In [3]:
```
# H0:연속적인 값이 임의적이다. H1: 연속적인 값이 임의적이 아니다 연관이있다.
import pandas as pd

data=['a','a','b','b','a','a','a','a','b','b','b']
from statsmodels.sandbox.stats.runs import runstest_1samp
data=pd.DataFrame(data,columns=['product'])
data['label']=data['product'].map({'a':0,'b':1})
runstest_1samp(data['label'],cutoff=.5, correction=True)
```

Out[3]:
```
(-1.2539054635675788, 0.20987636894228046)
```

In [ ]:
```
!pip install mlxtend
```

In [ ]:
```
from mlxtend.frequent_patterns import apriori,association_rules
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder

# 지지도: 얼마나 자주?

# 신뢰도: 얼마나 자주 함께?

# 향상도: 우연이 아닌 관계?
```

In [ ]:
```
dataset=[['Apple','Beer','Rice','Chicken',],
        ['Apple','Beer','Rice'],
        ['Apple','Beer'],
        ['Apple','Bananas'],
        ['Milk','Beer','Rice','Chicken'],
        ['Milk','Beer','Rice'],
        ['Milk','Beer'],
        ['Apple','Bananas']]
te=TransactionEncoder()
te_ary=te.fit_transform(dataset)
print(te.columns_)
df=pd.DataFrame(te_ary,columns=te.columns_)
apriori(df,min_support=0.6,use_colnames=True)
frequent_itemsets=apriori(df,min_support=0.3,use_colnames=True)
frequent_itemsets['length']=frequent_itemsets['itemsets'].apply(lambda x:len(x))
print(frequent_itemsets)
```

```
['Apple', 'Bananas', 'Beer', 'Chicken', 'Milk', 'Rice']
   support       itemsets  length
0    0.625        (Apple)       1
1    0.750         (Beer)       1
2    0.375         (Milk)       1
3    0.500         (Rice)       1
4    0.375  (Beer, Apple)       2
5    0.375   (Beer, Milk)       2
6    0.500   (Beer, Rice)       2
```

In [17]:
```
df=pd.read_csv("https://raw.githubusercontent.com/ADPclass/ADP_book_ver01/main/data/groceries.csv")
```

In [ ]:
```
df_split=df.iloc[:,0].str.split(',',expand=True)
df_split_ary=df_split.values
groceries=[]
for i in range(len(df_split_ary)):
    groceries.append(list(filter(None,df_split_ary[i])))
print(groceries)
```

In [ ]:
```
te=TransactionEncoder()
te_ary=te.fit_transform(groceries)
print(te.columns_)
df=pd.DataFrame(te_ary,columns=te.columns_)
apriori(df,min_support=0.6,use_colnames=True)
frequent_itemsets=apriori(df,min_support=0.01,use_colnames=True)
frequent_itemsets['length']=frequent_itemsets['itemsets'].apply(lambda x:len(x))
print(frequent_itemsets)
```

In [26]:
```
association_rules(frequent_itemsets,metric="confidence",min_threshold=0.3)
```

Out[26]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric
0	(beef)	(other vegetables)	0.052471	0.193512	0.019727	0.375969	1.942869	1.0	0.009574	1.292384	0.51217
1	(beef)	(root vegetables)	0.052471	0.109010	0.017389	0.331395	3.040058	1.0	0.011669	1.332612	0.70822
2	(beef)	(whole milk)	0.052471	0.255542	0.021253	0.405039	1.585018	1.0	0.007844	1.251271	0.38953
3	(berries)	(other vegetables)	0.033252	0.193512	0.010270	0.308869	1.596118	1.0	0.003836	1.166909	0.38632
4	(berries)	(whole milk)	0.033252	0.255542	0.011796	0.354740	1.388187	1.0	0.003299	1.153733	0.28925
...	...	...	...	...	...	...	...	...	...	...	..
120	(yogurt, soda)	(whole milk)	0.027354	0.255542	0.010474	0.382900	1.498382	1.0	0.003484	1.206381	0.34196
121	(yogurt, tropical fruit)	(whole milk)	0.029286	0.255542	0.015152	0.517361	2.024564	1.0	0.007668	1.542474	0.52133
122	(whole milk, tropical fruit)	(yogurt)	0.042302	0.139516	0.015152	0.358173	2.567255	1.0	0.009250	1.340679	0.63744
123	(yogurt, whipped/sour cream)	(whole milk)	0.020744	0.255542	0.010881	0.524510	2.052539	1.0	0.005580	1.565664	0.52366
124	(whole milk, whipped/sour cream)	(yogurt)	0.032235	0.139516	0.010881	0.337539	2.419361	1.0	0.006383	1.298921	0.60620

125 rows × 14 columns

In [27]:

```
rules=association_rules(frequent_itemsets,metric="lift",min_threshold=1)
rules['antecedent_len']=rules['antecedents'].apply(lambda x:len(x))
rules[(rules['antecedent_len']>=2)&(rules['confidence']>=0.4)&(rules['lift']>=3)]
```

Out[27]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric
420	(citrus fruit, root vegetables)	(other vegetables)	0.017694	0.193512	0.010372	0.586207	3.029300	1.0	0.006948	1.949012	0.68195
492	(tropical fruit, root vegetables)	(other vegetables)	0.021049	0.193512	0.012304	0.584541	3.020692	1.0	0.008231	1.941197	0.68333

✅ 1. 지지도 (Support)

정의:

> **A와 B가 동시에 등장하는 거래의 비율**
> (전체 거래 수 대비, A와 B가 함께 나타난 비율)

$$\text{Support}(A \Rightarrow B) = P(A \cap B)$$

해석:

- A와 B가 **얼마나 자주 함께 나타나는가**를 보여줌
- 너무 낮으면 → 통계적으로 의미 없는 규칙일 가능성 있음

예시:

- 예: "우유와 시리얼"의 지지도가 0.2 → 전체 거래의 20%에서 둘 다 구매됨

✅ 2. 신뢰도 (Confidence)

정의:

> **A가 발생했을 때, B도 발생할 조건부 확률**
> (즉, A를 산 사람이 B도 샀을 확률)

$$\text{Confidence}(A \Rightarrow B) = \frac{P(A \cap B)}{P(A)} = P(B|A)$$

해석:

- A를 산 고객 중에서, 얼마나 많은 사람이 B도 샀는지
- 마케팅, 추천 시스템에서 **"A 산 고객에게 B도 추천 가능성 높음"**을 의미

예시:

- Confidence = 0.8 → 우유 산 사람 중 80%가 시리얼도 샀음

✅ 3. 향상도 (Lift)

정의:

> **B가 독립적으로 발생할 확률 대비, A를 조건으로 B가 얼마나 더 많이 발생하는가**

$$\text{Lift}(A \Rightarrow B) = \frac{P(B|A)}{P(B)} = \frac{\text{Confidence}(A \Rightarrow B)}{P(B)}$$

해석:

- **1보다 크면**: A를 샀을 때 B를 살 가능성이 일반보다 높음 → **긍정적 연관**
- **1보다 작으면**: A와 B는 오히려 함께 잘 안 나옴 → **부정적 연관**
- **1이면**: A와 B는 독립 → **관련 없음**

예시:

- Lift = 1.5 → B 구매 확률이 A를 샀을 때 **1.5배** 높아짐
- Lift = 0.7 → 오히려 A 샀을 때 B는 **더 안 사는 경향**

✅ 표로 요약

지표	수식	의미 요약	값 해석
Support	$P(A \cap B)$	A와 B 동시에 나타날 확률	높을수록 규칙이 유의미
Confidence	$\frac{P(A \cap B)}{P(A)}$	A가 주어졌을 때 B의 확률	높을수록 신뢰할 만한 규칙
Lift	(\frac{P(B	A)}{P(B)})	독립적 확률 대비 얼마나 더 높은지