

**Objectives**

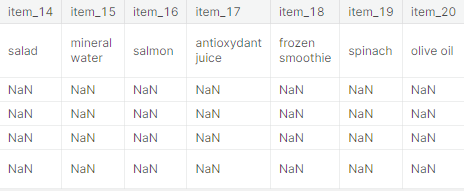
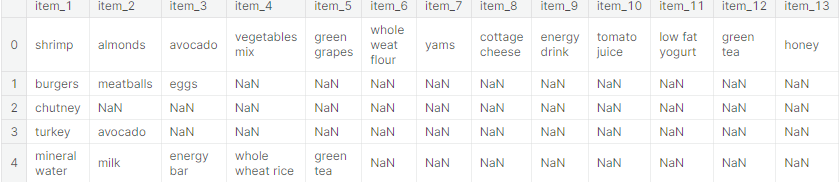
* **Check data quality.**
* **Use exploratory data analysis to derive insights on product performance.**
* **Apply association-rule-mining to discover opportunities for cross-selling.**
* import pandas as pd
* import matplotlib as mpl
* import seaborn as sns
* from matplotlib.axes import Axes
* sns.set\_palette("autumn")
* mpl.rc("axes", titlesize=18, titlepad=15, titleweight=500)
* mpl.rc("axes.spines", right=False, top=False)
* mpl.rc("figure", figsize=(10, 5.5))
* mpl.rc("font", family="serif", size=10)
* def annotate\_column\_chart(ax: Axes) -> Axes:
* *"""Add annotations to a column chart.*
* for p **in** ax.patches:
* p.set\_width(0.7)
* ax.annotate(f"**{**p.get\_height()**:**,**}**", ha="center",
* xy=(p.get\_x() + p.get\_width() / 2, p.get\_height() \* 1.01))
* return ax
* data = pd.read\_csv(
* header=None,
* names=[f"item\_**{**idx**}**" for idx **in** range(1, 21)]
* )

print(

)

data.head()

There were a total of 7,501 transactions, each containing between 1 and 20 items.



# 2. Data Cleaning

**One instance of the item "asparagus" contains leading whitespace. Other than that, the data looks fine.**



# 3. Exploratory Data Analysis

## 3.1 Best-selling products

**Assuming that only one unit of each item was bought in each transaction, mineral water is the most purchased product.**

**The top selling products are primarily food-stuff, but that's not at all surprising.**



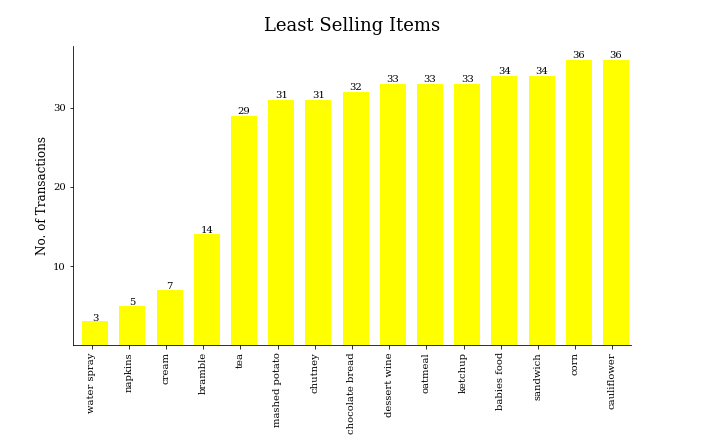


## 3.2 Worst performing products

**Assuming that only one unit of each item was bought in each transaction, water spray is sold the least.**

**It is quite unusual that the tea, chocolate bread and sandwiches are doing badly. This is worth investigating. Assuming this sample adequately captures the actual situation, then these products should probably be reviewed.**

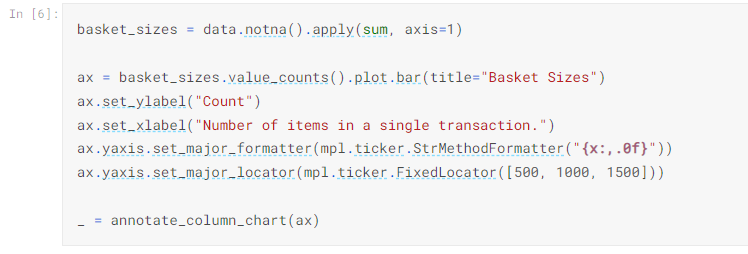


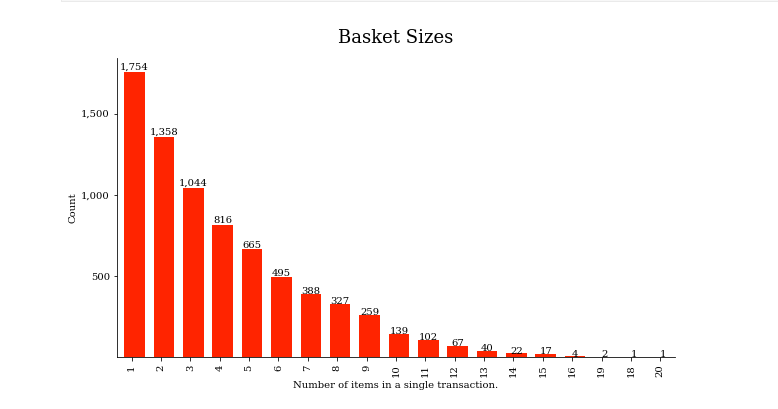


## 3.3 Distribution of Basket sizes

**The average basket-size was about 4 items. The largest transaction consisted of 20 items, and the smallest had just one.**

**Majority of the transactions involved a single item**.

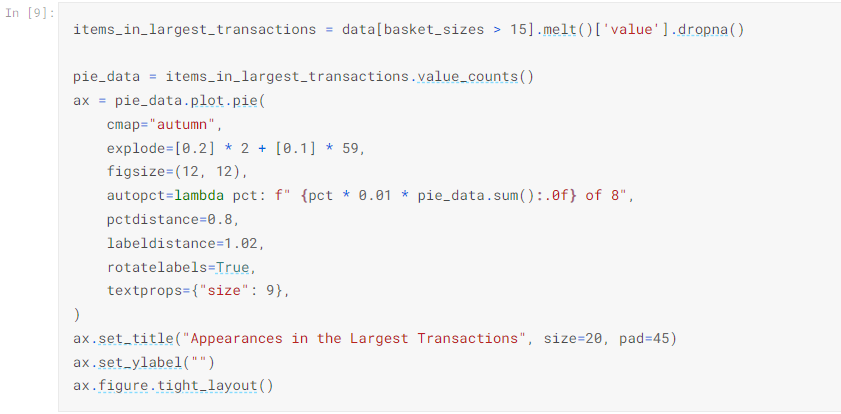


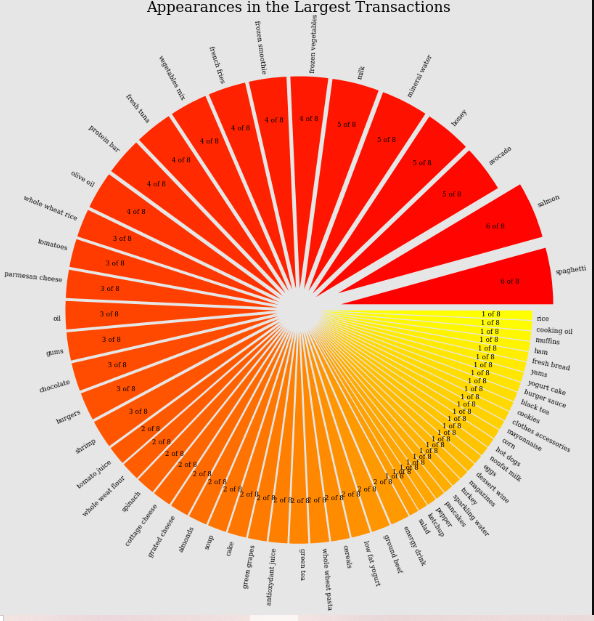


## 3.4 What's in the largest transactions?

**We'll consider transactions having more than 15 items (75% of maximum=20) as "large". There are 8 such transactions (16, 16, 16, 16, 18, 19, 19, 20).**

**Spaghetti and salmon are in 6 out of the eight largest transactions. Salmon's case is more striking, since we've already seen that spaghetti is the 3rd best seller. At face value, this might imply that customers who purchase a lot of items are more likely to buy salmon, so placing it next to the large trolleys/shopping-baskets might boost sales. But 8 out of 7501 cases doesn't inspire much confidence.**





## 4.1 Preprocessing

**Data input to the efficient-apriori.apriori function is required as a sequence of "baskets" e.g. a list of tuples containing items.**

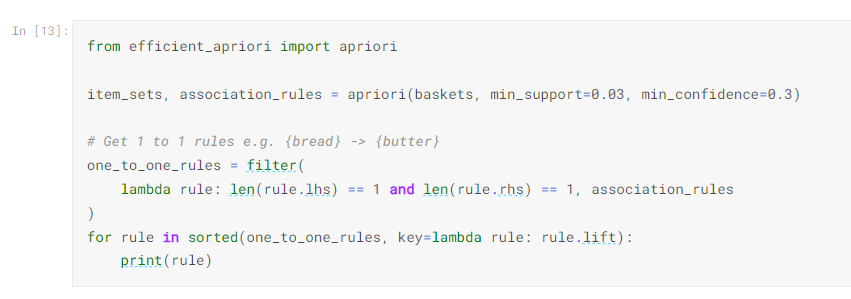
**In order to find item relationships, the baskets must include more than 1 item. We'll need to discard singleton transactions.**

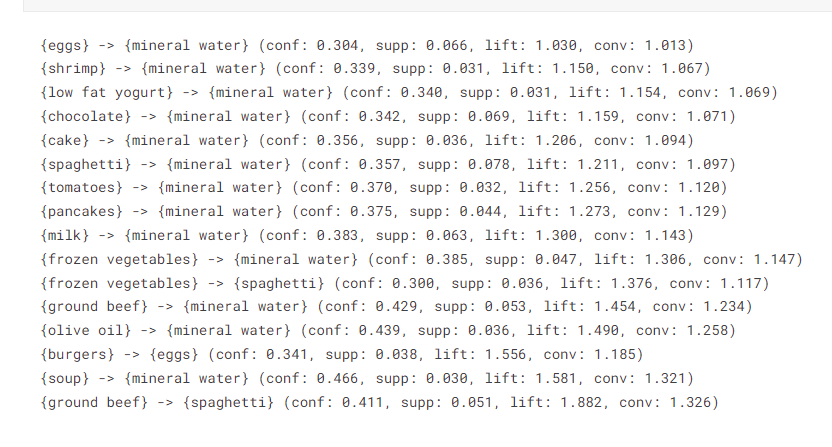
# 

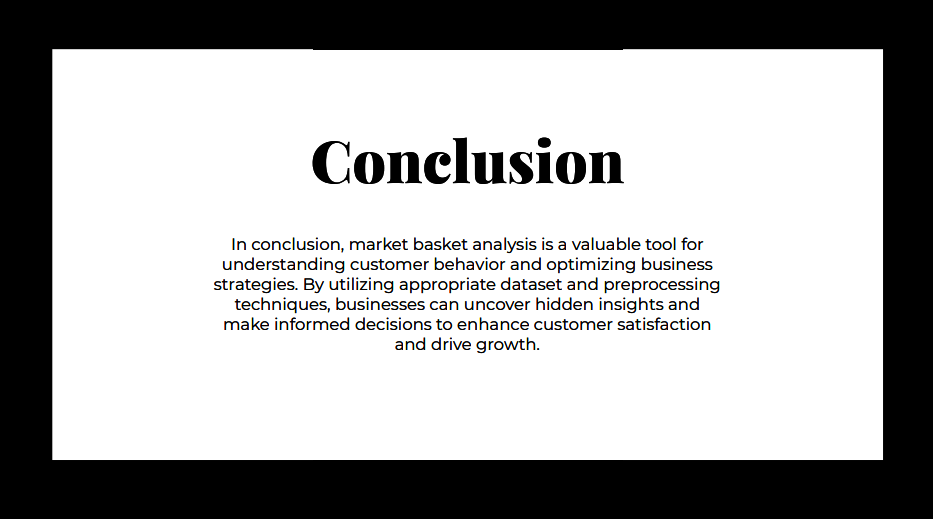
## 4.2 Association rules

**Potential opportunities for cross-selling are:**

* **frozen vegetables & spaghetti**
* **burgers & eggs**
* **ground beef & spaghetti**







**Presented by,**

**M . YOGAPPRIYA**