Q4 - Association Analysis

Consider the following market basket transactions shown in the Table below.

Transaction ID	Items Ordered
1	{Flour, Eggs, Bread}
2	{Soda, Coffee}
3	{Flour, Butter, Milk, Eggs}
4	{Bread, Eggs, Juice, Detergent}
5	{Bread, Milk, Eggs}
6	{Eggs, Bread}
7	{Detergent, Milk}
8	{Coffee, Soda, Juice}
9	{Butter, Juice, Bread}
10	{Milk, Bread, Detergent}

Import Libraries

```
from pprint import pprint
import itertools
import math
```

(a) How many items are in this data set? What is the maximum size of itemsets that can be extracted from this data set?

Transaction History

```
transactions = [(1, frozenset({'Flour', 'Eggs', 'Bread'})),
                (2, frozenset({'Soda', 'Coffee'})),
                (3, frozenset({'Flour', 'Butter', 'Milk', 'Eggs'})),
                (4, frozenset({'Bread', 'Eggs', 'Juice', 'Detergent'})),
                (5, frozenset({'Bread', 'Milk', 'Eggs'})),
                (6, frozenset({'Eggs', 'Bread'})),
                (7, frozenset({'Detergent', 'Milk'})),
                (8, frozenset({'Coffee', 'Soda', 'Juice'})),
                (9, frozenset({'Butter', 'Juice', 'Bread'})),
                (10, frozenset({'Milk', 'Bread', 'Detergent'})),
               ]
```

Count of Items

```
items = set()
for i, x in transactions:
    items.update(x)
print('Total items in the data set - {}'.format(len(items)))
print('The items are - {}'.format(items))
```

```
Total items in the data set - 9
The items are - {'Juice', 'Flour', 'Detergent', 'Eggs', 'Bread', 'Milk',
'Soda', 'Butter', 'Coffee'}
```

Itemsets with Support ≥ 1

```
itemsets = {}
for i in range(1, len(items)+1):
    for x in itertools.combinations(items, i):
        for _, y in transactions:
            if set(x).issubset(y):
                if frozenset(x) not in itemsets:
                    itemsets[frozenset(x)]=0
                itemsets[frozenset(x)]+=1
print('Total itemsets with support ≥ 1 are - {}'.format(len(itemsets)))
```

```
Total itemsets with support ≥ 1 are - 44
```

```
print('Maximum Size of Itemset - {}'.format(max([len(x) for x in
itemsets.keys()])))
```

```
Maximum Size of Itemset - 4
```

There are 9 unique items, and 44 itemsets with with support ≥ 1, with maximum number of items in an itemset being 4.

(b) What is the maximum number of association rules that can be extracted from this data (including rules that have zero support)?

```
cnt=math.pow(3,len(items))-math.pow(2,len(items)+1)+1
print('The maximum number of association rules that can be extracted are {:.0f}'.format(cnt))
```

```
The maximum number of association rules that can be extracted are 18660
```

The maximum number of association rules that can be generated using d distinct items is given by the following formula - $R = \sum_{k=1}^{d-1}[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}] = 3^d - 2^{d+1} + 1$. If we plugin $d = 9, R = 18660$.

(c) What is the maximum number of 2-itemsets that can be derived from this data set (including those have zero support)?

```
print('The maximum number of 2-itemsets that can be derived from this data set are {:.0f}'.format(
    math.factorial(len(items))/(math.factorial(len(items)-2)*2)))
```

```
The maximum number of 2-itemsets that can be derived from this data set are 36
```

The maximum number of 2-itemsets that can be derived using n distinct items is given by the following combination - $count = \binom{n}{2} = \frac{n!}{n!2!} = \frac{n(n-1)}{2}$. For 9 items, we get 36 2-itemsets.

(d) Find an itemset (of size 2 or larger) that has the largest support.

```
max_cnt=0
max_set=None
for x,cnt in itemsets.items():
    if len(x)>=2 and cnt>max_cnt:
        max_cnt=cnt
        max_set=x
print('The itemset of size 2 or larger with maximum support is -> {' + ',
'.join(max_set) + '}')
```

```
The itemset of size 2 or larger with maximum support is -> {Bread, Eggs}
```

We can calculate this by iterating over the all the itemsets, and checking itemsets which have support more than or equal to 2. We can then maintain a MAX variable to keep track of the set which has the maximum support. Using this approach we get the following itemset -> {Bread, Eggs} and its support is 4 or $\frac{4}{10} = 0.4$.

(e) Given minconf = 0.5, find two pairs of items, a and b, such that the rules {a} -> {b} and {b} -> {a} have the same confidence, and their confidence is greater than or equal to the minconf threshold.

```
s = set(itemsets.keys())
for x, y in itertools.combinations(s, 2):
    union = x.union(y)
    if union in itemsets and itemsets[x]==itemsets[y] and
itemsets[union]/itemsets[x]>=0.5:
        print('Confidence -> {:.2f}\t'.format(itemsets[union]/itemsets[x]),
              '[{}]'.format(', '.join(x)),
              '[{}]'.format(', '.join(y)))
```

```
Confidence -> 1.00   [Butter, Eggs] [Milk, Butter, Eggs]
Confidence -> 1.00   [Butter, Eggs] [Flour, Milk, Eggs]
Confidence -> 1.00   [Butter, Eggs] [Flour, Milk, Butter, Eggs]
Confidence -> 1.00   [Butter, Eggs] [Flour, Milk]
Confidence -> 1.00   [Butter, Eggs] [Flour, Butter, Eggs]
Confidence -> 1.00   [Butter, Eggs] [Flour, Butter]
```

```
Confidence -> 1.00   [Butter, Eggs] [Flour, Milk, Butter]
Confidence -> 1.00   [Butter, Eggs] [Milk, Butter]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Milk, Eggs]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Milk, Butter, Eggs]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Milk]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Butter, Eggs]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Butter]
Confidence -> 1.00   [Milk, Butter, Eggs] [Flour, Milk, Butter]
Confidence -> 1.00   [Milk, Butter, Eggs] [Milk, Butter]
Confidence -> 1.00   [Eggs, Detergent] [Bread, Juice, Eggs]
Confidence -> 1.00   [Eggs, Detergent] [Bread, Juice, Detergent]
Confidence -> 1.00   [Eggs, Detergent] [Juice, Eggs]
Confidence -> 1.00   [Eggs, Detergent] [Eggs, Juice, Detergent]
Confidence -> 1.00   [Eggs, Detergent] [Eggs, Bread, Detergent]
Confidence -> 1.00   [Eggs, Detergent] [Juice, Detergent]
Confidence -> 1.00   [Eggs, Detergent] [Eggs, Juice, Bread, Detergent]
Confidence -> 0.50   [Milk, Eggs] [Flour]
Confidence -> 0.50   [Milk, Eggs] [Bread, Milk]
Confidence -> 0.50   [Milk, Eggs] [Flour, Eggs]
Confidence -> 0.50   [Milk, Eggs] [Butter]
Confidence -> 1.00   [Flour] [Flour, Eggs]
Confidence -> 0.50   [Flour] [Butter]
Confidence -> 1.00   [Bread, Juice, Eggs] [Bread, Juice, Detergent]
Confidence -> 1.00   [Bread, Juice, Eggs] [Juice, Eggs]
Confidence -> 1.00   [Bread, Juice, Eggs] [Eggs, Juice, Detergent]
Confidence -> 1.00   [Bread, Juice, Eggs] [Eggs, Bread, Detergent]
Confidence -> 1.00   [Bread, Juice, Eggs] [Juice, Detergent]
Confidence -> 1.00   [Bread, Juice, Eggs] [Eggs, Juice, Bread, Detergent]
Confidence -> 1.00   [Juice, Coffee] [Juice, Soda, Coffee]
Confidence -> 1.00   [Juice, Coffee] [Juice, Soda]
Confidence -> 1.00   [Bread, Juice, Detergent] [Juice, Eggs]
Confidence -> 1.00   [Bread, Juice, Detergent] [Eggs, Juice, Detergent]
Confidence -> 1.00   [Bread, Juice, Detergent] [Eggs, Bread, Detergent]
Confidence -> 1.00   [Bread, Juice, Detergent] [Juice, Detergent]
Confidence -> 1.00   [Bread, Juice, Detergent] [Eggs, Juice, Bread,
Detergent]
Confidence -> 1.00   [Juice, Eggs] [Eggs, Juice, Detergent]
Confidence -> 1.00   [Juice, Eggs] [Eggs, Bread, Detergent]
Confidence -> 1.00   [Juice, Eggs] [Juice, Detergent]
Confidence -> 1.00   [Juice, Eggs] [Eggs, Juice, Bread, Detergent]
Confidence -> 1.00   [Eggs, Juice, Detergent] [Eggs, Bread, Detergent]
Confidence -> 1.00   [Eggs, Juice, Detergent] [Juice, Detergent]
Confidence -> 1.00   [Eggs, Juice, Detergent] [Eggs, Juice, Bread,
Detergent]
Confidence -> 1.00   [Juice, Soda, Coffee] [Juice, Soda]
Confidence -> 1.00   [Flour, Bread, Eggs] [Flour, Bread]
```

```
Confidence -> 1.00   [Eggs, Bread, Detergent] [Juice, Detergent]
Confidence -> 1.00   [Eggs, Bread, Detergent] [Eggs, Juice, Bread,
Detergent]
Confidence -> 1.00   [Flour, Milk, Eggs] [Flour, Milk, Butter, Eggs]
Confidence -> 1.00   [Flour, Milk, Eggs] [Flour, Milk]
Confidence -> 1.00   [Flour, Milk, Eggs] [Flour, Butter, Eggs]
Confidence -> 1.00   [Flour, Milk, Eggs] [Flour, Butter]
Confidence -> 1.00   [Flour, Milk, Eggs] [Flour, Milk, Butter]
Confidence -> 1.00   [Flour, Milk, Eggs] [Milk, Butter]
Confidence -> 1.00   [Soda, Coffee] [Soda]
Confidence -> 1.00   [Soda, Coffee] [Coffee]
Confidence -> 1.00   [Flour, Milk, Butter, Eggs] [Flour, Milk]
Confidence -> 1.00   [Flour, Milk, Butter, Eggs] [Flour, Butter, Eggs]
Confidence -> 1.00   [Flour, Milk, Butter, Eggs] [Flour, Butter]
Confidence -> 1.00   [Flour, Milk, Butter, Eggs] [Flour, Milk, Butter]
Confidence -> 1.00   [Flour, Milk, Butter, Eggs] [Milk, Butter]
Confidence -> 1.00   [Soda] [Coffee]
Confidence -> 1.00   [Bread, Juice, Butter] [Juice, Butter]
Confidence -> 1.00   [Bread, Juice, Butter] [Bread, Butter]
Confidence -> 0.50   [Bread, Juice] [Bread, Detergent]
Confidence -> 0.50   [Bread, Juice] [Butter]
Confidence -> 0.50   [Milk, Detergent] [Bread, Detergent]
Confidence -> 0.50   [Milk, Detergent] [Bread, Milk]
Confidence -> 0.50   [Bread, Detergent] [Bread, Milk]
Confidence -> 1.00   [Juice, Detergent] [Eggs, Juice, Bread, Detergent]
Confidence -> 0.50   [Flour, Eggs] [Butter]
Confidence -> 1.00   [Flour, Milk] [Flour, Butter, Eggs]
Confidence -> 1.00   [Flour, Milk] [Flour, Butter]
Confidence -> 1.00   [Flour, Milk] [Flour, Milk, Butter]
Confidence -> 1.00   [Flour, Milk] [Milk, Butter]
Confidence -> 1.00   [Juice, Butter] [Bread, Butter]
Confidence -> 1.00   [Flour, Butter, Eggs] [Flour, Butter]
Confidence -> 1.00   [Flour, Butter, Eggs] [Flour, Milk, Butter]
Confidence -> 1.00   [Flour, Butter, Eggs] [Milk, Butter]
Confidence -> 1.00   [Flour, Butter] [Flour, Milk, Butter]
Confidence -> 1.00   [Flour, Butter] [Milk, Butter]
Confidence -> 1.00   [Flour, Milk, Butter] [Milk, Butter]
```

The formula for confidence is given by $confidence(X->Y) = \frac{support(X \cup Y)}{support(X)}$. Given 2 itemsets, x and y, the confidence(x->y) and confidence(y->x) is equal if and only if the their individual supports are equal.

We listed out above all the possible pair of itemsets have a confidence ≥ 0.5 and have the same confidence.

If we assume a and b are items, then the question states to find 2 such pairs of 1-itemsets. From the above list these are [{Flour} {Butter}] and [{Soda}{Coffee}].

If we assume a and b are 2-itemsets, then the questions asks us to find a pair of 2-itemsets that satisfy the above condition. From the above list, this is [{Butter, Eggs}, {Flour, Milk}]