#Association Rules

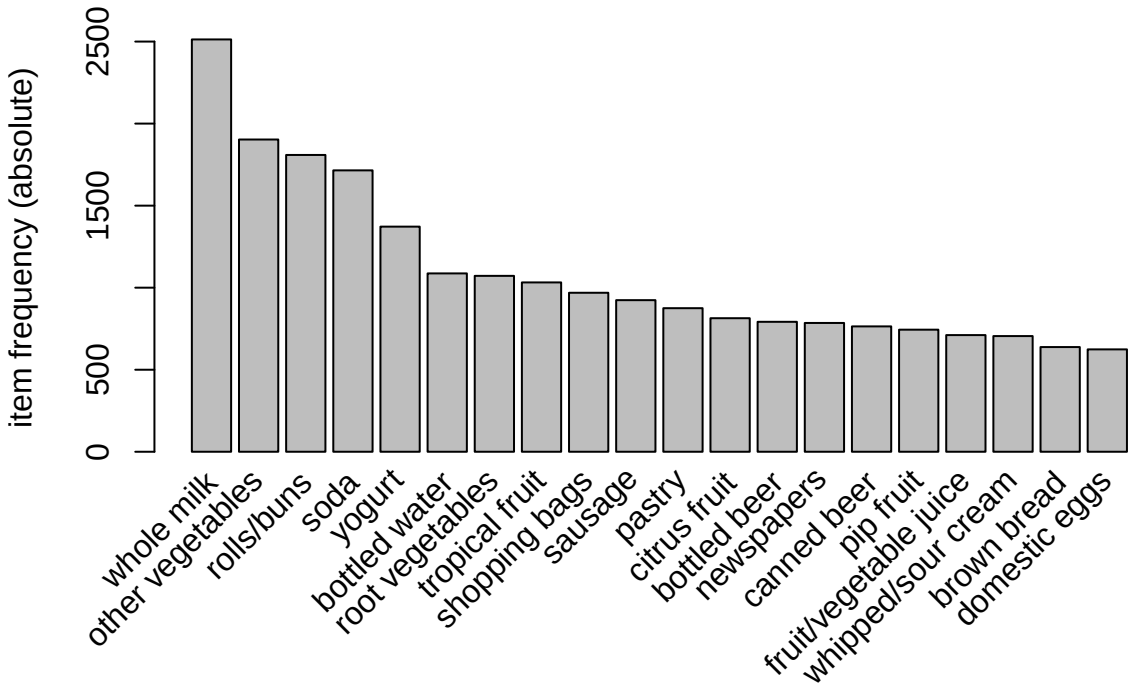
```
library(readr)
library(arules)
library(arulesViz)
library(dplyr)
library(ggplot2)

transactions <- read.transactions("/Users/samchen/Downloads/groceries.txt",
                                  format = "basket", sep = ",")

summary(transactions)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
##   17   18   19   20   21   22   23   24   26   27   28   29   32
##   29   14   14    9   11    4    6    1    1    1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```

```
itemFrequencyPlot(transactions, topN = 20, type = "absolute")
```



Here is a basic frequency plot of the various different items that you can buy at the grocery store. As can be seen, there are items that take up a large proportion of the entire data set.

```
rules <- apriori(transactions, parameter = list(supp = 0.001, conf = 0.1,
                                                maxlen = 5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.01s].
## writing ... [32731 rule(s)] done [0.00s].
```

```
## creating S4 object  ... done [0.00s].
```

The minimum support count is 9, as in there has to be at least 9 connections in the data set between those two particular items for it to register.

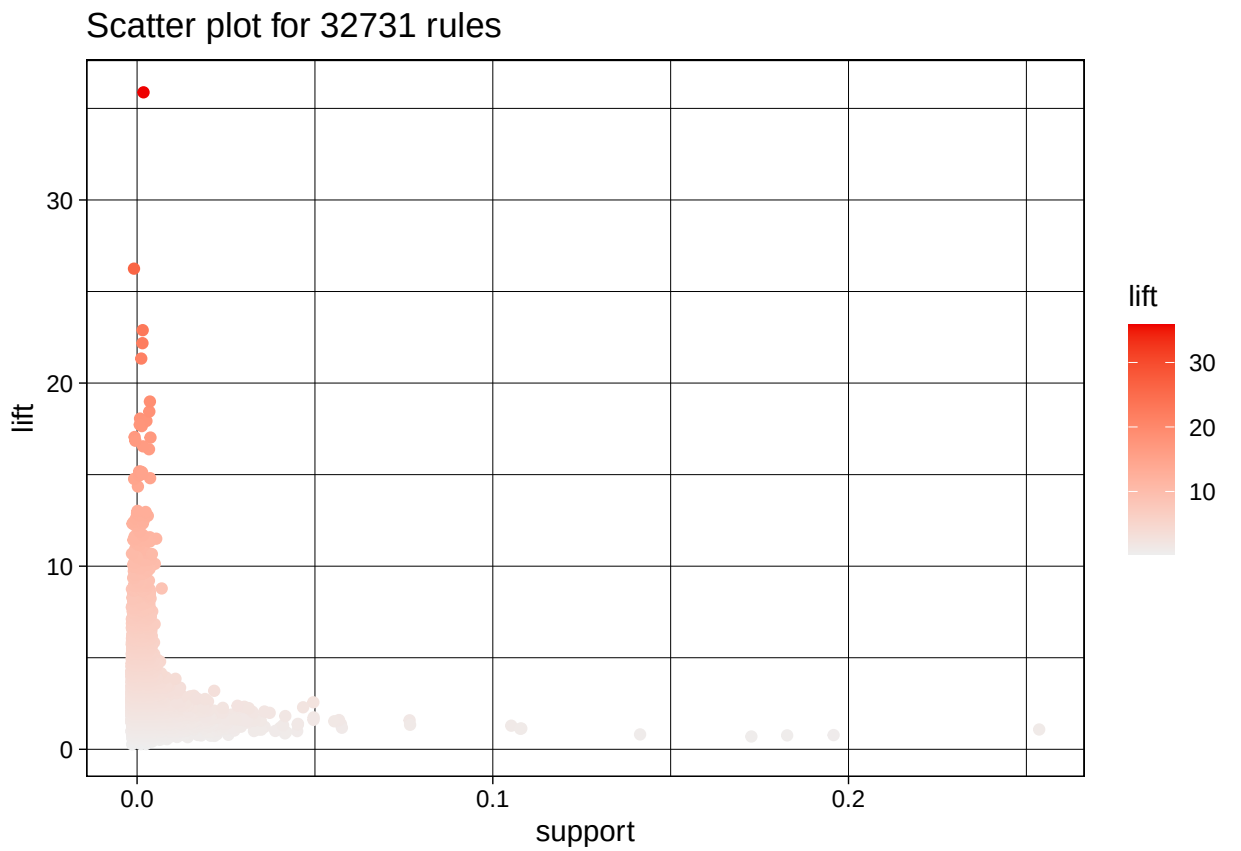
```
summary(rules)
```

```
## set of 32731 rules
##
## rule length distribution (lhs + rhs):sizes
##     1     2     3     4     5
##     8  2121 16468 12254  1880
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.424   4.000   5.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001017   Min.   :0.1000   Min.   :0.001017   Min.   : 0.4193
##  1st Qu.:0.001118   1st Qu.:0.1690   1st Qu.:0.003457   1st Qu.: 1.9568
##  Median :0.001423   Median :0.2603   Median :0.005897   Median : 2.5734
##  Mean   :0.002071   Mean   :0.3112   Mean   :0.009401   Mean   : 2.8339
##  3rd Qu.:0.002034   3rd Qu.:0.4167   3rd Qu.:0.009863   3rd Qu.: 3.3955
##  Max.   :0.255516   Max.   :1.0000   Max.   :1.000000   Max.   :35.7158
##      count
##  Min.   :  10.00
##  1st Qu.:  11.00
##  Median :  14.00
##  Mean   :  20.37
##  3rd Qu.:  20.00
##  Max.   :2513.00
##
## mining info:
##          data ntransactions support confidence
##  transactions          9835   0.001        0.1
##                                                                                call
##  apriori(data = transactions, parameter = list(supp = 0.001, conf = 0.1, maxlen = 5))
```

Just from a basic summary, we can see that whole milk, other vegetables, rolls/buns, soda, and yogurt are the most represented/bought items in the data set. Since there are so many distinct items in the data set, it makes sense to just keep support threshold low at 0.1%.

```
plot(rules, method = 'scatterplot', measure = c('support', 'lift'),
     shading = 'lift')
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Interesting, as your support decreases, the lift seems to increase. Based on this relationship, we can see that the support threshold that we chose is pretty valid in the sense that we would be losing a lot of information if the support threshold was pushed higher.

```
#find out which rules have positive associations
filtered_rules <- subset(rules, lift > 1)

inspect(filtered_rules[1:10,])
```

```
##      lhs                   rhs          support     confidence coverage
## [1]  {}                 => {yogurt}     0.139501779 0.1395018  1.000000000
## [2]  {}                 => {rolls/buns} 0.183934926 0.1839349  1.000000000
## [3]  {honey}            => {whole milk} 0.001118454 0.7333333  0.001525165
## [4]  {soap}             => {whole milk} 0.001118454 0.4230769  0.002643620
## [5]  {tidbits}          => {soda}       0.001016777 0.4347826  0.002338587
## [6]  {tidbits}          => {rolls/buns} 0.001220132 0.5217391  0.002338587
## [7]  {cocoa drinks}     => {whole milk} 0.001321810 0.5909091  0.002236909
## [8]  {snack products}   => {soda}       0.001118454 0.3666667  0.003050330
## [9]  {snack products}   => {rolls/buns} 0.001118454 0.3666667  0.003050330
## [10] {pudding powder}   => {whole milk} 0.001321810 0.5652174  0.002338587
##      lift     count
## [1]  1.000000 1372
## [2]  1.000000 1809
## [3]  2.870009   11
## [4]  1.655775   11
## [5]  2.493345   10
```

```
## [6]  2.836542   12
## [7]  2.312611   13
## [8]  2.102721   11
## [9]  1.993459   11
## [10] 2.212062   13
```

Looking at the top rules or connections, yogurt, soda, bottled water, whole milk, and other basic essentials are bought a lot on their own. The lifts of these rows flat line at 1 as they are practically connected with themselves.. Outside of that there is a strong drop off in support to the other item pairs in the list. This makes sense because both need to be present for the pair to be listed. What is is interesting is that there is a high variability of unique pairs even when customers are buying groceries. There are actually 32415 unique one direction rules in the data set.

Thresholds that we choose.

Support: given the sheer number of unique combinations in the data set, a threshold above 0.001 would really isolate pretty much every rule except for the rules with lift of 1.

Lift: We define lift of 18 as our threshold because wanted to see the top 0.1% (18 is around 99.9th percentile) of grocery item relationships.

Confidence: Choosing confidence at 0.5 means that the conditional has to be at least 0.5 to show that there is a strong confidence in our inference.

We think it also makes sense to look at which of these rules have the strongest lift. In this case, we will define high lift as lift > 8. Of the defined rules, we really want to see the rules that occupy the very top 0.1% of all rules to be concise.

```
#find very strong lift rules
filtered_rules_20 <- subset(rules, lift > 18)
filtered_rules_20_df <- inspect(filtered_rules_20)
```

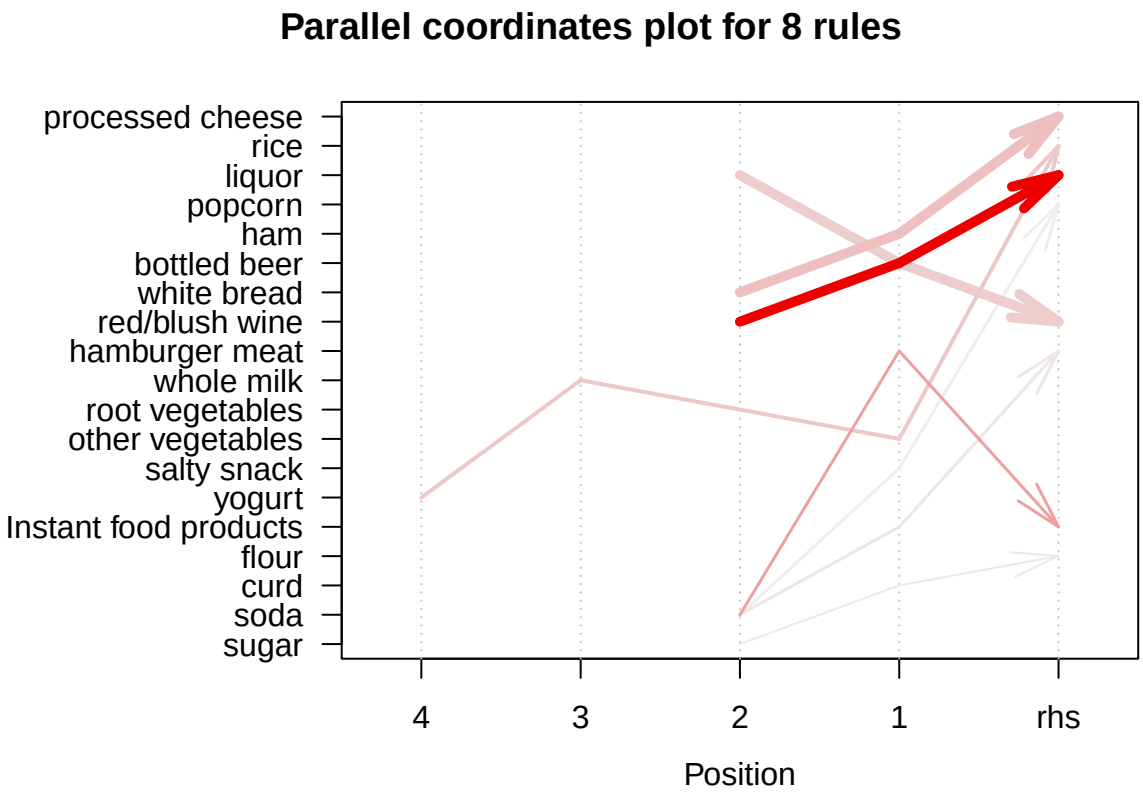
```
##     lhs                          rhs                          support confidence    coverage     lift
## [1] {bottled beer,
##      liquor}                  => {red/blush wine}         0.001931876  0.4130435 0.004677173 21.49356
## [2] {bottled beer,
##      red/blush wine}          => {liquor}                 0.001931876  0.3958333 0.004880529 35.71579
## [3] {salty snack,
##      soda}                    => {popcorn}                0.001220132  0.1304348 0.009354347 18.06797
## [4] {Instant food products,
##      soda}                    => {hamburger meat}         0.001220132  0.6315789 0.001931876 18.99565
## [5] {hamburger meat,
##      soda}                    => {Instant food products} 0.001220132  0.2105263 0.005795628 26.20919
## [6] {ham,
##      white bread}             => {processed cheese}       0.001931876  0.3800000 0.005083884 22.92822
## [7] {curd,
##      sugar}                   => {flour}                  0.001118454  0.3235294 0.003457041 18.60767
## [8] {other vegetables,
##      root vegetables,
##      whole milk,
##      yogurt}                  => {rice}                   0.001321810  0.1688312 0.007829181 22.13939
```

```
filtered_rules_20_df
```

```
## NULL
```

There are some very intuitive relationships represented in our high lift rule set. Buying beer and wine leads to buying liquor. Buying ham and processed cheese leads to buying bread, to make sandwiches. Buying a salty snack and soda leads to buying popcorn - for movies nights. High lift rules represent combos of items that are bought together specifically to pair with one another. Specifically, the practical interpretation of high lift rules is that there is a highly likelihood the right hand side of the rule is going to happen given the the presence of the left hand side of the rule.

```
plot(filtered_rules_20, method = "paracoord", control = list(reorder = TRUE))
```



Here we have a parallel coordinate graph. This graph shows the likely relationships and positions of different item pairings. For example, in the data frame with lift > 18, soda is likely to be on the left hand side of hamburger meat and hamburger meat is also likely to be on the left hand side of instant food products. The strongest line represents the strongest connection of all - that being the basketed purchases of beer, wine, and liquor. All of these combinations make sense.

```
filtered_rules_conf <- subset(rules,lift > 15 & confidence > 0.5)
inspect(filtered_rules_conf)
```

```
##     lhs                               rhs              support     confidence
## [1] {popcorn, soda}                => {salty snack}    0.001220132 0.6315789
## [2] {Instant food products, soda} => {hamburger meat} 0.001220132 0.6315789
## [3] {ham, processed cheese}        => {white bread}    0.001931876 0.6333333
## [4] {baking powder, flour}         => {sugar}          0.001016777 0.5555556
##     coverage    lift     count
## [1] 0.001931876 16.69779 12
```

```
## [2] 0.001931876 18.99565 12
## [3] 0.003050330 15.04549 19
## [4] 0.001830198 16.40807 10
```

Once we filter again for these relationships for a confidence of at least 0.5, we find the above to be the high lift and high confidence items pairs. The pairings make sense, and again are associated with paired items for specific dishes or activities. This time, sugar –> flour is replaced with baking power, flour –> sugar, the relationships seems to be more specific.

Overall, the network we produced with the thresholds that we defined above worked well to some of the relationships with basic grocery item pairings as well as pairings from very specific grocery items. This shows that consumers tend to buy things in conjunction in order to satisfy certain experiences and needs that cannot be achieved with just one item.