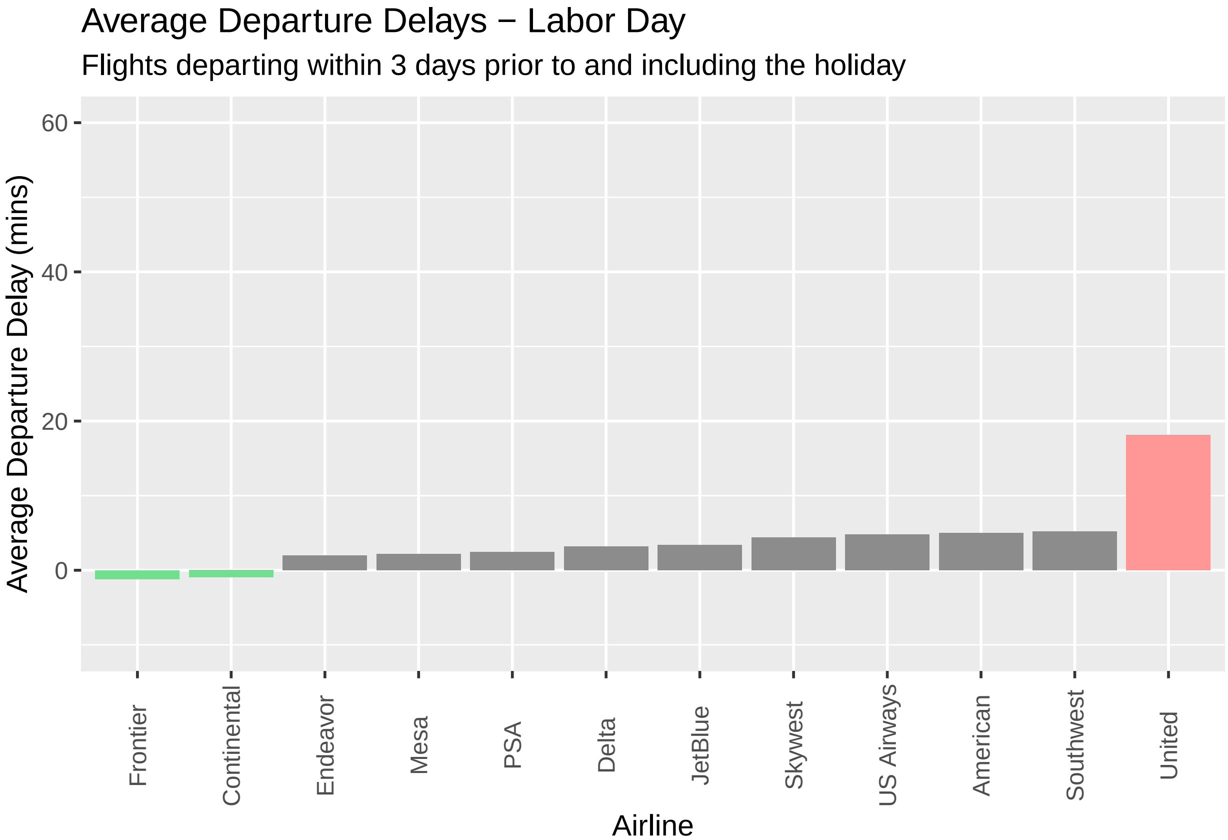
Pred Modeling Part 2 - Exercises 2

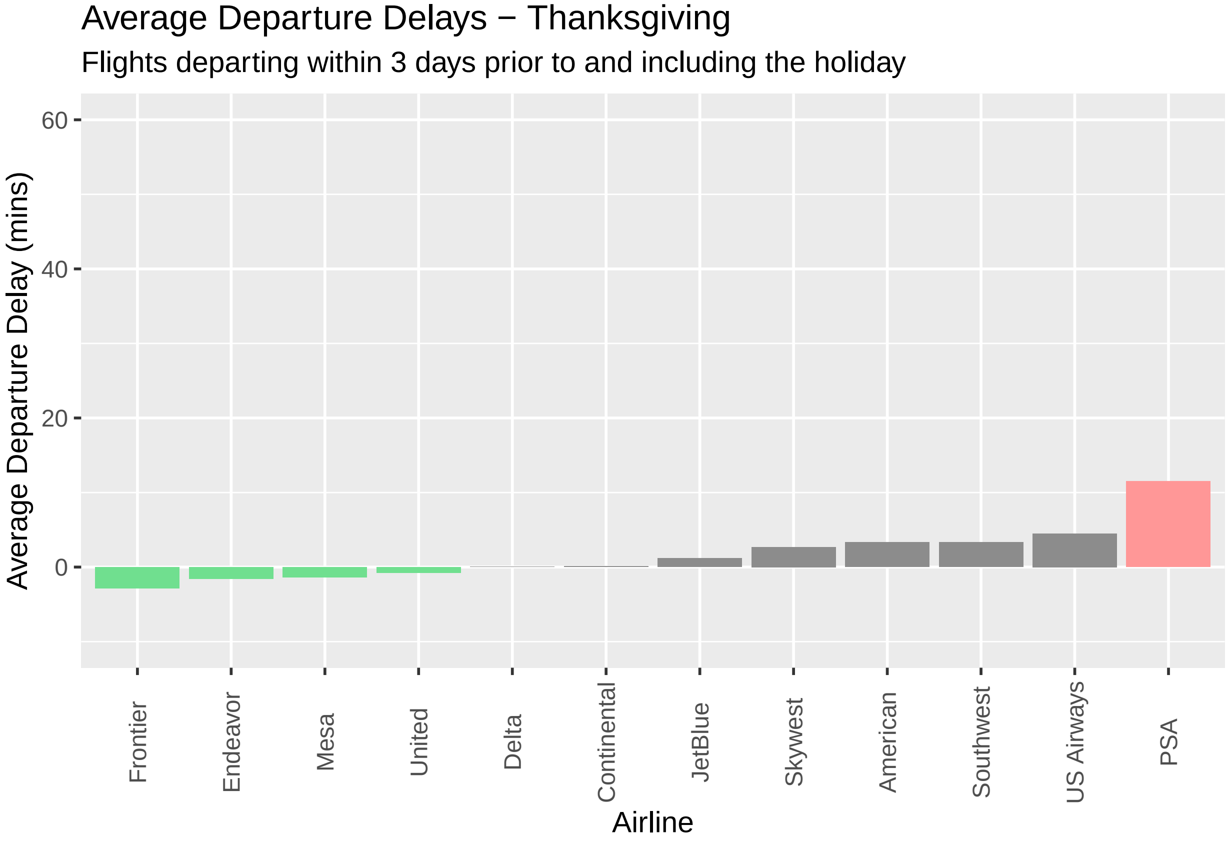
Ryan Hoff, Won Lee, Sam Malcolm, Cory Nguyen

8/20/2018

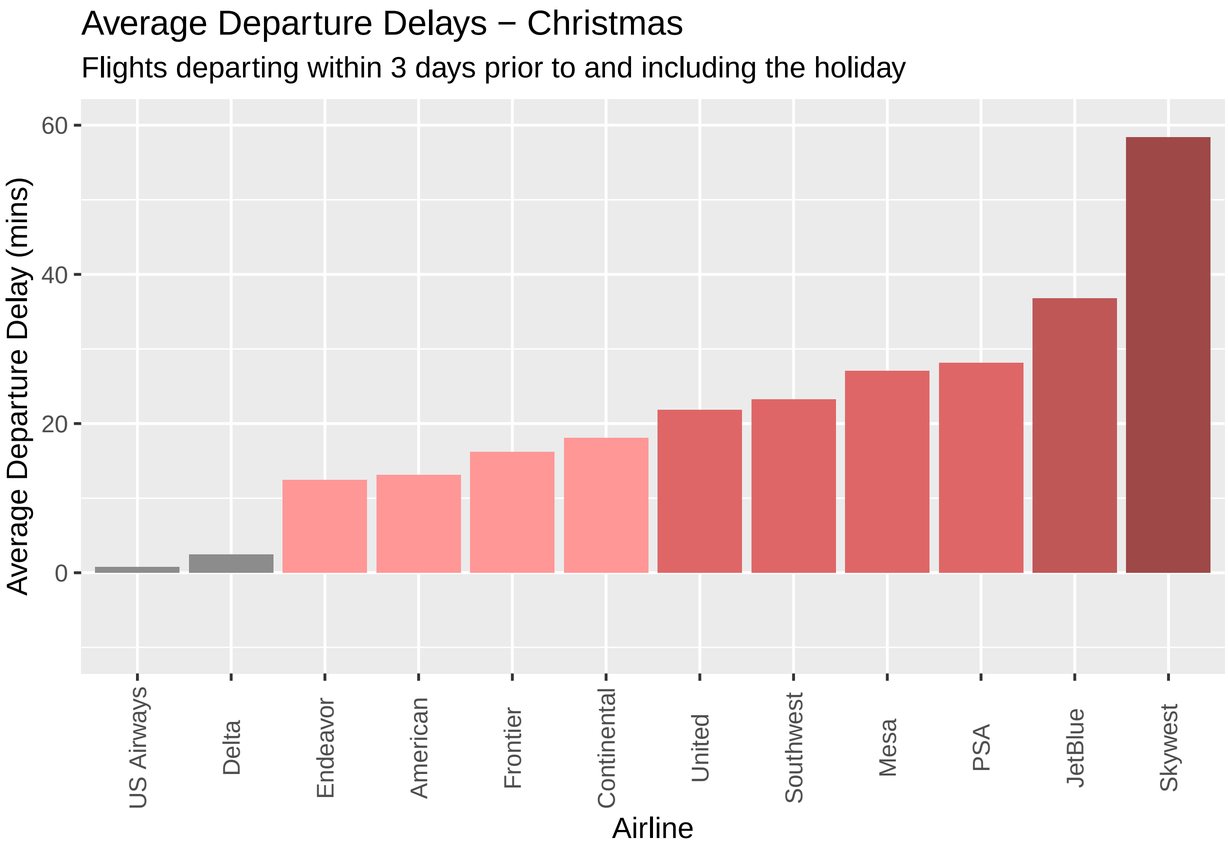
1: Flights at ABIA


Average Departure Delays − Labor Day

Flights departing within 3 days prior to and including the holiday


Average Departure Delays − Thanksgiving

Flights departing within 3 days prior to and including the holiday



Which airlines experience the longest departure delays during travel holidays?

We wanted to find the airlines that experienced the most and least delays around the holidays. We first tested which holidays experienced statistically significant departure delays. We then plotted the mean departure delays by airline for these holidays. We experimented with representing other statistics such as median departure delays (to help offset effect of outliers) and rate of flights delayed (to show how likely you were to experience a delay at all). However, we felt that mean departure delay still captured that information well for each airline (especially in proportion to one another). We also determined that the most utility would come from knowing if you would experience a major delay, so the magnitude of delay is important to represent. We felt this was an informative but interpretable way to portray it.

2. - Author Attribution

Naive Bayes : 63.4% out-of-sample accuracy

PCR: 63.8% out-of-sample accuracy

The two author attribution models that we built were based on the Naïve Bayes classifier and principal components regression using ridge regression. Based on the nature of the data (training and test sets had dimensions of at least 2500 x 3000), our first instinct was to choose a quick but reliable model to help us find a baseline accuracy to improve on. Since there were 50 authors in the data, we first chose to use a Multiple Naïve Bayes model. This model is both quick and reliable, and comes with the advantage of being able to handle multiple classes. The out-of-sample accuracy of this model is 63.4%.

Our next choice for model fitting was principle components regression using ridge regression. Since the number of features exceeded 3000, we wanted to employ a model that implemented dimensionality reduction.

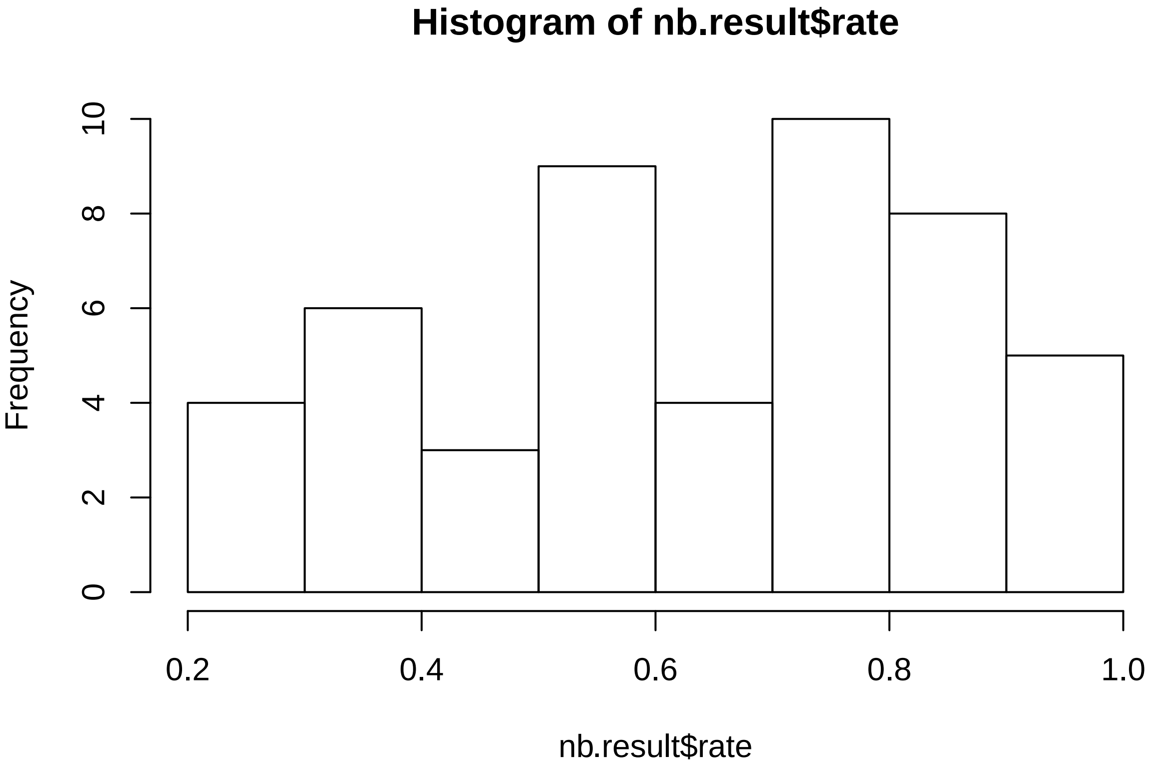
Our use of PCR completes this goal twofold, once when using PCA, and again when using ridge regression to “zero out” unimportant features when building a model for each author. This method will reduce the impact of noise on the models. The out-of-sample accuracy of this model when using the first 600 principal components is 63.8%, barely edging out Naïve Bayes.

In comparing these models, their accuracies are about the same. Naïve Bayes will run faster, but each author’s PCR model can be further analyzed to decide if they have patterns in their writing based on the significance of certain loadings in PCR.

It’s interesting that - at least with these two models - the ability to predict certain authors is fairly consistent. There are some differences, but the authors that were the most and least predictable were generally consistent. There were even a few that the models were very successful at predicting.

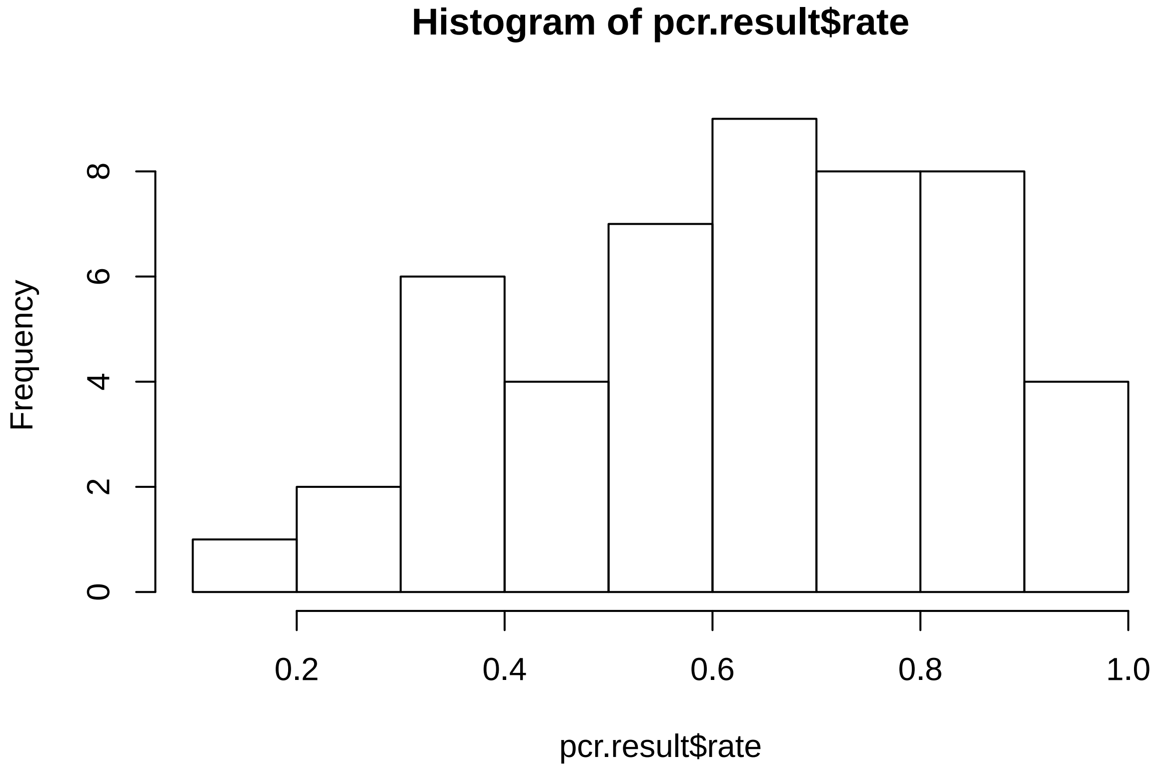
```
## [1] 0.634
```

```
## [1] 0.638
```



```
##               author rate
## 11    FumikoFujisaki 1.00
## 16      JimGilchrist 1.00
## 29   LynnleyBrowning 1.00
## 21       KarlPenhaul 0.94
## 1      AaronPressman 0.92
## 36         NickLouth 0.90
## 33      MatthewBunce 0.88
## 40        RobinSidel 0.88
## 12    GrahamEarnshaw 0.86
## 26 KouroshKarimkhany 0.86
```

```
##             author rate
## 10    EricAuchard 0.40
## 3   AlexanderSmith 0.38
## 44      ScottHillis 0.36
## 13 HeatherScoffield 0.34
## 4   BenjaminKangLim 0.32
## 35       MureDickie 0.32
## 7  DarrenSchuettler 0.28
## 46         TanEeLyn 0.24
## 8       DavidLawder 0.22
## 15    JaneMacartney 0.20
## 50             <NA>   NA
```



```
##             author rate
## 16    JimGilchrist 1.00
## 21     KarlPenhaul 1.00
## 29 LynnleyBrowning 1.00
## 11  FumikoFujisaki 0.98
## 22       KeithWeir 0.90
## 12  GrahamEarnshaw 0.88
## 33    MatthewBunce 0.88
## 34   MichaelConnor 0.88
## 1    AaronPressman 0.86
## 36       NickLouth 0.86
```

```
##              author rate
## 14       JanLopatka 0.44
## 10      EricAuchard 0.40
## 15    JaneMacartney 0.38
## 8       DavidLawder 0.36
## 35       MureDickie 0.36
## 49       ToddNissen 0.36
## 13 HeatherScoffield 0.34
## 7  DarrenSchuettler 0.26
## 4   BenjaminKangLim 0.24
## 44      ScottHillis 0.18
## 50             <NA>   NA
```

3. - Association Rule Mining

First, we prepared the data for apriori by stacking it in a matter like the example playlists.csv. We did this by using nested for loops to take each transaction and stack the items purchased in order using the transactions index as an id. Then after removing all the n/a entities created by transactions with less than four items, we were able to shift the data into a dataframe.

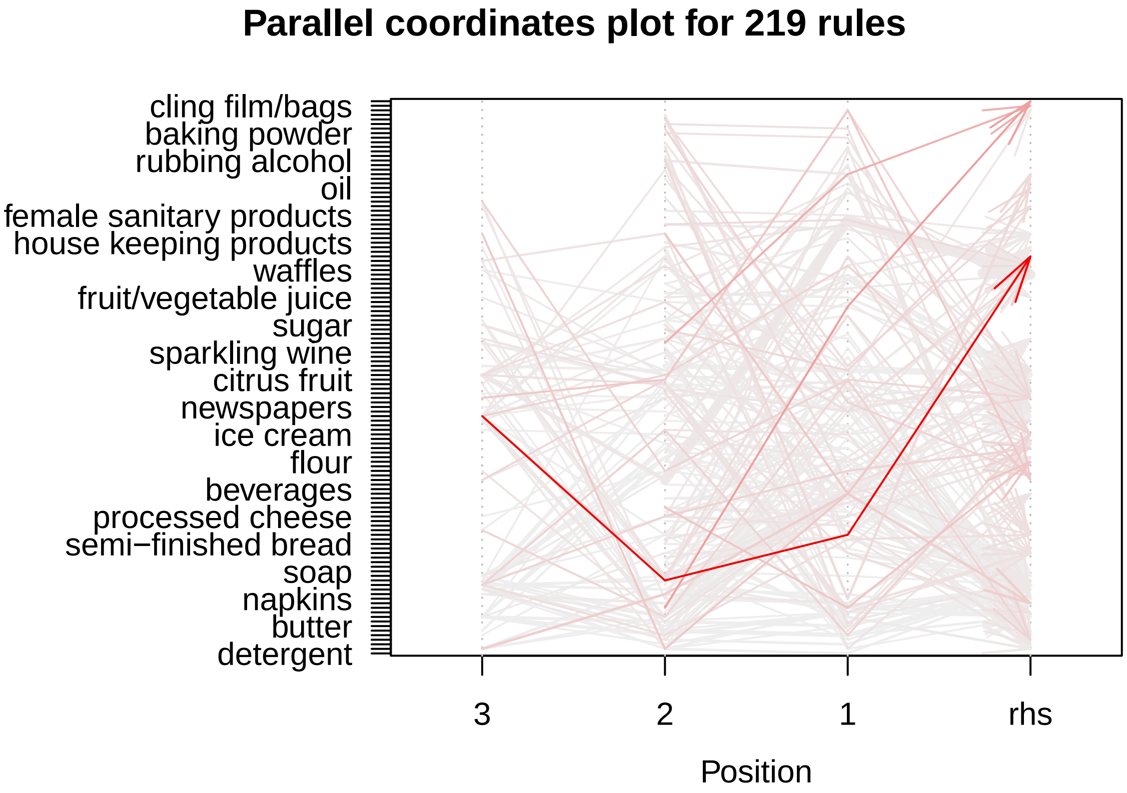
For apriori, we chose 75% confidence as our bar and a support of 0.00125 (as described in notes in our code). This gave us 219 rules which was enough to work with at a max length of 4.

Plotting the rule sets from apriori using parallels shows the complex relationships formed from the rules, which looks good but isn't great at conveying the information. Plotting the rules in a "bubble map" selecting only the top 50 rules, like the playlists example, shows the relationships from grocery data in a much clearer manner.

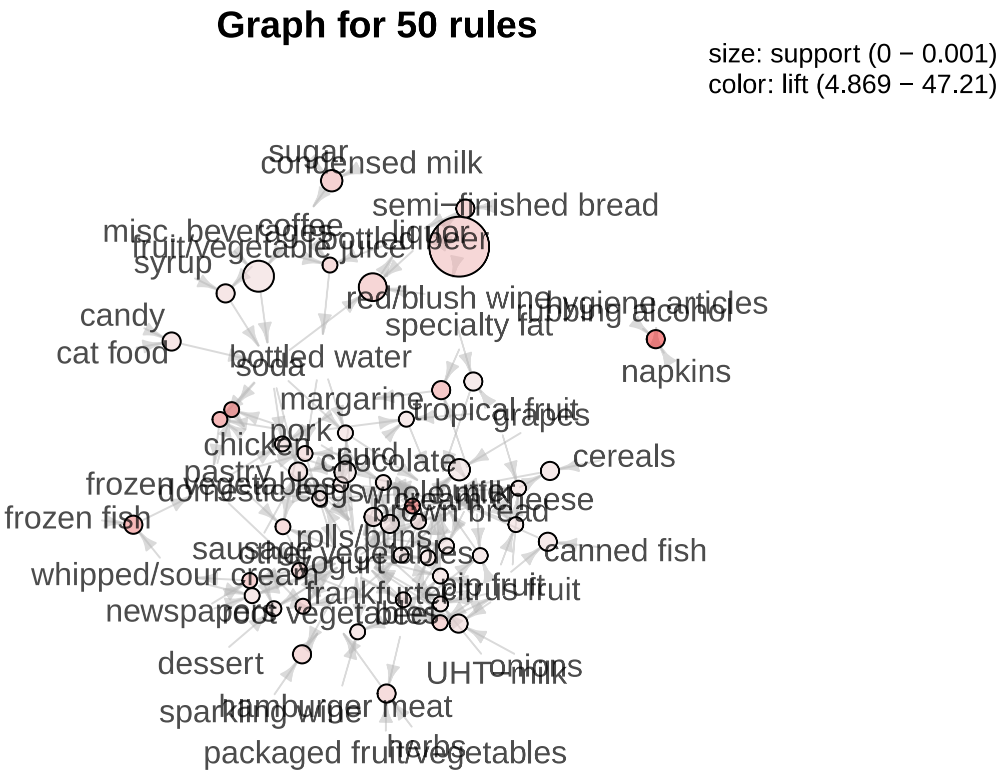
Finally analyzing the top 10 rules by both confidence and lift, we can see more pertinent correlations.

```
## transactions as itemMatrix in sparse format with
##  15296 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.01677625
##
## 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
## 3485 2630 2102 7079
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   2.835   4.000   4.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2            10
## 3           100
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime  support minlen
##        0.75    0.1    1 none FALSE            TRUE       5 0.000125      1
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [167 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [219 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```



```
## Warning: plot: Too many rules supplied. Only plotting the best 50 rules
## using 'support' (change control parameter max if needed)
```



```
##                                                       lhs
## 84                               {detergent,root vegetables}
## 104                                     {domestic eggs,meat}
## 106                                            {curd,onions}
## 160                          {butter,grapes,other vegetables}
## 195                         {chicken,curd,other vegetables}
## 79                             {frozen fish,root vegetables}
## 87                                          {herbs,napkins}
## 136 {packaged fruit/vegetables,tropical fruit,whole milk}
## 144                               {herbs,pip fruit,whole milk}
## 5                                          {rolls/buns,soap}
##                     rhs      support confidence     lift count
## 84         {whole milk} 0.0001961297       0.75 4.565062     3
## 104        {whole milk} 0.0001961297       0.75 4.565062     3
## 106        {whole milk} 0.0003922594       0.75 4.565062     6
## 160        {whole milk} 0.0002615063       0.80 4.869399     4
## 195        {whole milk} 0.0002615063       0.80 4.869399     4
## 79  {other vegetables} 0.0001961297       0.75 6.028376     3
## 87  {other vegetables} 0.0001961297       0.75 6.028376     3
## 136 {other vegetables} 0.0001961297       0.75 6.028376     3
## 144 {other vegetables} 0.0001961297       0.75 6.028376     3
## 5          {whole milk} 0.0001307531       1.00 6.086749     2
```

```
##                                      lhs                      rhs
## 2             {napkins,rubbing alcohol}      {hygiene articles}
## 3                  {decalcifier,waffles} {fruit/vegetable juice}
## 4  {bottled water,specialty vegetables}                  {soda}
## 5                     {rolls/buns,soap}            {whole milk}
## 6                   {ready soups,yogurt}            {rolls/buns}
## 7                 {coffee,pudding powder}                  {curd}
## 9                {meat,organic sausage}            {whole milk}
## 10            {berries,organic sausage}               {sausage}
## 11               {frozen fish,tidbits}            {whole milk}
## 12                    {coffee,tidbits}            {rolls/buns}
##         support confidence      lift count
## 2  0.0001961297          1 47.209877     3
## 3  0.0001307531          1 21.513361     2
## 4  0.0001307531          1  8.918950     2
## 5  0.0001307531          1  6.086749     2
## 6  0.0001307531          1  8.455500     2
## 7  0.0001307531          1 29.190840     2
## 9  0.0001307531          1  6.086749     2
## 10 0.0001307531          1 16.554113     2
## 11 0.0001307531          1  6.086749     2
## 12 0.0001307531          1  8.455500     2
```