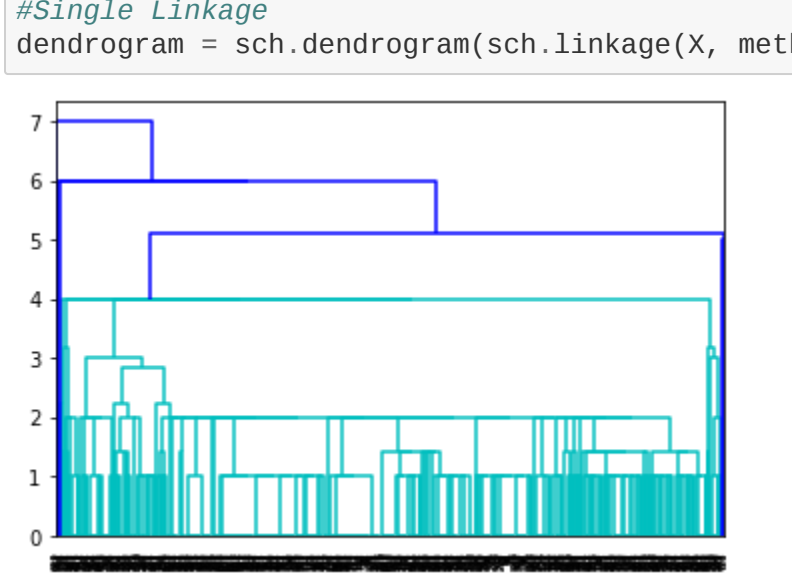
LAB 4

q1

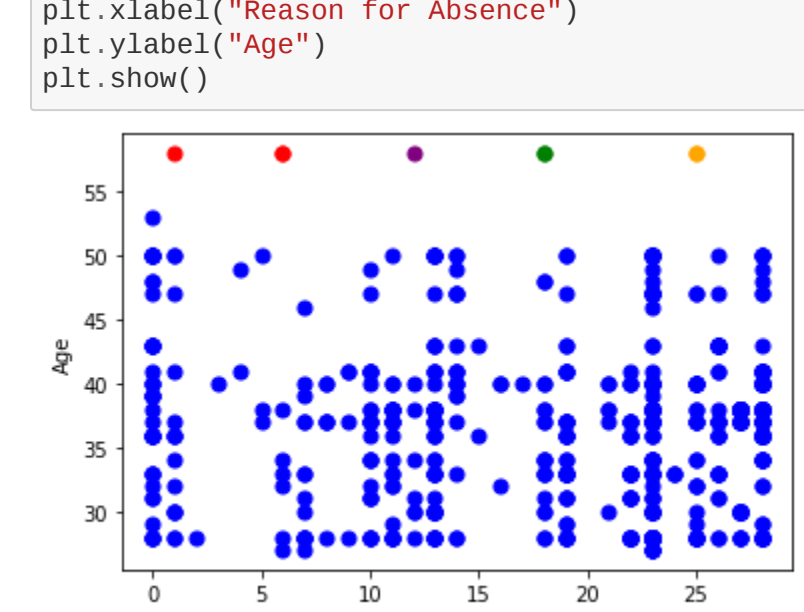
In [116]:
```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch

#considering only 2 coloumns (age and reason)
X = pd.read_excel('Absent.xls')
X = dataset.iloc[:,[1,8]].values
```

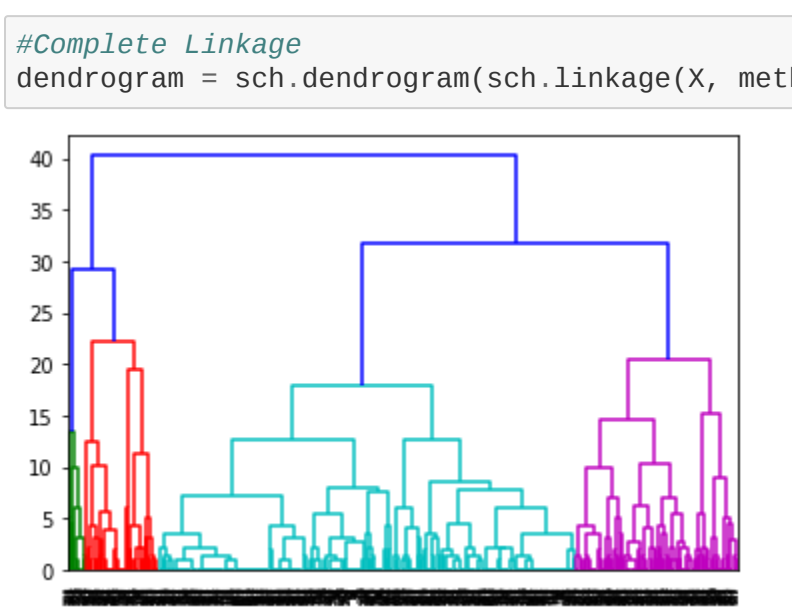
In [117]:
```python
#Single Linkage
dendrogram = sch.dendrogram(sch.linkage(X, method='single'))
```



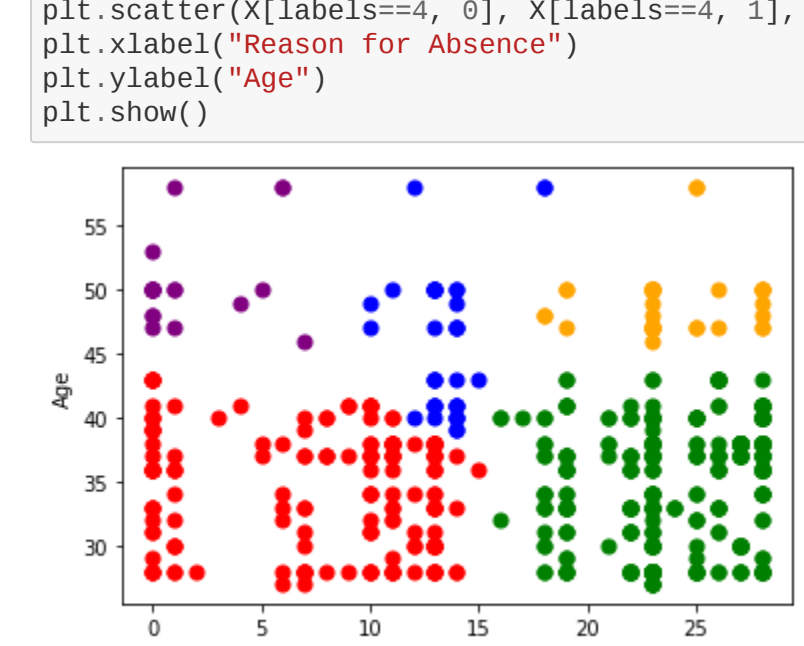
In [123]:
```python
model = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='single')
model.fit(X)
labels = model.labels_
plt.scatter(X[labels==0, 0], X[labels==0, 1], s=50, marker='o', color='red')
plt.scatter(X[labels==1, 0], X[labels==1, 1], s=50, marker='o', color='blue')
plt.scatter(X[labels==2, 0], X[labels==2, 1], s=50, marker='o', color='green')
plt.scatter(X[labels==3, 0], X[labels==3, 1], s=50, marker='o', color='purple')
plt.scatter(X[labels==4, 0], X[labels==4, 1], s=50, marker='o', color='orange')
plt.xlabel("Reason for Absence")
plt.ylabel("Age")
plt.show()
```



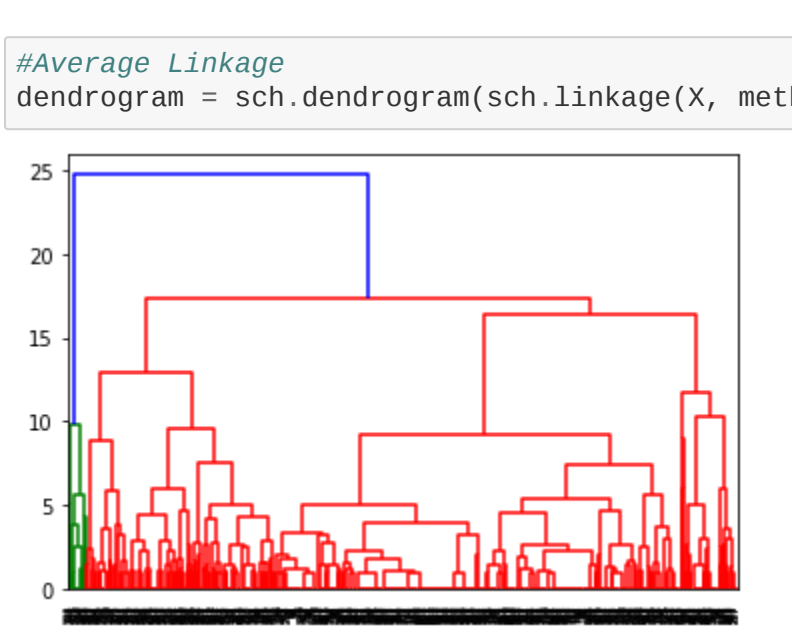
In [119]:
```python
#Complete Linkage
dendrogram = sch.dendrogram(sch.linkage(X, method='complete'))
```



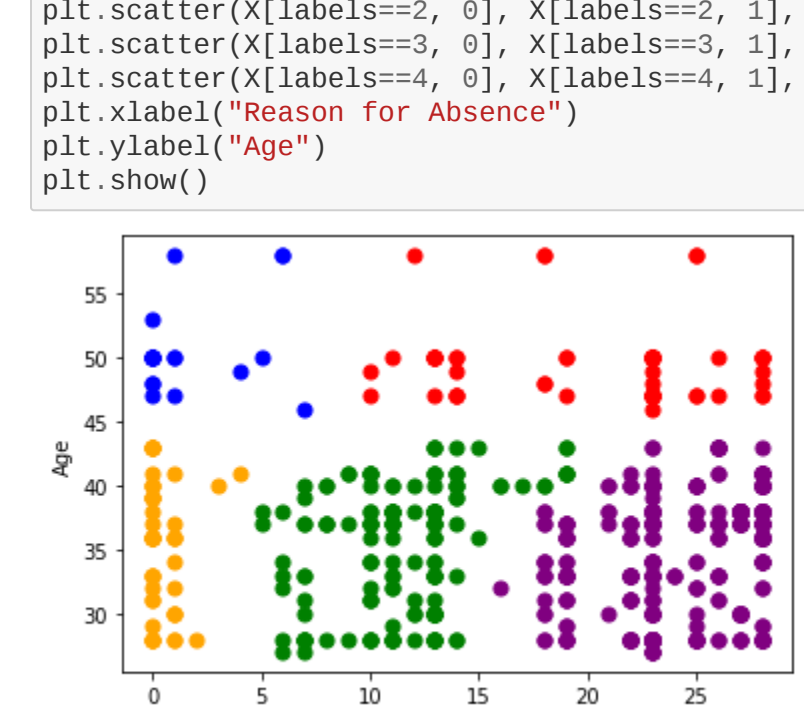
In [124]:
```python
model = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='complete')
model.fit(X)
labels = model.labels_
plt.scatter(X[labels==0, 0], X[labels==0, 1], s=50, marker='o', color='red')
plt.scatter(X[labels==1, 0], X[labels==1, 1], s=50, marker='o', color='blue')
plt.scatter(X[labels==2, 0], X[labels==2, 1], s=50, marker='o', color='green')
plt.scatter(X[labels==3, 0], X[labels==3, 1], s=50, marker='o', color='purple')
plt.scatter(X[labels==4, 0], X[labels==4, 1], s=50, marker='o', color='orange')
plt.xlabel("Reason for Absence")
plt.ylabel("Age")
plt.show()
```



In [125]:
```python
#Average Linkage
dendrogram = sch.dendrogram(sch.linkage(X, method='average'))
```



In [126]:
```python
model = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='average')
model.fit(X)
labels = model.labels_
plt.scatter(X[labels==0, 0], X[labels==0, 1], s=50, marker='o', color='red')
plt.scatter(X[labels==1, 0], X[labels==1, 1], s=50, marker='o', color='blue')
plt.scatter(X[labels==2, 0], X[labels==2, 1], s=50, marker='o', color='green')
plt.scatter(X[labels==3, 0], X[labels==3, 1], s=50, marker='o', color='purple')
plt.scatter(X[labels==4, 0], X[labels==4, 1], s=50, marker='o', color='orange')
plt.xlabel("Reason for Absence")
plt.ylabel("Age")
plt.show()
```



q2

In [11]:
```python
#import libraries and read data
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
df = pd.read_excel('Online Retail.xlsx')
df.head()

#cleaning data
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
df
```

Out[11]:

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

532621 rows × 8 columns

In [16]:
```python
#selecting one Australia to keep data set small
basket = (df[df['Country'] =="Australi"]
         .groupby(['InvoiceNo', 'Description'])['Quantity']
         .sum().unstack().reset_index().fillna(0)
         .set_index('InvoiceNo'))
basket
```

Out[16]:

Description	10 COLOUR SPACEBOY PEN	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	16 PIECE CUTLERY SET PANTRY DESIGN	20 DOLLY PEGS RETROSPOT	3 HOOK HANGER MAGIC GARDEN	3 STRIPEY MICE FELTCRAFT	3 TIER CAKE TIN GREEN AND CREAM
InvoiceNo									
536389	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
537676	0.0	0.0	0.0	0.0	0.0	24.0	0.0	0.0	0.0
539419	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
540267	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
540280	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
540557	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
540700	0.0	0.0	0.0	0.0	0.0	0.0	12.0	0.0	0.0
541149	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
541271	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
541520	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
541657	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
542542	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
543357	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
543372	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
543376	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
543989	0.0	384.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
545065	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
545475	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32.0
546135	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
547659	0.0	0.0	12.0	12.0	0.0	0.0	0.0	0.0	0.0
548661	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
549313	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
552956	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
553546	0.0	0.0	0.0	0.0	24.0	0.0	0.0	0.0	0.0
554037	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
554126	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
556917	48.0	0.0	240.0	0.0	0.0	0.0	0.0	120.0	0.0
556918	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
558536	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
558537	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
559919	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
559920	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
560033	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
560473	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
560491	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
561040	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
561228	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
563179	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
563614	0.0	0.0	0.0	0.0	0.0	0.0	0.0	80.0	0.0
565145	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
565146	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
565466	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
567085	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
568145	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
568687	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
568695	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
568708	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
569647	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
569650	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
569722	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
569723	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
574014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
574138	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
574469	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
576394	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
576586	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
578459	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

57 rows × 608 columns

In [18]:
```python
#encoding
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1
basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
basket_sets
```

Out[18]:

Description	10 COLOUR SPACEBOY PEN	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	16 PIECE CUTLERY SET PANTRY DESIGN	20 DOLLY PEGS RETROSPOT	3 HOOK HANGER MAGIC GARDEN	3 STRIPEY MICE FELTCRAFT	3 TIER CAKE TIN GREEN AND CREAM
InvoiceNo									
536389	0	0	0	0	0	0	0	0	0
537676	0	0	0	0	0	1	0	0	0
539419	0	0	0	0	0	0	0	0	0
540267	0	0	0	0	0	0	0	0	0
540280	0	0	0	0	0	0	0	0	0
540557	0	0	0	0	0	0	0	0	0
540700	0	0	0	0	0	0	1	0	0
541149	0	0	0	0	0	0	0	0	0
541271	0	0	0	0	0	0	0	0	0
541520	0	0	0	0	0	0	0	0	0
541657	0	0	0	0	0	0	0	0	0
542542	0	0	0	0	0	0	0	0	0
543357	0	0	0	0	0	0	0	0	0
543372	0	0	0	0	0	0	0	0	0
543376	0	0	0	0	0	0	0	0	0
543989	0	1	0	0	0	0	0	0	0
545065	0	0	0	0	0	0	0	0	0
545475	0	0	0	0	0	0	0	0	1
546135	0	0	0	0	0	0	0	0	0
547659	0	0	1	1	0	0	0	0	0
548661	0	0	0	0	0	0	0	0	0
549313	0	0	0	0	0	0	0	0	0
552956	0	0	0	0	0	0	0	0	0
553546	0	0	0	0	1	0	0	0	0
554037	0	0	0	0	0	0	0	0	0
554126	0	0	0	0	0	0	0	0	0
556917	1	0	1	0	0	0	0	1	0
556918	0	0	0	0	0	0	0	0	0
558536	0	0	0	0	0	0	0	0	0
558537	0	0	0	0	0	0	0	0	0
559919	0	0	0	0	0	0	0	0	0
559920	0	0	0	0	0	0	0	0	0
560033	0	0	0	0	0	0	0	0	0
560473	0	0	0	0	0	0	0	0	0
560491	0	0	0	0	0	0	0	1	0
561040	0	0	0	0	0	0	0	0	0
561228	0	0	0	0	0	0	0	0	0
563179	0	0	0	0	0	0	0	0	0
563614	0	0	0	0	0	0	0	1	0
565145	0	0	0	0	0	0	0	0	0
565146	0	0	0	0	0	0	0	0	0
565466	0	0	0	0	0	0	0	0	0
567085	0	0	0	0	0	0	0	0	0
568145	0	0	0	0	0	0	0	0	0
568687	0	0	0	0	0	0	0	0	0
568695	0	0	0	0	0	0	0	0	0
568708	0	0	0	0	0	0	0	0	0
569647	0	0	0	0	0	0	0	0	0
569650	0	0	0	0	0	0	0	0	0
569722	0	0	0	0	0	0	0	0	0
569723	0	0	0	0	0	0	0	0	0
574014	0	0	0	0	0	0	0	0	0
574138	0	0	0	0	0	0	0	0	0
574469	0	0	0	0	0	0	0	0	0
576394	0	0	0	0	0	0	0	0	0
576586	0	0	0	0	0	0	0	0	0
578459	0	0	0	0	0	0	0	0	0

57 rows × 607 columns

In [22]:
```python
#generate frequent itemsets and rules
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules
```

Out[22]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(RED RETROSPOT CAKE STAND)	(36 PENCILS TUBE RED RETROSPOT)	0.070175	0.070175	0.070175	1.000000	14.250000	0.065251	inf
1	(36 PENCILS TUBE RED RETROSPOT)	(RED RETROSPOT CAKE STAND)	0.070175	0.070175	0.070175	1.000000	14.250000	0.065251	inf
2	(SET OF 3 CAKE TINS PANTRY DESIGN)	(36 PENCILS TUBE RED RETROSPOT)	0.157895	0.070175	0.070175	0.444444	6.333333	0.059095	1.673684
3	(36 PENCILS TUBE RED RETROSPOT)	(SET OF 3 CAKE TINS PANTRY DESIGN)	0.070175	0.157895	0.070175	1.000000	6.333333	0.059095	inf
4	(SET OF 6 SOLDIER SKITTLES)	(4 TRADITIONAL SPINNING TOPS)	0.122807	0.087719	0.070175	0.571429	6.514286	0.059403	2.128655
...	...	...	...	...	...	...	...	...	...
795	(HOMEMADE JAM SCENTED CANDLES)	(DOLLY GIRL LUNCH BOX, ROSES REGENCY TEACUP AN...	0.087719	0.070175	0.070175	0.800000	11.400000	0.064020	4.649123
796	(ROSES REGENCY TEACUP AND SAUCER)	(DOLLY GIRL LUNCH BOX, HOMEMADE JAM SCENTED CA...	0.140351	0.070175	0.070175	0.500000	7.125000	0.060326	1.859649
797	(REGENCY CAKESTAND 3 TIER)	(DOLLY GIRL LUNCH BOX, HOMEMADE JAM SCENTED CA...	0.105263	0.070175	0.070175	0.666667	9.500000	0.062789	2.789474
798	(CIRCUS PARADE LUNCH BOX)	(DOLLY GIRL LUNCH BOX, HOMEMADE JAM SCENTED CA...	0.087719	0.070175	0.070175	0.800000	11.400000	0.064020	4.649123
799	(SPACEBOY LUNCH BOX)	(DOLLY GIRL LUNCH BOX, HOMEMADE JAM SCENTED CA...	0.105263	0.070175	0.070175	0.666667	9.500000	0.062789	2.789474

800 rows × 9 columns

In [23]:
```python
#filtering the dataframe. large lift (6) and high confidence (.8)
rules[ (rules['lift'] >= 6) &
       (rules['confidence'] >= 0.8) ]
```

Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(RED RETROSPOT CAKE STAND)	(36 PENCILS TUBE RED RETROSPOT)	0.070175	0.070175	0.070175	1.0	14.250000	0.065251	inf
1	(36 PENCILS TUBE RED RETROSPOT)	(RED RETROSPOT CAKE STAND)	0.070175	0.070175	0.070175	1.0	14.250000	0.065251	inf
3	(36 PENCILS TUBE RED RETROSPOT)	(SET OF 3 CAKE TINS PANTRY DESIGN)	0.070175	0.157895	0.070175	1.0	6.333333	0.059095	inf
5	(4 TRADITIONAL SPINNING TOPS)	(SET OF 6 SOLDIER SKITTLES)	0.087719	0.122807	0.070175	0.8	6.514286	0.059403	4.385965
6	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.105263	0.105263	0.105263	1.0	9.500000	0.094183	inf