# Questão 1

In [142…
```
import pandas as pd
from apyori import apriori
```

In [143…
```
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

dataset
```

Out[143…
```
[['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
 ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
 ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
 ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
 ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]
```

In [144…
```
rules = apriori(dataset, min_support=0.6, min_confidence=0.7, min_lift=1.0, min_length=2)

results = list(rules)

results
```

Out[144…

```
[RelationRecord(items=frozenset({'Eggs'}), support=0.8, ordered_statistics=[OrderedStatistic(items_base=frozenset
(), items_add=frozenset({'Eggs'}), confidence=0.8, lift=1.0)]),
 RelationRecord(items=frozenset({'Kidney Beans'}), support=1.0, ordered_statistics=[OrderedStatistic(items_base=fr
ozenset(), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0)]),
 RelationRecord(items=frozenset({'Kidney Beans', 'Eggs'}), support=0.8, ordered_statistics=[OrderedStatistic(items
_base=frozenset(), items_add=frozenset({'Kidney Beans', 'Eggs'}), confidence=0.8, lift=1.0), OrderedStatistic(item
s_base=frozenset({'Eggs'}), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0), OrderedStatistic(ite
ms_base=frozenset({'Kidney Beans'}), items_add=frozenset({'Eggs'}), confidence=0.8, lift=1.0)]),
 RelationRecord(items=frozenset({'Onion', 'Eggs'}), support=0.6, ordered_statistics=[OrderedStatistic(items_base=f
rozenset({'Eggs'}), items_add=frozenset({'Onion'}), confidence=0.7499999999999999, lift=1.2499999999999998), Order
edStatistic(items_base=frozenset({'Onion'}), items_add=frozenset({'Eggs'}), confidence=1.0, lift=1.25)]),
 RelationRecord(items=frozenset({'Kidney Beans', 'Milk'}), support=0.6, ordered_statistics=[OrderedStatistic(items
_base=frozenset({'Milk'}), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0)]),
 RelationRecord(items=frozenset({'Kidney Beans', 'Onion'}), support=0.6, ordered_statistics=[OrderedStatistic(item
s_base=frozenset({'Onion'}), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0)]),
 RelationRecord(items=frozenset({'Kidney Beans', 'Yogurt'}), support=0.6, ordered_statistics=[OrderedStatistic(ite
ms_base=frozenset({'Yogurt'}), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0)]),
 RelationRecord(items=frozenset({'Kidney Beans', 'Onion', 'Eggs'}), support=0.6, ordered_statistics=[OrderedStatis
tic(items_base=frozenset({'Eggs'}), items_add=frozenset({'Kidney Beans', 'Onion'}), confidence=0.7499999999999999,
lift=1.2499999999999998), OrderedStatistic(items_base=frozenset({'Onion'}), items_add=frozenset({'Kidney Beans',
'Eggs'}), confidence=1.0, lift=1.25), OrderedStatistic(items_base=frozenset({'Kidney Beans', 'Eggs'}), items_add=f
rozenset({'Onion'}), confidence=0.7499999999999999, lift=1.2499999999999998), OrderedStatistic(items_base=frozense
t({'Onion', 'Eggs'}), items_add=frozenset({'Kidney Beans'}), confidence=1.0, lift=1.0), OrderedStatistic(items_bas
e=frozenset({'Kidney Beans', 'Onion'}), items_add=frozenset({'Eggs'}), confidence=1.0, lift=1.25)])]
```

In [145…

```
def inspect(results):
    formatted_rules = []
    for result in results:
        support = result[1]
        for ordered_stat in result[2]:
            lhs_items = tuple(ordered_stat.items_base)
            rhs_items = tuple(ordered_stat.items_add)
            if not lhs_items or not rhs_items:
                continue
            confidence = ordered_stat.confidence
            lift = ordered_stat.lift
            formatted_rules.append((lhs_items, rhs_items, support, confidence, lift))
    return formatted_rules

df_rules = pd.DataFrame(
```

```
    inspect(results),
    columns=['Se', 'Então', 'Suporte', 'Confiança', 'Lift']
)

df_rules
```

Out[145…

| | Se | Então | Suporte | Confiança | Lift |
|---|---|---|---|---|---|
| **0** | (Eggs,) | (Kidney Beans,) | 0.8 | 1.00 | 1.00 |
| **1** | (Kidney Beans,) | (Eggs,) | 0.8 | 0.80 | 1.00 |
| **2** | (Eggs,) | (Onion,) | 0.6 | 0.75 | 1.25 |
| **3** | (Onion,) | (Eggs,) | 0.6 | 1.00 | 1.25 |
| **4** | (Milk,) | (Kidney Beans,) | 0.6 | 1.00 | 1.00 |
| **5** | (Onion,) | (Kidney Beans,) | 0.6 | 1.00 | 1.00 |
| **6** | (Yogurt,) | (Kidney Beans,) | 0.6 | 1.00 | 1.00 |
| **7** | (Eggs,) | (Kidney Beans, Onion) | 0.6 | 0.75 | 1.25 |
| **8** | (Onion,) | (Kidney Beans, Eggs) | 0.6 | 1.00 | 1.25 |
| **9** | (Kidney Beans, Eggs) | (Onion,) | 0.6 | 0.75 | 1.25 |
| **10** | (Onion, Eggs) | (Kidney Beans,) | 0.6 | 1.00 | 1.00 |
| **11** | (Kidney Beans, Onion) | (Eggs,) | 0.6 | 1.00 | 1.25 |

# Questão 2

In [146…

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

In [147…

```
df = pd.read_csv('Sales_October_2019.csv')

df.head()
```

Out[147…

| | Order ID | Product | Quantity Ordered | Price Each | Order Date | Purchase Address |
|---|---|---|---|---|---|---|
| **0** | 259358 | 34in Ultrawide Monitor | 1 | 379.99 | 10/28/19 10:56 | 609 Cherry St, Dallas, TX 75001 |
| **1** | 259359 | 27in 4K Gaming Monitor | 1 | 389.99 | 10/28/19 17:26 | 225 5th St, Los Angeles, CA 90001 |
| **2** | 259360 | AAA Batteries (4-pack) | 2 | 2.99 | 10/24/19 17:20 | 967 12th St, New York City, NY 10001 |
| **3** | 259361 | 27in FHD Monitor | 1 | 149.99 | 10/14/19 22:26 | 628 Jefferson St, New York City, NY 10001 |
| **4** | 259362 | Wired Headphones | 1 | 11.99 | 10/07/19 16:10 | 534 14th St, Los Angeles, CA 90001 |

In [148…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20379 entries, 0 to 20378
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Order ID          20317 non-null  object
 1   Product           20317 non-null  object
 2   Quantity Ordered  20317 non-null  object
 3   Price Each        20317 non-null  object
 4   Order Date        20317 non-null  object
 5   Purchase Address  20317 non-null  object
dtypes: object(6)
memory usage: 955.4+ KB
```

In [149…
```
# quanto de linhas e colunas em branco
df.isnull().sum()
```

Out[149…
```
Order ID            62
Product             62
Quantity Ordered    62
Price Each          62
Order Date          62
Purchase Address    62
dtype: int64
```

In [150…
```
df.dropna(axis=0, subset=['Order ID'], inplace=True)
df.isnull().sum()
```

Out[150…
```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

In [151…
```
df['Quantity Ordered'] = pd.to_numeric(df['Quantity Ordered'], errors='coerce')
```

In [152…
```
basket = (df.groupby(['Order ID', 'Product'])['Quantity Ordered']
          .sum().unstack().reset_index().fillna(0)
          .set_index('Order ID'))
```

In [153…
```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.map(encode_units)

basket_sets.head()
```

Out[153…

| Order ID | Product | 20in Monitor | 27in 4K Gaming Monitor | 27in FHD Monitor | 34in Ultrawide Monitor | AA Batteries (4-pack) | AAA Batteries (4-pack) | Apple Airpods Headphones | Bose SoundSport Headphones | Flatscreen TV | Google Phone | LG Dryer | LG Washing Machine | Light Char C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 259358 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 259359 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 259360 | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 259361 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 259362 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

In [157…

```
min_support_val = 0.001
frequent_itemsets = apriori(basket_sets, min_support=min_support_val, use_colnames=True)

print(f"Itemsets frequentes encontrados: {len(frequent_itemsets)}")

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)

print("Top 10 Regras encontradas:")
cols = ['antecedents', 'consequents', 'support', 'confidence', 'lift']
rules[cols].sort_values(by='lift', ascending=False).head(10)
```

```
Itemsets frequentes encontrados: 27
Top 10 Regras encontradas:
/home/viniciusmonnerat/.local/lib/python3.14/site-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

Out[157...

| | antecedents | consequents | support | confidence | lift |
|---|---|---|---|---|---|
| **5** | (Vareebadd Phone) | (USB-C Charging Cable) | 0.002109 | 0.201970 | 1.610462 |
| **4** | (USB-C Charging Cable) | (Vareebadd Phone) | 0.002109 | 0.016817 | 1.610462 |
| **1** | (Google Phone) | (USB-C Charging Cable) | 0.005453 | 0.177554 | 1.415775 |
| **0** | (USB-C Charging Cable) | (Google Phone) | 0.005453 | 0.043478 | 1.415775 |
| **2** | (Lightning Charging Cable) | (iPhone) | 0.004835 | 0.039004 | 1.060476 |
| **3** | (iPhone) | (Lightning Charging Cable) | 0.004835 | 0.131469 | 1.060476 |