET HOME HOOK', 'BATHROOM HOOK', 'BLUE HARMONICA IN
BOX', 'RED  HARMONICA IN BOX', 'HEART SHAPED HOLLY WREATH', 'ASS FLORAL PRINT MULTI SCREWDRIVER', 'BEADED CRYSTAL HEA
RT GREEN LARGE', 'BEADED CRYSTAL HEART PINK SMALL', 'BEADED CRYSTAL HEART BLUE  LARGE', '12 RED ROSE PEG PLACE SETTIN
GS', 'POPCORN HOLDER', 'MODERN FLORAL STATIONERY SET', 'FEATHER PEN,LIGHT PINK', 'BALLOON PUMP WITH 10 BALLOONS', 'SE
T OF 3 NOTEBOOKS IN PARCEL', 'DOORMAT UNION FLAG', 'DOORMAT ENGLISH ROSE', 'BIRTHDAY CARD, RETRO SPOT']
['564248', 'BATHROOM METAL SIGN', 'RED HANGING HEART T-LIGHT HOLDER', 'RED RETROSPOT BOWL', 'DOORMAT RED RETROSPOT',
'WHITE HANGING HEART T-LIGHT HOLDER', 'CERAMIC CAKE BOWL + HANGING CAKES']
```

In [45]:
```
association_rules = apriori(baskets, min_support=0.01, min_confidence=0.2,
                            min_lift=3, min_length=2)
association_results = list(association_rules)
```

In [46]:
```
print('Rules generated: ', len(association_results))
```

```
Rules generated:  112
```

In [47]:
```
print(association_results[0])
```

```
RelationRecord(items=frozenset({'60 TEATIME FAIRY CAKE CASES', 'PACK OF 72 RETROSPOT CAKE CASES'}), support=0.0113537
11790393014, ordered_statistics=[OrderedStatistic(items_base=frozenset({'60 TEATIME FAIRY CAKE CASES'}), items_add=fr
ozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), confidence=0.4262295081967214, lift=10.495328750220343), OrderedStatist
ic(items_base=frozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), items_add=frozenset({'60 TEATIME FAIRY CAKE CASES'}), c
onfidence=0.2795698924731183, lift=10.495328750220343)])
```

In [48]:
```
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("=====================================")
```

In [49]:

```
display_rules(association_results[:5])
```

```
Rule: 60 TEATIME FAIRY CAKE CASES -> PACK OF 72 RETROSPOT CAKE CASES
Support: 0.011353711790393014
Confidence: 0.4262295081967214
Lift: 10.495328750220343
=====================================
Rule: ALARM CLOCK BAKELIKE CHOCOLATE -> ALARM CLOCK BAKELIKE RED
Support: 0.010480349344978166
Confidence: 0.6666666666666666
Lift: 17.649325626204238
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE GREEN
Support: 0.010043668122270743
Confidence: 0.2948717948717949
Lift: 15.346736596736596
=====================================
Rule: ALARM CLOCK BAKELIKE PINK -> ALARM CLOCK BAKELIKE GREEN
Support: 0.017685589519650654
Confidence: 0.5192307692307692
Lift: 17.746842709529275
=====================================
Rule: ALARM CLOCK BAKELIKE RED -> ALARM CLOCK BAKELIKE GREEN
Support: 0.0222707423580786
Confidence: 0.6538461538461539
Lift: 17.309915518008005
=====================================
```

In [50]:

```
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[50]:

```
[('WHITE HANGING HEART T-LIGHT HOLDER', 362),
 ('JUMBO BAG RED RETROSPOT', 323),
 ('REGENCY CAKESTAND 3 TIER', 298),
 ('PARTY BUNTING', 256),
 ('SET OF 3 CAKE TINS PANTRY DESIGN', 221),
 ('NATURAL SLATE HEART CHALKBOARD', 217),
 ('LUNCH BAG RED RETROSPOT', 217),
 ('ASSORTED COLOUR BIRD ORNAMENT', 208),
 ('JUMBO BAG PINK POLKADOT', 206),
 ('RECIPE BOX PANTRY YELLOW DESIGN', 203)]
```

In [51]:

```
#we now need to extract the transactions done on Sundays
day_to_extract2 = 'Sunday'
new_data2 = data[data['day_of_the_week'] == day_to_extract2]
```

In [52]:
```python
#we use this to check the first five rows of the dataset
new_data2.head()
```

Out[52]:

_1	item_2	item_3	item_4	item_5	...	item_32	item_33	item_34	item_35	item_36	item_37	item_38	item_39	item_40	day_
HT SS ED JE	FILIGREE HEART BUTTERFLY WHITE	METAL 4 HOOK HANGER FRENCH CHATEAU	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
ED IG T- HT ER	60 TEATIME FAIRY CAKE CASES	ASSORTED COLOUR BIRD ORNAMENT	BULL DOG BOTTLE OPENER	PARISIENNE JEWELLERY DRAWER	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EL RD T I VE ON	ABSTRACT CIRCLE JOURNAL	WHITE STITCHED WALL CLOCK	PACK OF SIX LED TEA LIGHTS	WHITE HANGING HEART T-LIGHT HOLDER	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
IG HE ED AL GN	FRENCH GARDEN SIGN BLUE METAL	TOILET SIGN OCCUPIED OR VACANT	HAND OVER THE CHOCOLATE SIGN	MORE BUTTER METAL SIGN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
RT LE TY	WALL ART DOG LICENCE	WRAP 50'S CHRISTMAS	ROLL WRAP VINTAGE CHRISTMAS	ROLL WRAP 50'S CHRISTMAS	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

```
#We need to drop unnecessary columns that are not useful to the transaction. We are dropping "Unamed = 0", because we
don't know what it is, "Date", "Time" and "day_of_the_week", because they are not necessary.  Day_of_the_week is not
necessary since we already know that we are analysing for Sunday
```

In [53]:
```python
items_df2=new_data2.drop(["Unnamed: 0","Date", "Time", "day_of_the_week"], axis=1)
```

In [54]:
```
items_df2.head()
```

Out[54]:

	transaction_ID	item_0	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	...	it
4	570089	VICTORIAN GLASS HANGING T-LIGHT	T-LIGHT GLASS FLUTED ANTIQUE	FILIGREE HEART BUTTERFLY WHITE	METAL 4 HOOK HANGER FRENCH CHATEAU	NaN	NaN	NaN	NaN	NaN	...	
14	562455	PLACE SETTING WHITE HEART	RED HANGING HEART T-LIGHT HOLDER	60 TEATIME FAIRY CAKE CASES	ASSORTED COLOUR BIRD ORNAMENT	BULL DOG BOTTLE OPENER	PARISIENNE JEWELLERY DRAWER	JAM MAKING SET WITH JARS	GROW A FLYTRAP OR SUNFLOWER IN TIN	NATURAL SLATE HEART CHALKBOARD	...	
23	563930	TRAVEL CARD WALLET SKULLS	TRAVEL CARD WALLET I LOVE LONDON	ABSTRACT CIRCLE JOURNAL	WHITE STITCHED WALL CLOCK	PACK OF SIX LED TEA LIGHTS	WHITE HANGING HEART T-LIGHT HOLDER	NaN	NaN	NaN	...	
40	560256	LARGE CHINESE STYLE SCISSOR	POTTERING IN THE SHED METAL SIGN	FRENCH GARDEN SIGN BLUE METAL	TOILET SIGN OCCUPIED OR VACANT	HAND OVER THE CHOCOLATE SIGN	MORE BUTTER METAL SIGN	PAINTED METAL PEARS ASSORTED	ENGLISH ROSE GARDEN SECATEURS	VICTORIAN METAL POSTCARD SPRING	...	
46	570014	WALL ART CLASSIC PUDDINGS	WALL ART BICYCLE SAFETY	WALL ART DOG LICENCE	WRAP 50'S CHRISTMAS	ROLL WRAP VINTAGE CHRISTMAS	ROLL WRAP 50'S CHRISTMAS	NaN	NaN	NaN	...	

5 rows × 42 columns

In [55]:
```
baskets2 = items_df2.T.apply(lambda x: x.dropna().tolist()).tolist()
```

In [56]:

```
for i in baskets2[:5]:
    print(i)
```

```
['570089', 'VICTORIAN GLASS HANGING T-LIGHT', 'T-LIGHT GLASS FLUTED ANTIQUE', 'FILIGREE HEART BUTTERFLY WHITE', 'META
L 4 HOOK HANGER FRENCH CHATEAU']
['562455', 'PLACE SETTING WHITE HEART', 'RED HANGING HEART T-LIGHT HOLDER', '60 TEATIME FAIRY CAKE CASES', 'ASSORTED
COLOUR BIRD ORNAMENT', 'BULL DOG BOTTLE OPENER', 'PARISIENNE JEWELLERY DRAWER', 'JAM MAKING SET WITH JARS', 'GROW A F
LYTRAP OR SUNFLOWER IN TIN', 'NATURAL SLATE HEART CHALKBOARD', 'WRAP ENGLISH ROSE', 'WRAP PINK FAIRY CAKES', 'DOORSTO
P RETROSPOT HEART', 'SWEETHEART CERAMIC TRINKET BOX', 'GLITTER HANGING BUTTERFLY STRING', 'STRAWBERRY FAIRY CAKE TEAP
OT', 'SET/6 PINK  BUTTERFLY T-LIGHTS', 'SET/6 TURQUOISE BUTTERFLY T-LIGHTS']
['563930', 'TRAVEL CARD WALLET SKULLS', 'TRAVEL CARD WALLET I LOVE LONDON', 'ABSTRACT CIRCLE JOURNAL', 'WHITE STITCHE
D WALL CLOCK', 'PACK OF SIX LED TEA LIGHTS', 'WHITE HANGING HEART T-LIGHT HOLDER']
['560256', 'LARGE CHINESE STYLE SCISSOR', 'POTTERING IN THE SHED METAL SIGN', 'FRENCH GARDEN SIGN BLUE METAL', 'TOILE
T SIGN OCCUPIED OR VACANT', 'HAND OVER THE CHOCOLATE   SIGN', 'MORE BUTTER METAL SIGN', 'PAINTED METAL PEARS ASSORTE
D', 'ENGLISH ROSE GARDEN SECATEURS', 'VICTORIAN  METAL POSTCARD SPRING', 'PAINTED METAL PEARS ASSORTED', 'EASTER DECO
RATION HANGING BUNNY', 'ASSORTED COLOUR BIRD ORNAMENT', 'CHOCOLATE THIS WAY METAL SIGN', 'PETIT TRAY CHIC', 'TREASURE
TIN BUFFALO BILL', 'SET/6 COLLAGE PAPER PLATES', 'SMALL WHITE HEART OF WICKER', 'FINE WICKER HEART', 'SET OF TEA COFF
EE SUGAR TINS PANTRY', 'SET OF 3 REGENCY CAKE TINS', 'PICTURE FRAME WOOD TRIPLE PORTRAIT', 'CREAM HEART CARD HOLDER',
'SET OF 3 WOODEN HEART DECORATIONS', 'FAMILY ALBUM WHITE PICTURE FRAME', 'WOOD 2 DRAWER CABINET WHITE FINISH']
['570014', 'WALL ART CLASSIC PUDDINGS', 'WALL ART BICYCLE SAFETY', 'WALL ART DOG LICENCE', "WRAP 50'S  CHRISTMAS", 'R
OLL WRAP VINTAGE CHRISTMAS', "ROLL WRAP 50'S CHRISTMAS"]
```

In [57]:

```
association_rules2 = apriori(baskets2, min_support=0.01, min_confidence=0.2,
                            min_lift=3, min_length=2)
association_results2 = list(association_rules2)
```

In [58]:

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

```
Rules generated:  165
```

In [59]:

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

```
RelationRecord(items=frozenset({'60 TEATIME FAIRY CAKE CASES', 'PACK OF 72 RETROSPOT CAKE CASES'}), support=0.0149337
84164553396, ordered_statistics=[OrderedStatistic(items_base=frozenset({'60 TEATIME FAIRY CAKE CASES'}), items_add=fr
ozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), confidence=0.5196078431372549, lift=11.974598930481285), OrderedStatist
ic(items_base=frozenset({'PACK OF 72 RETROSPOT CAKE CASES'}), items_add=frozenset({'60 TEATIME FAIRY CAKE CASES'}), c
onfidence=0.34415584415584416, lift=11.974598930481283)])
```

In [60]:

```
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 [61]:

```
display_rules(association_results2[:5])
```

```
Rule: 60 TEATIME FAIRY CAKE CASES -> PACK OF 72 RETROSPOT CAKE CASES
Support: 0.014933784164553396
Confidence: 0.5196078431372549
Lift: 11.974598930481285
=====================================
Rule: ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE GREEN
Support: 0.014370245139475908
Confidence: 0.3805970149253731
Lift: 18.253227107704717
=====================================
Rule: ALARM CLOCK BAKELIKE ORANGE -> ALARM CLOCK BAKELIKE GREEN
Support: 0.010989010989010999
Confidence: 0.291044776119403
Lift: 19.489017178259644
=====================================
Rule: ALARM CLOCK BAKELIKE PINK -> ALARM CLOCK BAKELIKE GREEN
Support: 0.013524936601859678
Confidence: 0.3582089552238806
Lift: 16.727415553809895
=====================================
Rule: ALARM CLOCK BAKELIKE RED -> ALARM CLOCK BAKELIKE GREEN
Support: 0.027895181741335588
Confidence: 0.7388059701492538
Lift: 19.567331254176874
=====================================
```

In [62]:

```
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[62]:

```
[('WHITE HANGING HEART T-LIGHT HOLDER', 362),
 ('JUMBO BAG RED RETROSPOT', 323),
 ('REGENCY CAKESTAND 3 TIER', 298),
 ('PARTY BUNTING', 256),
 ('SET OF 3 CAKE TINS PANTRY DESIGN', 221),
 ('NATURAL SLATE HEART CHALKBOARD', 217),
 ('LUNCH BAG RED RETROSPOT', 217),
 ('ASSORTED COLOUR BIRD ORNAMENT', 208),
 ('JUMBO BAG PINK POLKADOT', 206),
 ('RECIPE BOX PANTRY YELLOW DESIGN', 203)]
```

Comparing the 2 days, Wednesday and Sundays, people buy some, but not all the same items together on the 2 days. For instance, while displaying the first 5 association rules for the 2 different days, it was discovered that for the 2 days, only one association was different. Out of 5 associations, the following items were bought together : PACK OF 72 RETROSPOT CAKE CASES -> 60 TEATIME FAIRY CAKE CASE ALARM CLOCK BAKELIKE IVORY -> ALARM CLOCK BAKELIKE GREEN ALARM CLOCK BAKELIKE GREEN -> ALARM CLOCK BAKELIKE PINK ALARM CLOCK BAKELIKE RED -> ALARM CLOCK BAKELIKE GREEN The other items like ALARM CLOCK BAKELIKE RED was paired with ALARM CLOCK BAKELIKE CHOCOLATE on Wednesday, while the ALARM CLOCK BAKELIKE ORANGE was bought together with the ALARM CLOCK BAKELIKE GREEN on Sunday. On Wednesday, 112 association rules were generated, while on Sunday, 165 rules were generated. Even though, from the graph, more transactions were made on Wednesday than on Sunday, the higher number of association rules on Sunday showed that customers tend to pair some particular items together on Sundays, more than they do during the week. Also from the results, it can be seen that people usually bought some particular items together irrespective of the day of the week. As can be seen in the examples of the association rules made on Wednesday and Sunday, 4 out of the 5 rules examined were similar for both days, pointing to the fact that people usually had particular items they bought together irrespecive of the day of the week. With this, we can make some sales decisions. These include; putting the items on opposite sides of the store, so that customers would have to walk all the way to get the other item, and in the process be able to view other items being sold in the store. Alternatively, the items can be put together so that customers don't have to go through the stress of looking for the 2 items in the shop. We can also do sales promotions on the items so that people would keep buying them. It would also help to ensure proper stocking. Since we know the items that are bought together, we can stock up the same number during stockng. Then on Sundays, we have to be more intentional about the arrangement of the items, because more association rules are generated over the weekend.

In [ ]:

In [ ]: