Association Rule Mining

2024-08-18

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
##
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
```

```
##
##     union
```

```
library(arules)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Attaching package: 'arules'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
```

```
grocery_raw = read.csv("/Users/krummelha/Downloads/groceries.txt", header = FALSE)

str(grocery_raw)
```

```
## 'data.frame':    15296 obs. of  4 variables:
##  $ V1: chr  "citrus fruit" "tropical fruit" "whole milk" "pip fruit" ...
##  $ V2: chr  "semi-finished bread" "yogurt" "" "yogurt" ...
##  $ V3: chr  "margarine" "coffee" "" "cream cheese " ...
##  $ V4: chr  "ready soups" "" "" "meat spreads" ...
```

```
summary(grocery_raw)
```

```
##       V1                 V2                 V3                 V4
##  Length:15296       Length:15296       Length:15296       Length:15296
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
transactions_list <- split(grocery_raw, seq(nrow(grocery_raw)))
transactions_list <- lapply(transactions_list, function(x) x[x != ""])


Groctrans <- as(transactions_list, "transactions")

summary(Groctrans)
```

```
## 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             2
## 3             3
```

```
# rules with support > .005 & confidence >.1 & length (# items in cart) <= 3
GroceryRule = apriori(Groctrans,
                      parameter=list(support=.005, confidence=.1, maxlen=3))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5   0.005      1
##  maxlen target  ext
##       3  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 76
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [101 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(Groctrans, parameter = list(support = 0.005, confidence =
## 0.1, : Mining stopped (maxlen reached). Only patterns up to a length of 3
## returned!
```

```
##  done [0.00s].
## writing ... [118 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(GroceryRule)
```

```
##       lhs                          rhs                  support
## [1]   {}                        => {soda}               0.112120816
## [2]   {}                        => {rolls/buns}         0.118266213
## [3]   {}                        => {other vegetables}   0.124411611
## [4]   {}                        => {whole milk}         0.164291318
## [5]   {butter milk}             => {whole milk}         0.005033996
## [6]   {onions}                  => {root vegetables}    0.005295502
## [7]   {onions}                  => {other vegetables}   0.007452929
## [8]   {onions}                  => {whole milk}         0.005360879
## [9]   {berries}                 => {other vegetables}   0.005164749
## [10]  {berries}                 => {whole milk}         0.005230126
## [11]  {hamburger meat}          => {other vegetables}   0.006210774
## [12]  {hamburger meat}          => {whole milk}         0.005818515
## [13]  {dessert}                 => {whole milk}         0.006603033
## [14]  {cream cheese }           => {yogurt}             0.005033996
## [15]  {chocolate}               => {soda}               0.005360879
## [16]  {chicken}                 => {other vegetables}   0.007975941
## [17]  {chicken}                 => {whole milk}         0.006341527
## [18]  {frozen vegetables}       => {whole milk}         0.005687762
## [19]  {canned beer}             => {soda}               0.006537657
## [20]  {beef}                    => {citrus fruit}       0.005099372
## [21]  {beef}                    => {root vegetables}    0.008695084
## [22]  {root vegetables}         => {beef}               0.008695084
## [23]  {beef}                    => {other vegetables}   0.008302824
## [24]  {beef}                    => {whole milk}         0.008172071
## [25]  {curd}                    => {yogurt}             0.007649059
## [26]  {curd}                    => {other vegetables}   0.007060669
## [27]  {curd}                    => {whole milk}         0.012617678
## [28]  {margarine}               => {rolls/buns}         0.005491632
## [29]  {butter}                  => {yogurt}             0.006210774
## [30]  {butter}                  => {other vegetables}   0.008237448
## [31]  {butter}                  => {whole milk}         0.014382845
## [32]  {pork}                    => {root vegetables}    0.006733787
## [33]  {pork}                    => {other vegetables}   0.009283473
## [34]  {pork}                    => {whole milk}         0.008695084
## [35]  {frankfurter}             => {sausage}            0.006472280
## [36]  {sausage}                 => {frankfurter}        0.006472280
## [37]  {frankfurter}             => {tropical fruit}     0.005557008
## [38]  {frankfurter}             => {rolls/buns}         0.006210774
## [39]  {frankfurter}             => {other vegetables}   0.007060669
## [40]  {frankfurter}             => {whole milk}         0.008237448
## [41]  {bottled beer}            => {bottled water}      0.006929916
## [42]  {bottled beer}            => {soda}               0.008302824
```

```
## [43]  {brown bread}                => {pastry}             0.005033996
## [44]  {brown bread}                => {rolls/buns}         0.006276151
## [45]  {brown bread}                => {whole milk}         0.006733787
## [46]  {domestic eggs}              => {rolls/buns}         0.008172071
## [47]  {domestic eggs}              => {whole milk}         0.008172071
## [48]  {fruit/vegetable juice}      => {bottled water}      0.005753138
## [49]  {fruit/vegetable juice}      => {soda}               0.009348849
## [50]  {shopping bags}              => {soda}               0.006406904
## [51]  {whipped/sour cream}         => {yogurt}             0.009741109
## [52]  {yogurt}                     => {whipped/sour cream} 0.009741109
## [53]  {whipped/sour cream}         => {rolls/buns}         0.005360879
## [54]  {whipped/sour cream}         => {other vegetables}   0.008302824
## [55]  {whipped/sour cream}         => {whole milk}         0.011440900
## [56]  {pip fruit}                  => {citrus fruit}       0.008172071
## [57]  {citrus fruit}               => {pip fruit}          0.008172071
## [58]  {pip fruit}                  => {sausage}            0.006210774
## [59]  {sausage}                    => {pip fruit}          0.006210774
## [60]  {pip fruit}                  => {tropical fruit}     0.012683054
## [61]  {tropical fruit}             => {pip fruit}          0.012683054
## [62]  {pip fruit}                  => {root vegetables}    0.008106695
## [63]  {root vegetables}            => {pip fruit}          0.008106695
## [64]  {pip fruit}                  => {other vegetables}   0.010917887
## [65]  {pip fruit}                  => {whole milk}         0.012552301
## [66]  {pastry}                     => {soda}               0.007256799
## [67]  {pastry}                     => {rolls/buns}         0.010198745
## [68]  {pastry}                     => {whole milk}         0.009414226
## [69]  {citrus fruit}               => {sausage}            0.006929916
## [70]  {sausage}                    => {citrus fruit}       0.006929916
## [71]  {citrus fruit}               => {tropical fruit}     0.012486925
## [72]  {tropical fruit}             => {citrus fruit}       0.012486925
## [73]  {citrus fruit}               => {root vegetables}    0.008695084
## [74]  {root vegetables}            => {citrus fruit}       0.008695084
## [75]  {citrus fruit}               => {yogurt}             0.006733787
## [76]  {citrus fruit}               => {other vegetables}   0.012813808
## [77]  {other vegetables}           => {citrus fruit}       0.012813808
## [78]  {citrus fruit}               => {whole milk}         0.012813808
## [79]  {sausage}                    => {tropical fruit}     0.008172071
## [80]  {tropical fruit}             => {sausage}            0.008172071
## [81]  {sausage}                    => {root vegetables}    0.007322176
## [82]  {root vegetables}            => {sausage}            0.007322176
## [83]  {sausage}                    => {rolls/buns}         0.010787134
## [84]  {sausage}                    => {other vegetables}   0.012617678
## [85]  {other vegetables}           => {sausage}            0.012617678
## [86]  {sausage}                    => {whole milk}         0.012552301
## [87]  {bottled water}              => {soda}               0.014644351
## [88]  {soda}                       => {bottled water}      0.014644351
## [89]  {bottled water}              => {rolls/buns}         0.008564331
## [90]  {tropical fruit}             => {root vegetables}    0.010983264
## [91]  {root vegetables}            => {tropical fruit}     0.010983264
## [92]  {tropical fruit}             => {yogurt}             0.008172071
## [93]  {tropical fruit}             => {other vegetables}   0.015494247
## [94]  {other vegetables}           => {tropical fruit}     0.015494247
## [95]  {tropical fruit}             => {whole milk}         0.018305439
## [96]  {whole milk}                 => {tropical fruit}     0.018305439
```

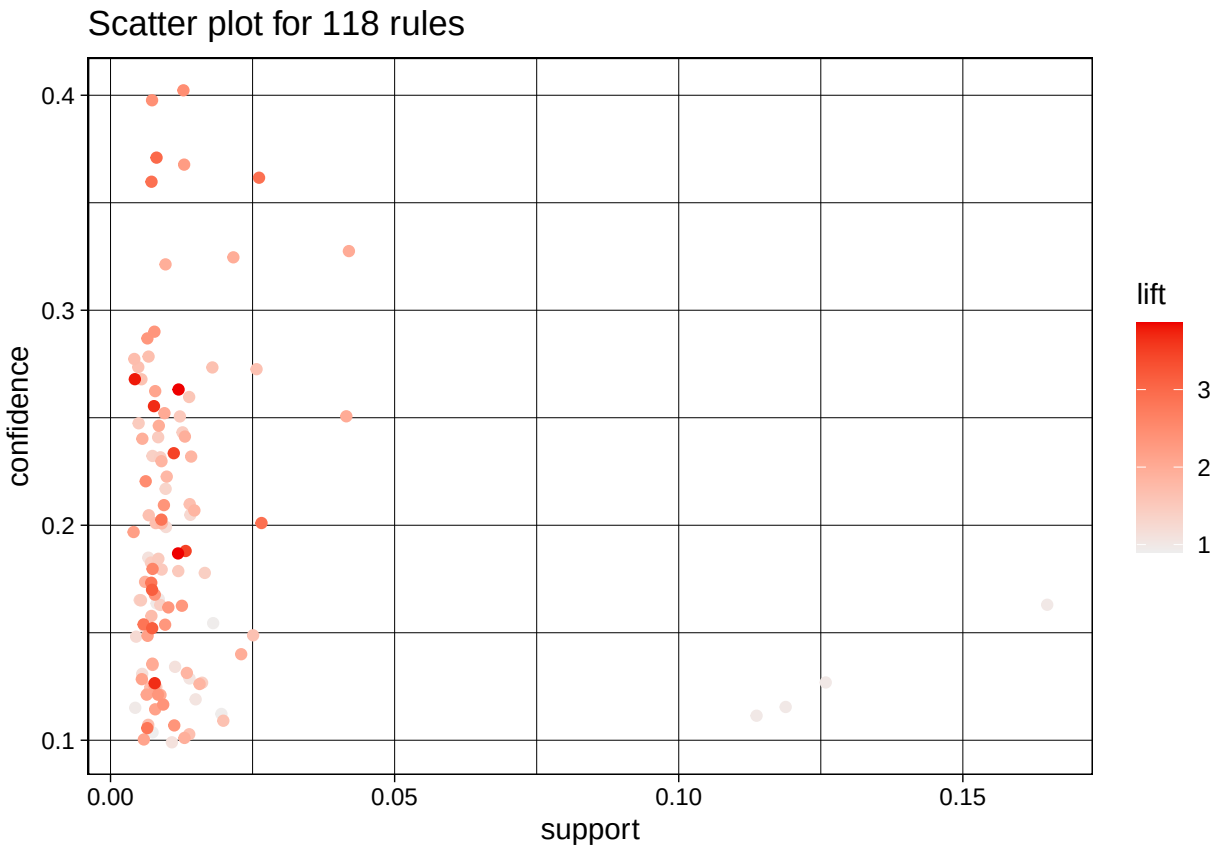
```
## [97]  {root vegetables}                    => {other vegetables}   0.025366109
## [98]  {other vegetables}                   => {root vegetables}    0.025366109
## [99]  {root vegetables}                    => {whole milk}         0.022620293
## [100] {whole milk}                         => {root vegetables}    0.022620293
## [101] {yogurt}                             => {rolls/buns}         0.011898536
## [102] {rolls/buns}                         => {yogurt}             0.011898536
## [103] {yogurt}                             => {other vegetables}   0.015886506
## [104] {other vegetables}                   => {yogurt}             0.015886506
## [105] {yogurt}                             => {whole milk}         0.024254707
## [106] {whole milk}                         => {yogurt}             0.024254707
## [107] {soda}                               => {rolls/buns}         0.014252092
## [108] {rolls/buns}                         => {soda}               0.014252092
## [109] {rolls/buns}                         => {whole milk}         0.018305439
## [110] {whole milk}                         => {rolls/buns}         0.018305439
## [111] {other vegetables}                   => {whole milk}         0.040860356
## [112] {whole milk}                         => {other vegetables}   0.040860356
## [113] {other vegetables, root vegetables} => {whole milk}         0.008172071
## [114] {root vegetables, whole milk}        => {other vegetables}   0.008172071
## [115] {other vegetables, whole milk}       => {root vegetables}    0.008172071
## [116] {other vegetables, yogurt}           => {whole milk}         0.006341527
## [117] {whole milk, yogurt}                 => {other vegetables}   0.006341527
## [118] {other vegetables, whole milk}       => {yogurt}             0.006341527
##       confidence coverage   lift      count
## [1]   0.1121208  1.00000000 1.0000000 1715
## [2]   0.1182662  1.00000000 1.0000000 1809
## [3]   0.1244116  1.00000000 1.0000000 1903
## [4]   0.1642913  1.00000000 1.0000000 2513
## [5]   0.2800000  0.01797856 1.7042897   77
## [6]   0.2655738  0.01993985 3.7893810   81
## [7]   0.3737705  0.01993985 3.0043055  114
## [8]   0.2688525  0.01993985 1.6364374   82
## [9]   0.2415902  0.02137814 1.9418623   79
## [10]  0.2446483  0.02137814 1.4891129   80
## [11]  0.2905199  0.02137814 2.3351508   95
## [12]  0.2721713  0.02137814 1.6566381   89
## [13]  0.2767123  0.02386245 1.6842785  101
## [14]  0.1974359  0.02549686 2.2011512   77
## [15]  0.1680328  0.03190377 1.4986761   82
## [16]  0.2890995  0.02758891 2.3237343  122
## [17]  0.2298578  0.02758891 1.3990868   97
## [18]  0.1839323  0.03092312 1.1195500   87
## [19]  0.1308901  0.04994770 1.1674019  100
## [20]  0.1511628  0.03373431 2.8405234   78
## [21]  0.2577519  0.03373431 3.6777739  133
## [22]  0.1240672  0.07008368 3.6777739  133
## [23]  0.2461240  0.03373431 1.9783044  127
## [24]  0.2422481  0.03373431 1.4745031  125
## [25]  0.2232824  0.03425732 2.4893063  117
## [26]  0.2061069  0.03425732 1.6566530  108
## [27]  0.3683206  0.03425732 2.2418751  193
## [28]  0.1458333  0.03765690 1.2330938   84
## [29]  0.1743119  0.03563023 1.9433493   95
## [30]  0.2311927  0.03563023 1.8582885  126
## [31]  0.4036697  0.03563023 2.4570363  220
```

```
## [32]  0.1816578  0.03706851 2.5920135  103
## [33]  0.2504409  0.03706851 2.0130028  142
## [34]  0.2345679  0.03706851 1.4277559  133
## [35]  0.1706897  0.03791841 2.8256158   99
## [36]  0.1071429  0.06040795 2.8256158   99
## [37]  0.1465517  0.03791841 2.1721465   85
## [38]  0.1637931  0.03791841 1.3849526   95
## [39]  0.1862069  0.03791841 1.4967003  108
## [40]  0.2172414  0.03791841 1.3222937  126
## [41]  0.1338384  0.05177824 1.8833412  106
## [42]  0.1603535  0.05177824 1.4301852  127
## [43]  0.1206897  0.04171025 2.1097931   77
## [44]  0.1504702  0.04171025 1.2723010   96
## [45]  0.1614420  0.04171025 0.9826570  103
## [46]  0.2003205  0.04079498 1.6938102  125
## [47]  0.2003205  0.04079498 1.2193007  125
## [48]  0.1237693  0.04648274 1.7416521   88
## [49]  0.2011252  0.04648274 1.7938255  143
## [50]  0.1011352  0.06334990 0.9020198   98
## [51]  0.2113475  0.04609048 2.3562475  149
## [52]  0.1086006  0.08969665 2.3562475  149
## [53]  0.1163121  0.04609048 0.9834766   82
## [54]  0.1801418  0.04609048 1.4479504  127
## [55]  0.2482270  0.04609048 1.5108951  175
## [56]  0.1680108  0.04864017 3.1571161  125
## [57]  0.1535627  0.05321653 3.1571161  125
## [58]  0.1276882  0.04864017 2.1137644   95
## [59]  0.1028139  0.06040795 2.1137644   95
## [60]  0.2607527  0.04864017 3.8647995  194
## [61]  0.1879845  0.06746862 3.8647995  194
## [62]  0.1666667  0.04864017 2.3781095  124
## [63]  0.1156716  0.07008368 2.3781095  124
## [64]  0.2244624  0.04864017 1.8041915  167
## [65]  0.2580645  0.04864017 1.5707739  192
## [66]  0.1268571  0.05720450 1.1314326  111
## [67]  0.1782857  0.05720450 1.5074949  156
## [68]  0.1645714  0.05720450 1.0017050  144
## [69]  0.1302211  0.05321653 2.1556952  106
## [70]  0.1147186  0.06040795 2.1556952  106
## [71]  0.2346437  0.05321653 3.4778203  191
## [72]  0.1850775  0.06746862 3.4778203  191
## [73]  0.1633907  0.05321653 2.3313653  133
## [74]  0.1240672  0.07008368 2.3313653  133
## [75]  0.1265356  0.05321653 1.4107062  103
## [76]  0.2407862  0.05321653 1.9354001  196
## [77]  0.1029953  0.12441161 1.9354001  196
## [78]  0.2407862  0.05321653 1.4656054  196
## [79]  0.1352814  0.06040795 2.0051008  125
## [80]  0.1211240  0.06746862 2.0051008  125
## [81]  0.1212121  0.06040795 1.7295341  112
## [82]  0.1044776  0.07008368 1.7295341  112
## [83]  0.1785714  0.06040795 1.5099108  165
## [84]  0.2088745  0.06040795 1.6788984  193
## [85]  0.1014188  0.12441161 1.6788984  193
```

```
## [86]  0.2077922  0.06040795 1.2647790  192
## [87]  0.2060718  0.07106433 1.8379438  224
## [88]  0.1306122  0.11212082 1.8379438  224
## [89]  0.1205152  0.07106433 1.0190161  131
## [90]  0.1627907  0.06746862 2.3228046  168
## [91]  0.1567164  0.07008368 2.3228046  168
## [92]  0.1211240  0.06746862 1.3503740  125
## [93]  0.2296512  0.06746862 1.8458982  237
## [94]  0.1245402  0.12441161 1.8458982  237
## [95]  0.2713178  0.06746862 1.6514435  280
## [96]  0.1114206  0.16429132 1.6514435  280
## [97]  0.3619403  0.07008368 2.9092164  388
## [98]  0.2038886  0.12441161 2.9092164  388
## [99]  0.3227612  0.07008368 1.9645663  346
## [100] 0.1376840  0.16429132 1.9645663  346
## [101] 0.1326531  0.08969665 1.1216480  182
## [102] 0.1006081  0.11826621 1.1216480  182
## [103] 0.1771137  0.08969665 1.4236107  243
## [104] 0.1276931  0.12441161 1.4236107  243
## [105] 0.2704082  0.08969665 1.6459066  371
## [106] 0.1476323  0.16429132 1.6459066  371
## [107] 0.1271137  0.11212082 1.0748099  218
## [108] 0.1205086  0.11826621 1.0748099  218
## [109] 0.1547816  0.11826621 0.9421170  280
## [110] 0.1114206  0.16429132 0.9421170  280
## [111] 0.3284288  0.12441161 1.9990636  625
## [112] 0.2487067  0.16429132 1.9990636  625
## [113] 0.3221649  0.02536611 1.9609371  125
## [114] 0.3612717  0.02262029 2.9038421  125
## [115] 0.2000000  0.04086036 2.8537313  125
## [116] 0.3991770  0.01588651 2.4296899   97
## [117] 0.2614555  0.02425471 2.1015364   97
## [118] 0.1552000  0.04086036 1.7302764   97
```

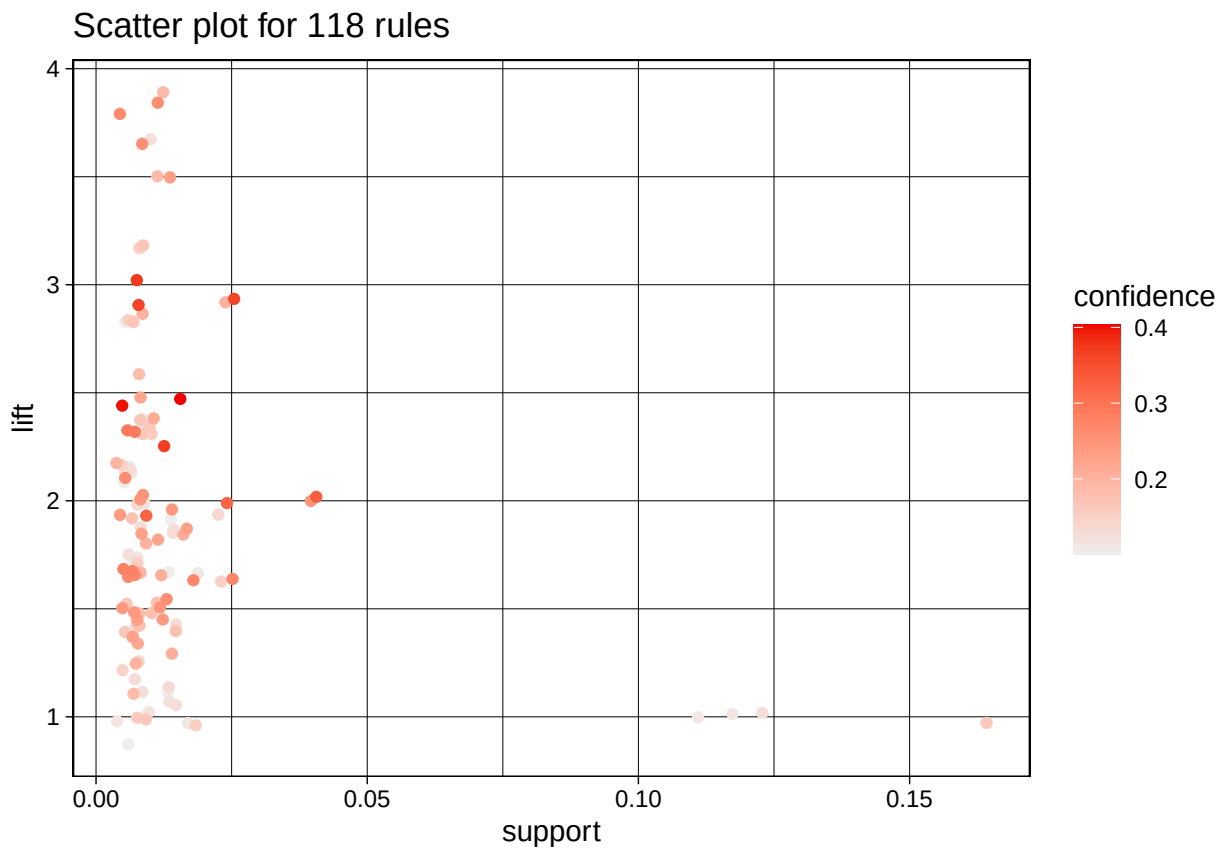
```
plot(GroceryRule)
```

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



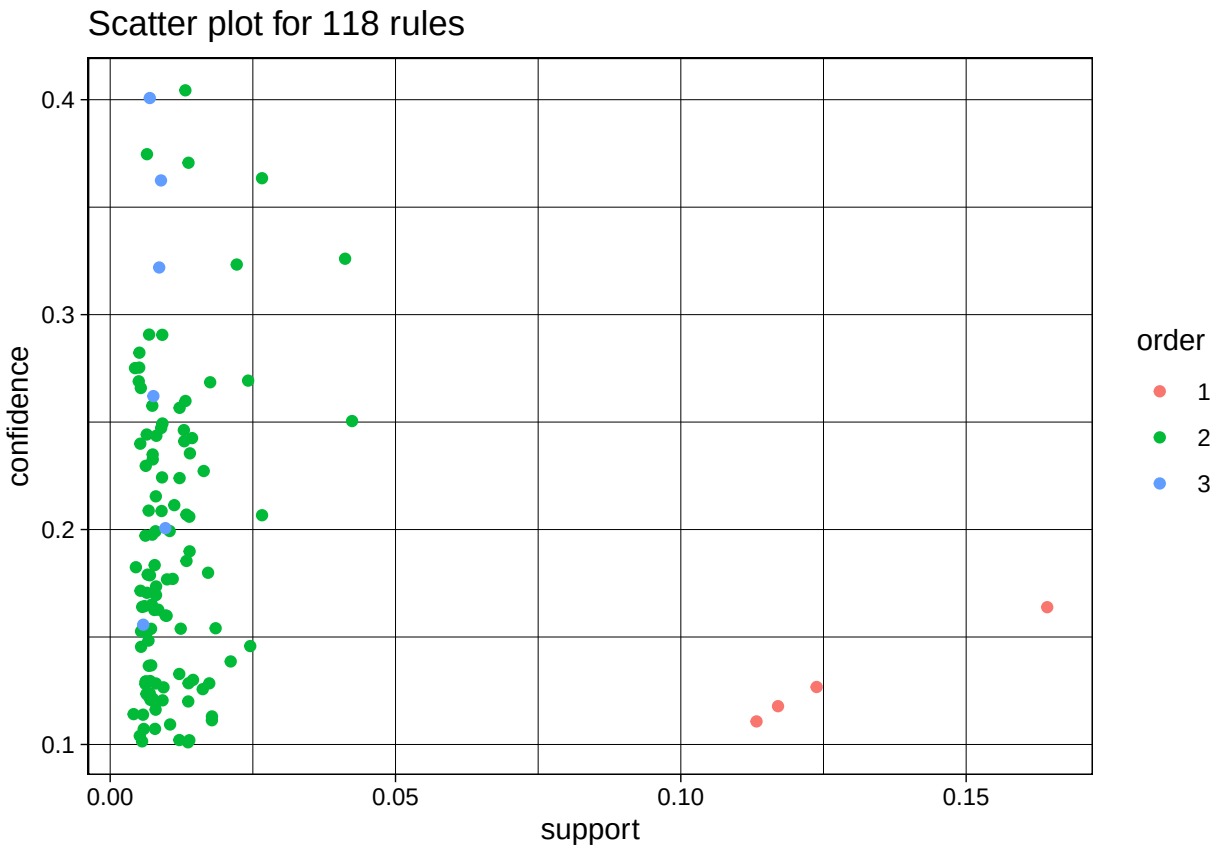

```
plot(GroceryRule, measure = c("support", "lift"), shading = "confidence")
```

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



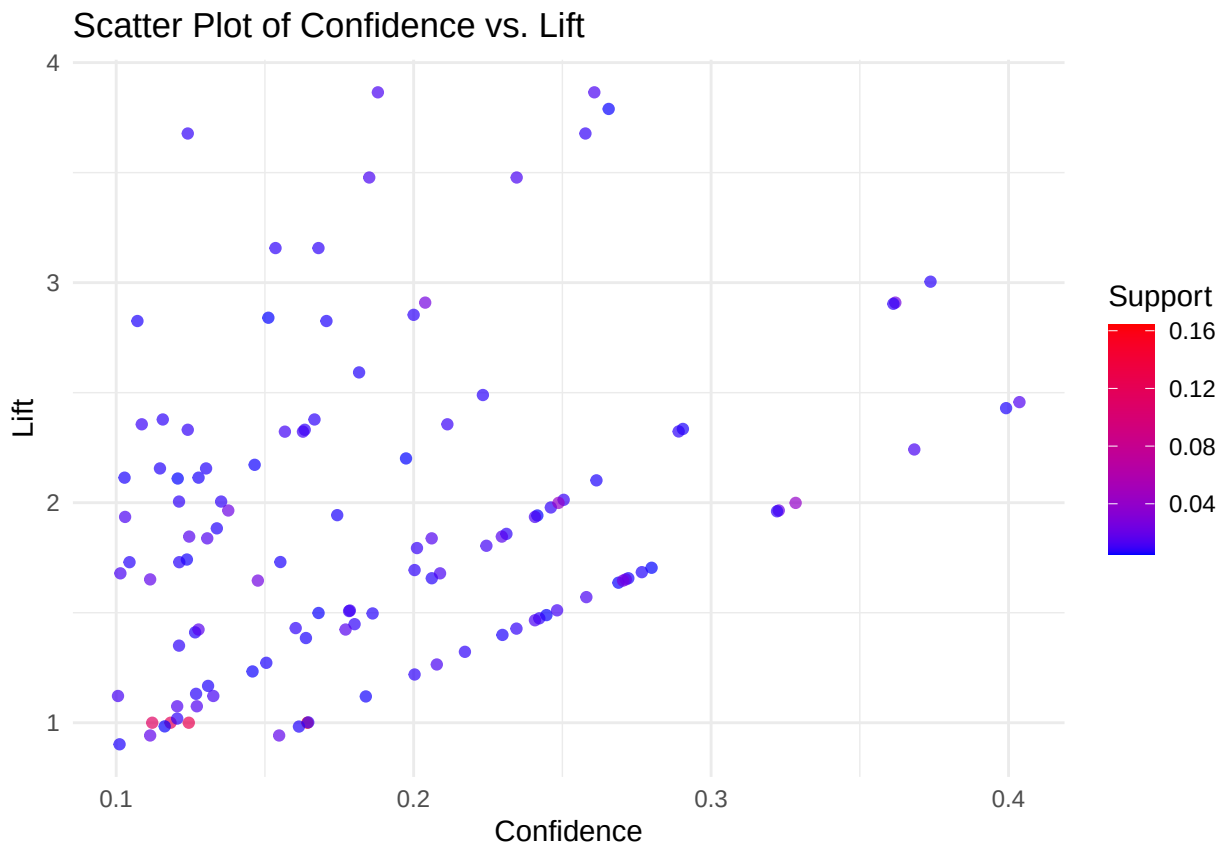

```
plot(GroceryRule, method='two-key plot')
```

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




```r
# Visualize rule metrics
metrics <- data.frame(
  confidence = quality(GroceryRule)$confidence,
  lift = quality(GroceryRule)$lift,
  support = quality(GroceryRule)$support
)
```

```r
# confidence vs. lift with color by support
ggplot(metrics, aes(x = confidence, y = lift, color = support)) +
  geom_point(alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Scatter Plot of Confidence vs. Lift",
       x = "Confidence",
       y = "Lift",
       color = "Support") +
  theme_minimal()
```



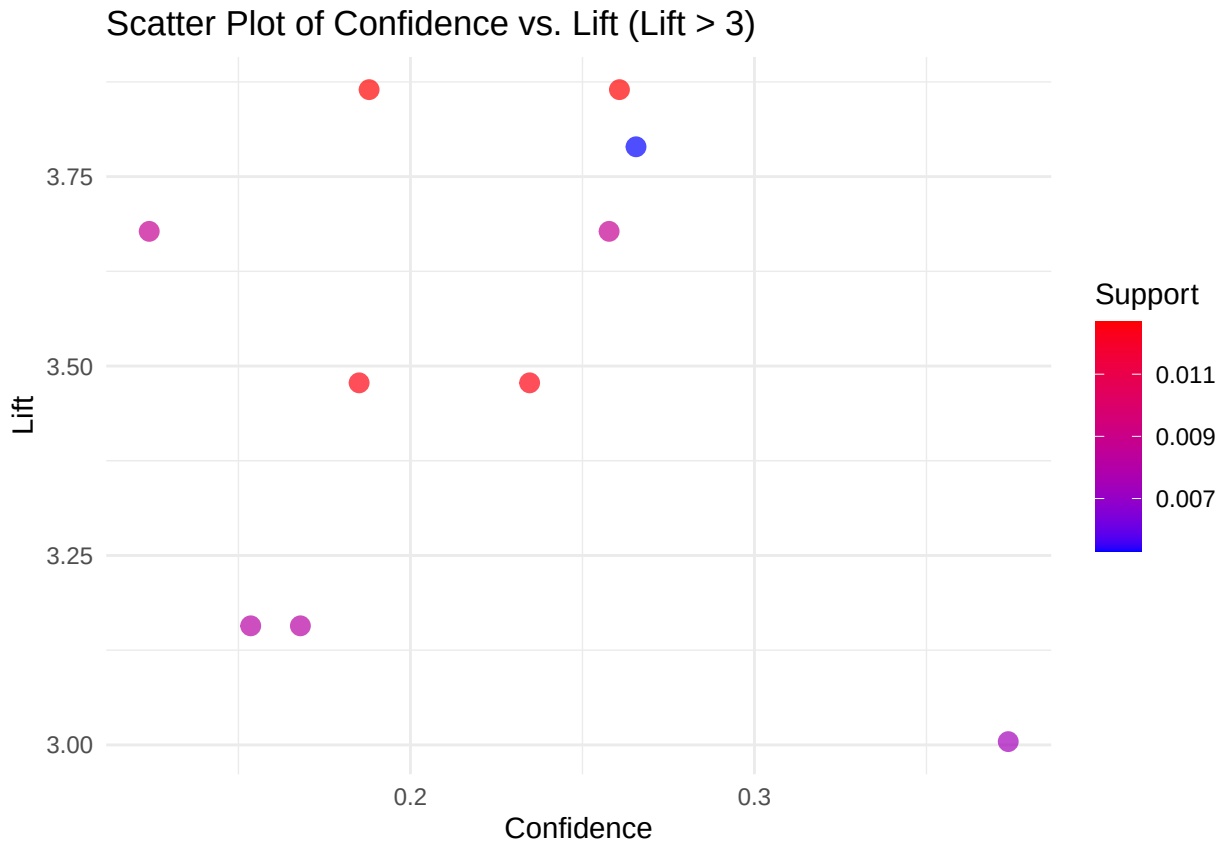

```
# Examine top lifts
inspect(subset(GroceryRule, lift>3))
```

```
##      lhs                   rhs                support     confidence coverage
## [1]  {onions}           => {root vegetables}  0.005295502 0.2655738  0.01993985
## [2]  {onions}           => {other vegetables} 0.007452929 0.3737705  0.01993985
## [3]  {beef}             => {root vegetables}  0.008695084 0.2577519  0.03373431
## [4]  {root vegetables}  => {beef}             0.008695084 0.1240672  0.07008368
## [5]  {pip fruit}        => {citrus fruit}     0.008172071 0.1680108  0.04864017
## [6]  {citrus fruit}     => {pip fruit}        0.008172071 0.1535627  0.05321653
## [7]  {pip fruit}        => {tropical fruit}   0.012683054 0.2607527  0.04864017
## [8]  {tropical fruit}   => {pip fruit}        0.012683054 0.1879845  0.06746862
## [9]  {citrus fruit}     => {tropical fruit}   0.012486925 0.2346437  0.05321653
## [10] {tropical fruit}   => {citrus fruit}     0.012486925 0.1850775  0.06746862
##      lift     count
## [1]  3.789381  81
## [2]  3.004306 114
## [3]  3.677774 133
## [4]  3.677774 133
## [5]  3.157116 125
## [6]  3.157116 125
## [7]  3.864800 194
## [8]  3.864800 194
## [9]  3.477820 191
## [10] 3.477820 191
```

```r
subset_rules <- subset(GroceryRule, lift > 3)

# Extract metrics from filtered rules
metrics_subset <- data.frame(
  confidence = quality(subset_rules)$confidence,
  lift = quality(subset_rules)$lift,
  support = quality(subset_rules)$support
)
```

```r
# Scatter plot of confidence vs. lift with color by support
ggplot(metrics_subset, aes(x = confidence, y = lift, color = support)) +
  geom_point(size = 3, alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Scatter Plot of Confidence vs. Lift (Lift > 3)",
       x = "Confidence",
       y = "Lift",
       color = "Support") +
  theme_minimal()
```



Although there are many rows, or shopping carts, in the grocery data, the maximum amount of groceries per single basket in the dataset is 4. It is hard to draw conclusions of associations from that since a lot of items have a low support.

Looking at the plot of lift by confidence, shaded by support, it seems the most confident rules happen at a lift between 2-3, not necessarily at the highest lift (greater than 3).

Some interesting things can be found in the high lift rules. Of those rules that had a lift higher than 3, most

of them were either vegetables or fruit. The combination of these are all complements. One that particularly sticks out is beef and root vegetables. These seem to also be complements of each other. People love stew! Yet, there is no obvious pattern of confidence/lift/support except that they have higher lifts than the rest.

We decided to isolate rules with a lift value greater than or equal to three to focus on those with the highest significance within the dataset. This removed rules which did not provide as much illuminating analysis in the relationships between items bought in this dataset. We chose a support threshold of 0.5%, to minimize the size of our dataset and only focus on the rules and relationships that were the most impactful and insightful. Moreover, we chose a confidence threshold of 10% to minimize our dataset even further and make sure that we understood the most clear lift relationships rather than focus on niche lift rules that may not have offered as much insight.