```
import pandas as pd
```

```
df=pd.read_csv('/content/Market_Basket_Optimisation.csv', header=None)
```

```
!pip install apyori
from apyori import apriori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5953 sha256=992568ec0cc46555a26bb75ef3019722206cc23fb50e45ba
  Stored in directory: /root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a8cb7c94d8ada15573538c7f4baebe2d
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```
df
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmor
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7501 rows × 20 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxy j
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.isnull().sum()
```

	0
0	0
1	1754
2	3112
3	4156
4	4972
5	5637
6	6132
7	6520
8	6847
9	7106
10	7245
11	7347
12	7414
13	7454
14	7476
15	7493
16	7497
17	7497
18	7498
19	7500

```
df.fillna(0,inplace=True)
```

```
df.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxy j
1	burgers	meatballs	eggs	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	chutney	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	turkey	avocado	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	mineral water	milk	energy bar	whole wheat rice	green tea	0	0	0	0	0	0	0	0	0	0	0	

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
transactions = []
for i in range(0, 7501):
  transactions.append([str(df.values[i,j]) for j in range(0, 20)])
transactions[0]
```

```
['shrimp',
 'almonds',
 'avocado',
 'vegetables mix',
 'green grapes',
 'whole weat flour',
 'yams',
 'cottage cheese',
 'energy drink',
 'tomato juice',
 'low fat yogurt',
 'green tea',
 'honey',
 'salad',
 'mineral water',
 'salmon',
 'antioxydant juice',
 'frozen smoothie',
```

```
 'spinach',
 'olive oil']
```

```
rule_list = apriori(transactions, min_support = 0.003, min_lift = 3, min_length = 2)
rule_list
```

```
<generator object apriori at 0x7d50a2d57ca0>
```

```
Results = list(rule_list)
```

```
print(len(Results))
```

```
188
```

```
results = pd.DataFrame(Results)
```

```
results.head()
```

	items	support	ordered_statistics
0	(cottage cheese, brownies)	0.003466	[((brownies), (cottage cheese), 0.102766798418...
1	(chicken, light cream)	0.004533	[((chicken), (light cream), 0.0755555555555555...
2	(escalope, mushroom cream sauce)	0.005733	[((escalope), (mushroom cream sauce), 0.072268...
3	(escalope, pasta)	0.005866	[((escalope), (pasta), 0.07394957983193277, 4....
4	(tomato juice, fresh bread)	0.004266	[((fresh bread), (tomato juice), 0.09907120743...

Next steps: Generate code with results | View recommended plots | New interactive sheet

```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

```
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df = pd.DataFrame(te_ary, columns=te.columns_)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_c
  and should_run_async(code)
```

```
frequent_itemsets = apriori(df, min_support=0.003, use_colnames=True)

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=3)

antecedents = rules['antecedents'].apply(lambda x: list(x))
consequents = rules['consequents'].apply(lambda x: list(x))

df_rules = pd.DataFrame({'antecedents': antecedents,'consequents': consequents,'support': rules['support']
})
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_c
  and should_run_async(code)
```

```
df_rules.head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_c
  and should_run_async(code)
```

	antecedents	consequents	support
0	[cottage cheese]	[brownies]	0.003466
1	[brownies]	[cottage cheese]	0.003466
2	[chicken]	[light cream]	0.004533
3	[light cream]	[chicken]	0.004533
4	[escalope]	[mushroom cream sauce]	0.005733

Next steps: Generate code with df_rules | View recommended plots | New interactive sheet

Start coding or generate with AI.

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_c
  and should_run_async(code)
```