In [4]:

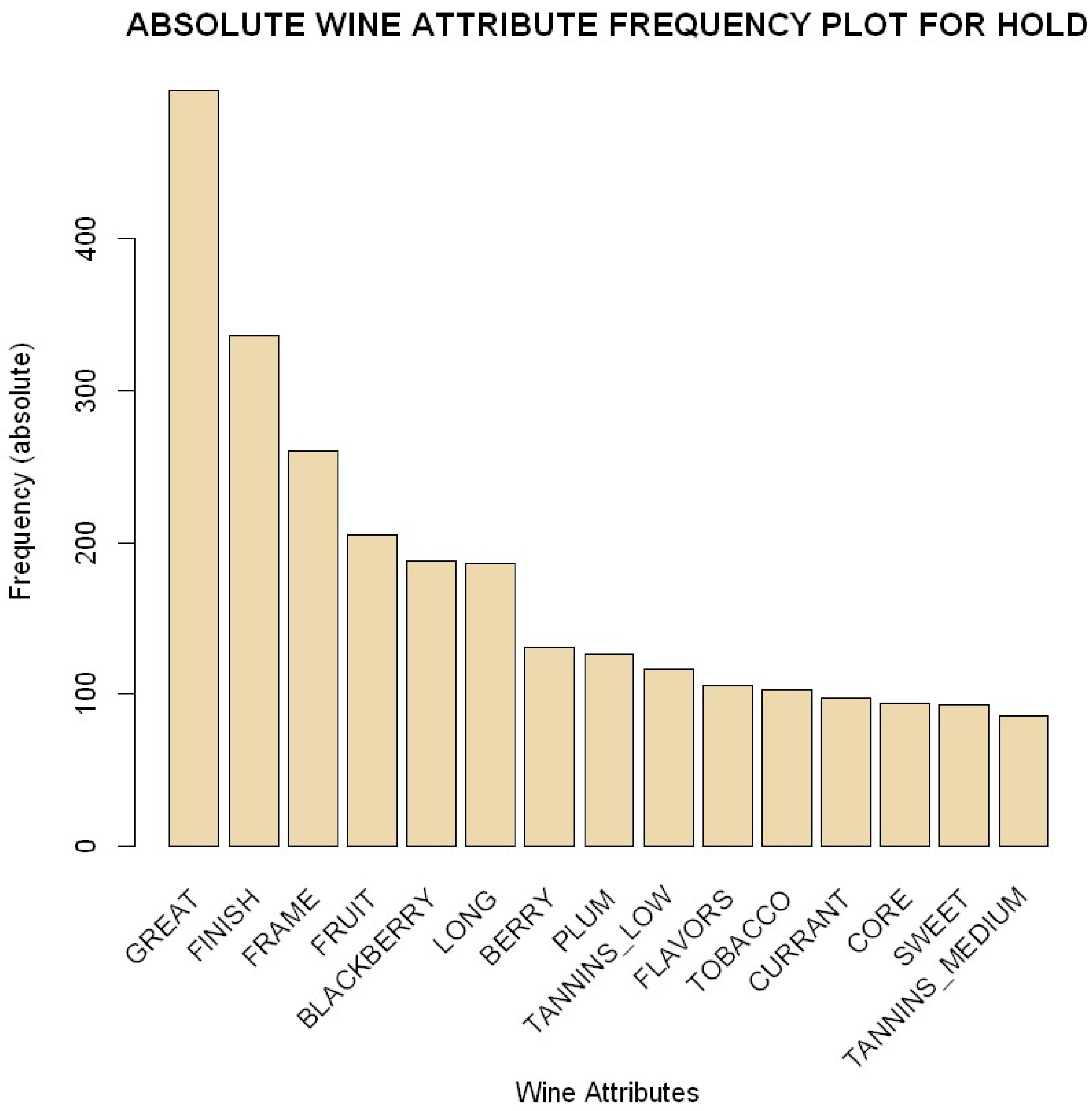
```
# Load libraries
library(tidyverse) # data manipulation
library(arules) # mining association rules and frequent itemsets
library(arulesViz) # visualization techniques for association rules
library(knitr) # dynamic report generation
library(gridExtra) # provides a number of user-level functions to work with "grid"
graphics
library(lubridate) # work with dates and times

# Read the data
trans <- read.transactions("./hold_wine_attributes_string.csv", format="basket")
head(trans)
```

```
Warning message in asMethod(object):
"removing duplicated items in transactions"
```

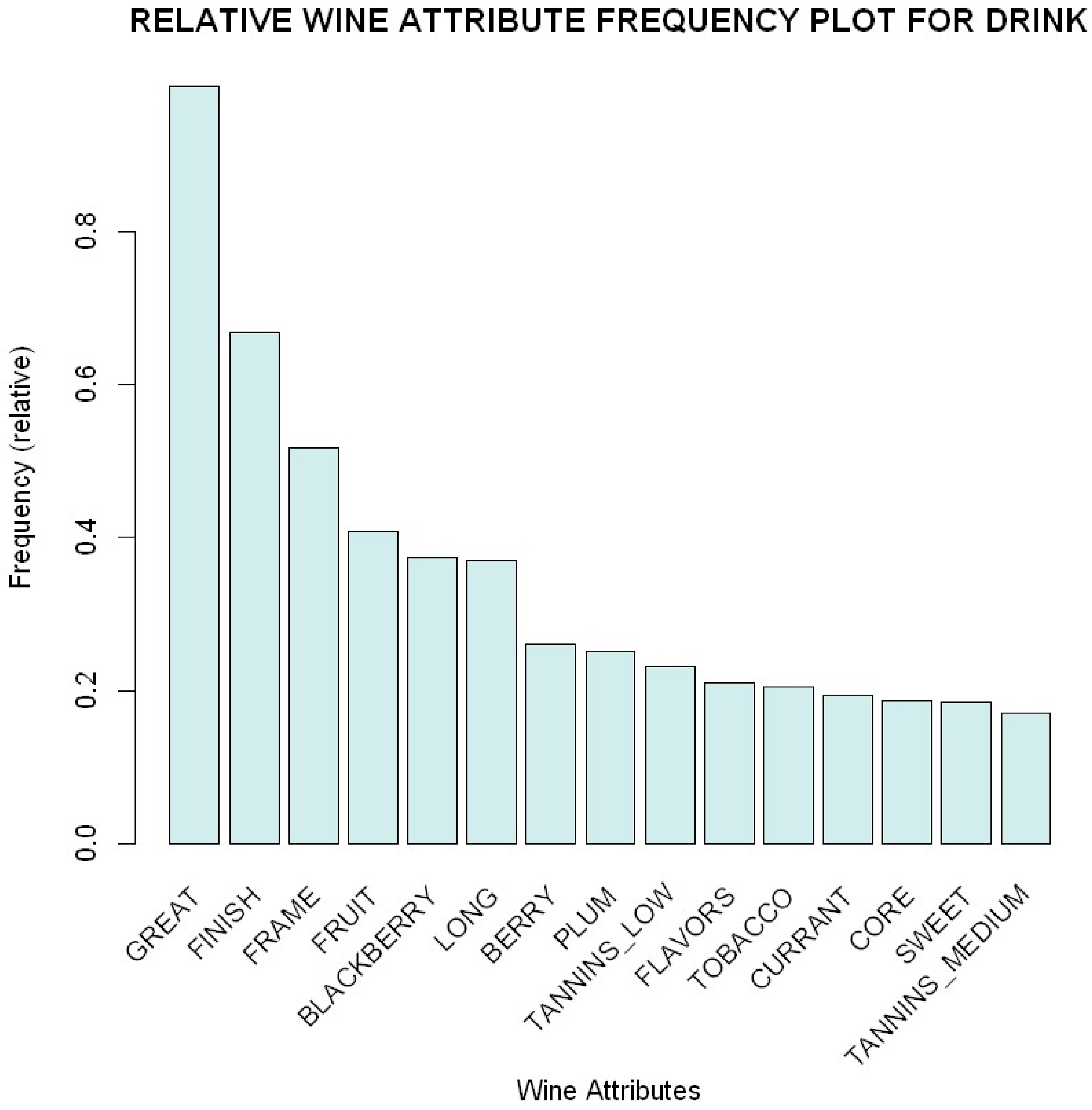
```
transactions in sparse format with
 6 transactions (rows) and
 897 items (columns)
```

In [5]:

```
# Absolute Item Frequency Plot
itemFrequencyPlot(trans, topN=15, type="absolute", col="wheat2",xlab="Wine Attribut
es",
                  ylab="Frequency (absolute)", main="ABSOLUTE WINE ATTRIBUTE FREQUE
NCY PLOT FOR HOLD")
```



```
In [6]: itemFrequencyPlot(trans, topN=15, type="relative", col="lightcyan2", xlab="Wine Att
        ributes",
                          ylab="Frequency (relative)", main="RELATIVE WINE ATTRIBUTE FREQUE
        NCY PLOT FOR HOLD")
```



In [7]:

```
# Support and confidence values
supportLevels <- c(0.5, 0.4, 0.2)
confidenceLevels <- c(0.8, 0.5, 0.3)

# Empty integers
rules_sup50 <- integer(length=9)
rules_sup40 <- integer(length=9)
rules_sup20 <- integer(length=9)

# Apriori algorithm with a support level of 50 and confidence of 80%

rules_sup50 <- length(apriori(trans, parameter=list(sup=supportLevels[1],
                                   conf=confidenceLevels[1], target="rules")))
rules_sup50

# Apriori algorithm with a support level of 40 and confidence of 50%

rules_sup40 <- length(apriori(trans, parameter=list(sup=supportLevels[2],
                                  conf=confidenceLevels[2], target="rules")))

rules_sup40

# Apriori algorithm with a support level of 20 and confidence of 30%

rules_sup20 <- length(apriori(trans, parameter=list(sup=supportLevels[3],
                                  conf=confidenceLevels[3], target="rules")))
rules_sup20
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5     0.5      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 251

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [3 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [3 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

3

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.5    0.1    1 none FALSE            TRUE       5     0.4      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 201

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [8 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

8

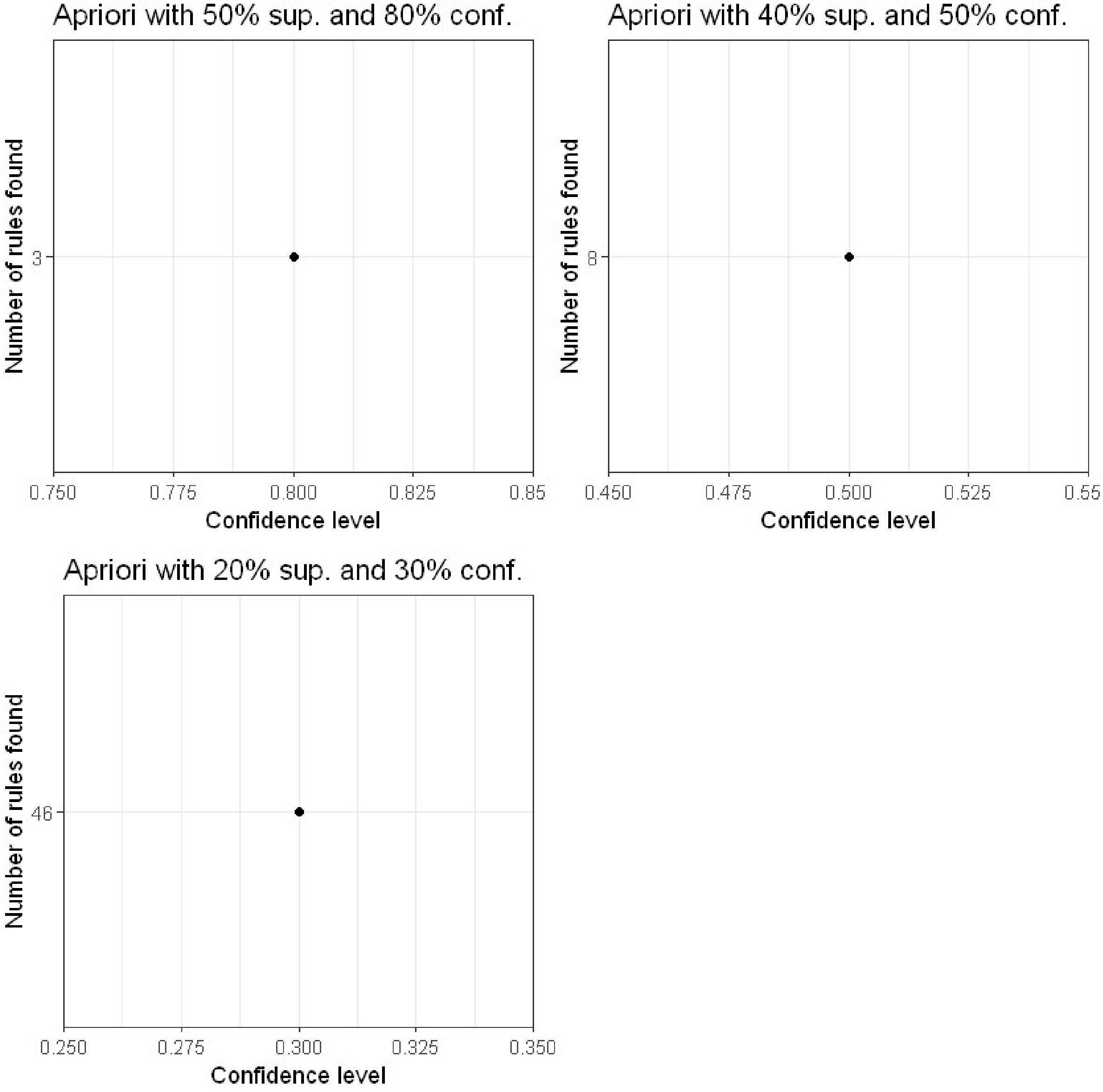
```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.3    0.1    1 none FALSE            TRUE       5     0.2      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 100

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [46 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

46

In [17]:

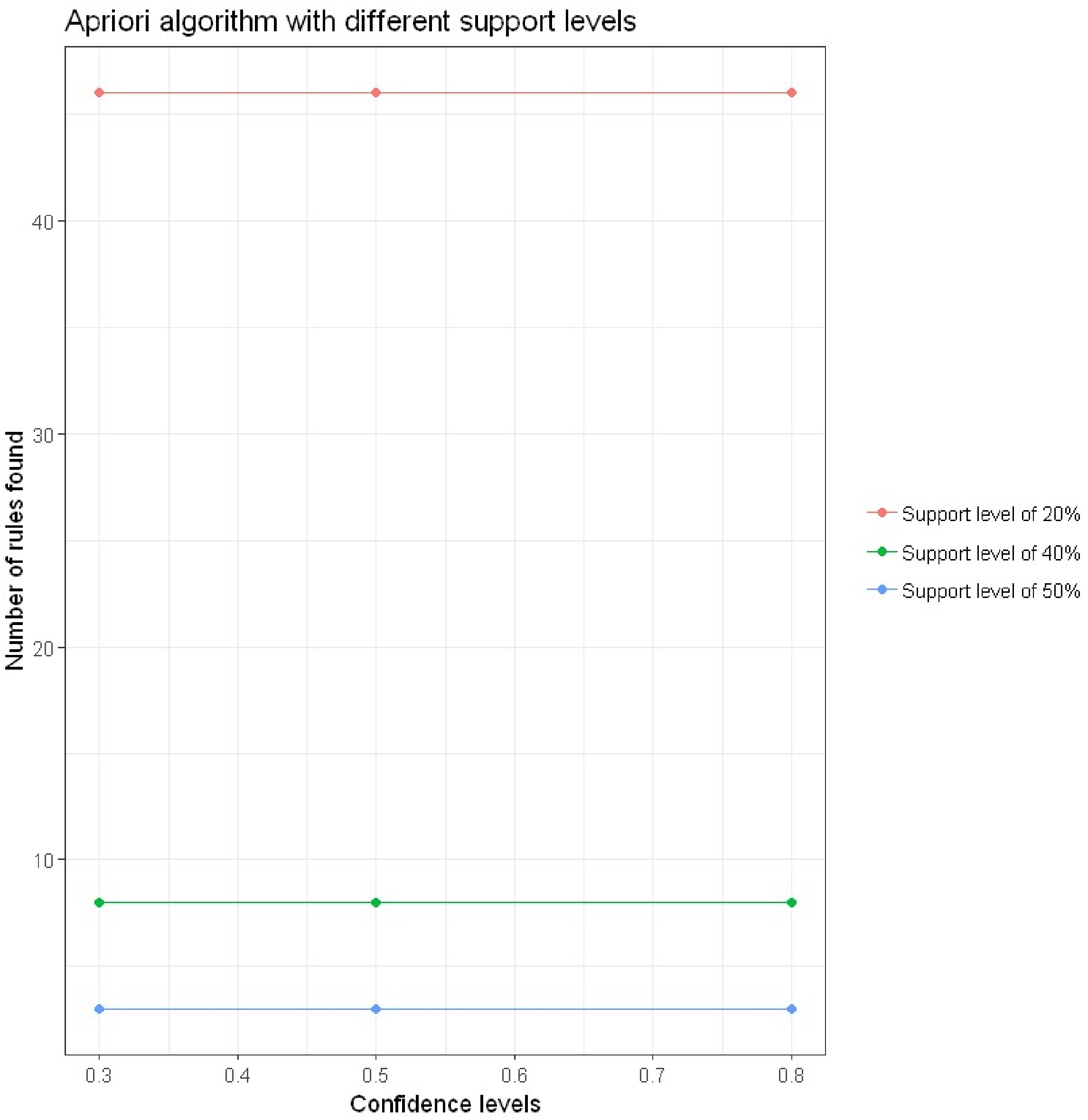
```
# Number of rules found with a support level of 70%
plot1 <- qplot(confidenceLevels[1], rules_sup50, geom=c("point", "line"),
               xlab="Confidence level", ylab="Number of rules found",
               main="Apriori with 50% sup. and 80% conf.") +   scale_y_continuous(breaks=seq(0, 10, 1))+
  theme_bw()

# Number of rules found with a support level of 40%
plot2 <- qplot(confidenceLevels[2], rules_sup40, geom=c("point", "line"),
               xlab="Confidence level", ylab="Number of rules found",
               main="Apriori with 40% sup. and 50% conf.") +
  scale_y_continuous(breaks=seq(0, 10, 2)) +
  theme_bw()

# Number of rules found with a support level of 20%
plot3 <- qplot(confidenceLevels[3], rules_sup20, geom=c("point", "line"),
               xlab="Confidence level", ylab="Number of rules found",
               main="Apriori with 20% sup. and 30% conf.") +
  scale_y_continuous(breaks=seq(10, 100, 2)) +
  theme_bw()


# Subplot
grid.arrange(plot1, plot2, plot3, ncol=2)
```

```
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?
```



In [18]:

```
# Data frame
num_rules <- data.frame(rules_sup50, rules_sup40, rules_sup20, confidenceLevels)

# Number of rules found with a support level of 10%, 5%, 1% and 0.5%
ggplot(data=num_rules, aes(x=confidenceLevels)) +

  # Plot line and points (support level of 70%)
  geom_line(aes(y=rules_sup50, colour="Support level of 50%")) +
  geom_point(aes(y=rules_sup50, colour="Support level of 50%")) +

  # Plot line and points (support level of 40%)
  geom_line(aes(y=rules_sup40, colour="Support level of 40%")) +
  geom_point(aes(y=rules_sup40, colour="Support level of 40%")) +

  # Plot line and points (support level of 20%)
  geom_line(aes(y=rules_sup20, colour="Support level of 20%")) +
  geom_point(aes(y=rules_sup20, colour="Support level of 20%")) +

  # Labs and theme
  labs(x="Confidence levels", y="Number of rules found",
       title="Apriori algorithm with different support levels") +
  theme_bw() +
  theme(legend.title=element_blank())
```


Apriori algorithm with different support levels
Number of rules found
40
30
20
10
0.3
0.4
0.5
0.6
0.7
0.8
Confidence levels
Support level of 20%
Support level of 40%
Support level of 50%

In [20]:

```
# Apriori algorithm execution with a support level of 7% and a confidence level of
80%
rules_sup1_conf80 <- apriori(trans, parameter=list(sup=0.5,
                             conf=0.8, target="rules"))
rules_sup1_conf80
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5     0.5      1
 maxlen target  ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 251

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [3 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [3 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 3 rules
```

In [21]:

```
# Apriori algorithm execution with a support level of 40% and a confidence level of
50%
rules_sup1_conf50 <- apriori(trans, parameter=list(sup=0.4,
                             conf=0.5, target="rules"))
rules_sup1_conf50
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.5    0.1    1 none FALSE            TRUE       5     0.4      1
 maxlen target  ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 201

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [8 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 8 rules
```

In [22]:

```
# Apriori algorithm execution with a support level of 40% and a confidence level of
50%
rules_sup1_conf30 <- apriori(trans, parameter=list(sup=0.2,
                              conf=0.3, target="rules"))
rules_sup1_conf30
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.3    0.1    1 none FALSE            TRUE       5     0.2      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 100

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[897 item(s), 503 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [46 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 46 rules
```

In [23]:

```
# Inspect association rules
inspect(rules_sup1_conf80)
```

```
    lhs         rhs     support   confidence lift      count
[1] {}       => {GREAT} 0.9900596 0.9900596  1.0000000 498
[2] {FRAME}  => {GREAT} 0.5149105 0.9961538  1.0061554 259
[3] {FINISH} => {GREAT} 0.6600398 0.9880952  0.9980159 332
```

In [24]:

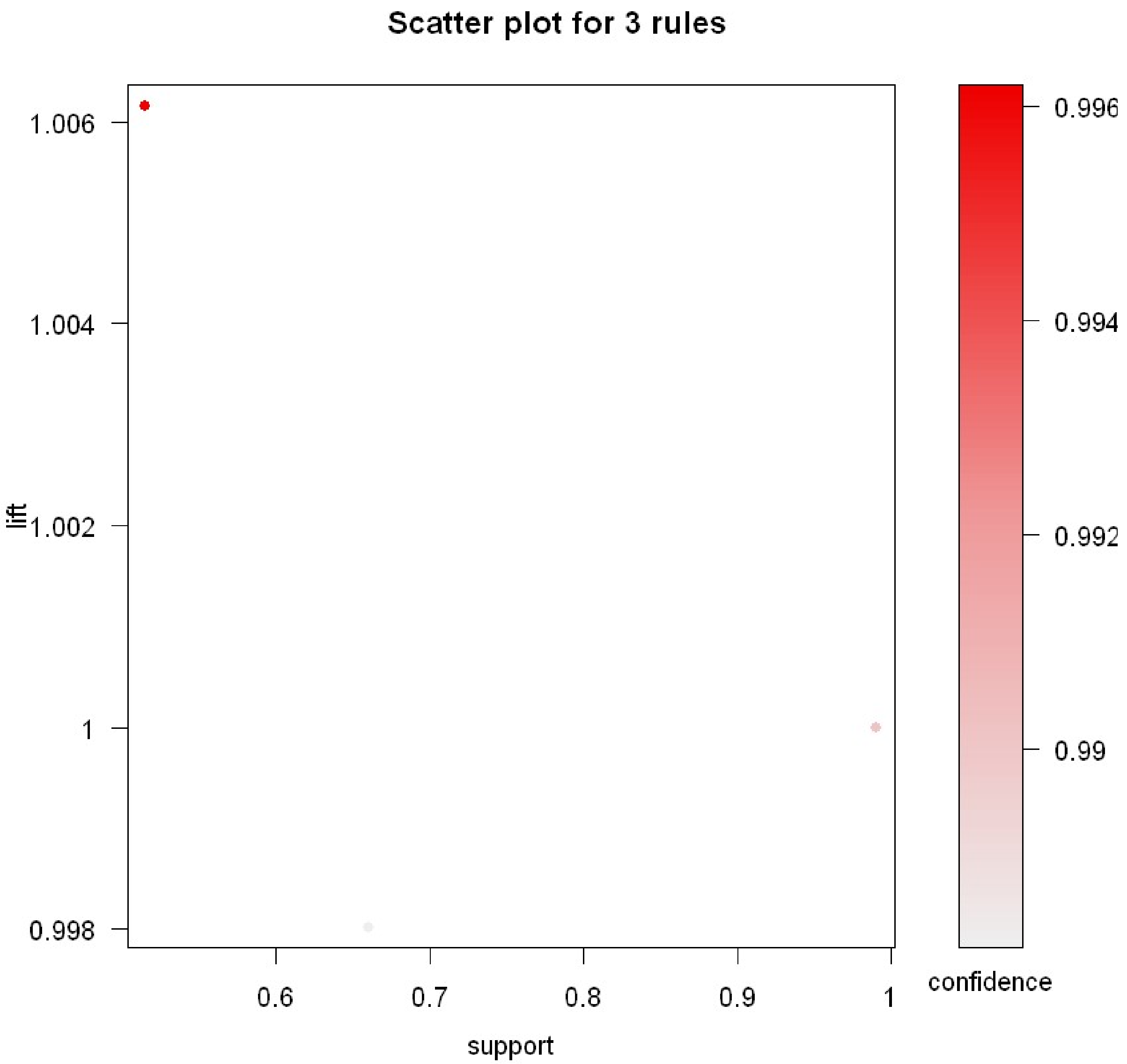
```
# Inspect association rules
inspect(rules_sup1_conf50)
```

```
    lhs         rhs      support   confidence lift      count
[1] {}       => {FRAME}  0.5168986 0.5168986  1.0000000 260
[2] {}       => {FINISH} 0.6679920 0.6679920  1.0000000 336
[3] {}       => {GREAT}  0.9900596 0.9900596  1.0000000 498
[4] {FRUIT}  => {GREAT}  0.4035785 0.9902439  1.0001861 203
[5] {FRAME}  => {GREAT}  0.5149105 0.9961538  1.0061554 259
[6] {GREAT}  => {FRAME}  0.5149105 0.5200803  1.0061554 259
[7] {FINISH} => {GREAT}  0.6600398 0.9880952  0.9980159 332
[8] {GREAT}  => {FINISH} 0.6600398 0.6666667  0.9980159 332
```

In [25]:
```
# Inspect association rules
inspect(rules_sup1_conf30)
```

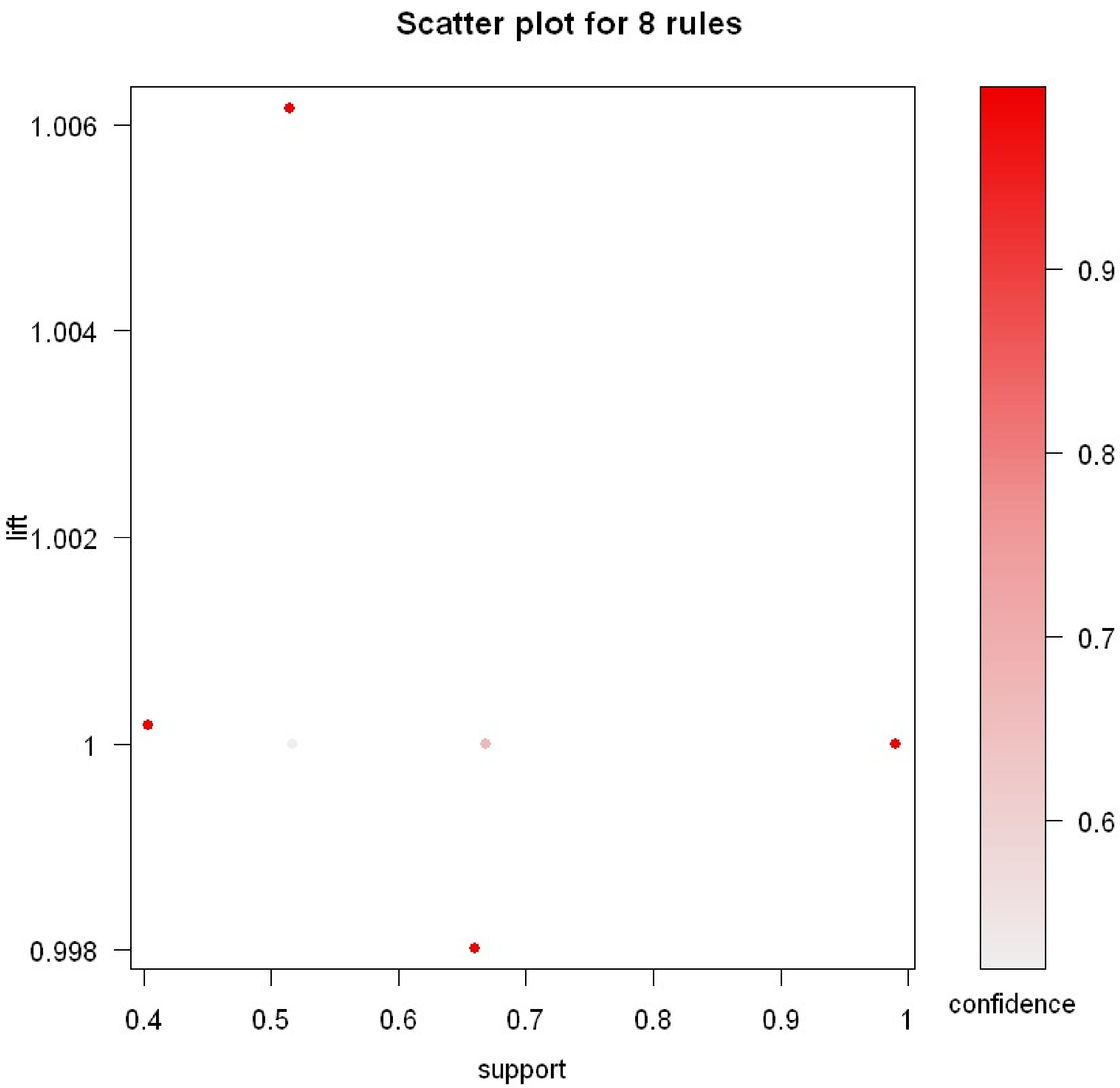
```
     lhs                        rhs          support   confidence lift      count
[1]  {}                      => {LONG}       0.3697813 0.3697813  1.0000000 186
[2]  {}                      => {BLACKBERRY} 0.3737575 0.3737575  1.0000000 188
[3]  {}                      => {FRUIT}      0.4075547 0.4075547  1.0000000 205
[4]  {}                      => {FRAME}      0.5168986 0.5168986  1.0000000 260
[5]  {}                      => {FINISH}     0.6679920 0.6679920  1.0000000 336
[6]  {}                      => {GREAT}      0.9900596 0.9900596  1.0000000 498
[7]  {TANNINS_LOW}           => {GREAT}      0.2306163 0.9914530  1.0014073 116
[8]  {TOBACCO}               => {GREAT}      0.2047714 1.0000000  1.0100402 103
[9]  {FLAVORS}               => {GREAT}      0.2067594 0.9811321  0.9909828 104
[10] {BERRY}                 => {GREAT}      0.2604374 1.0000000  1.0100402 131
[11] {PLUM}                  => {GREAT}      0.2504970 0.9921260  1.0020871 126
[12] {LONG}                  => {FRAME}      0.2544732 0.6881720  1.3313482 128
[13] {FRAME}                 => {LONG}       0.2544732 0.4923077  1.3313482 128
[14] {LONG}                  => {FINISH}     0.2584493 0.6989247  1.0463070 130
[15] {FINISH}                => {LONG}       0.2584493 0.3869048  1.0463070 130
[16] {LONG}                  => {GREAT}      0.3677932 0.9946237  1.0046098 185
[17] {GREAT}                 => {LONG}       0.3677932 0.3714859  1.0046098 185
[18] {BLACKBERRY}            => {FINISH}     0.2465209 0.6595745  0.9873987 124
[19] {FINISH}                => {BLACKBERRY} 0.2465209 0.3690476  0.9873987 124
[20] {BLACKBERRY}            => {GREAT}      0.3717694 0.9946809  1.0046676 187
[21] {GREAT}                 => {BLACKBERRY} 0.3717694 0.3755020  1.0046676 187
[22] {FRUIT}                 => {FINISH}     0.2564612 0.6292683  0.9420296 129
[23] {FINISH}                => {FRUIT}      0.2564612 0.3839286  0.9420296 129
[24] {FRUIT}                 => {GREAT}      0.4035785 0.9902439  1.0001861 203
[25] {GREAT}                 => {FRUIT}      0.4035785 0.4076305  1.0001861 203
[26] {FRAME}                 => {FINISH}     0.2962227 0.5730769  0.8579098 149
[27] {FINISH}                => {FRAME}      0.2962227 0.4434524  0.8579098 149
[28] {FRAME}                 => {GREAT}      0.5149105 0.9961538  1.0061554 259
[29] {GREAT}                 => {FRAME}      0.5149105 0.5200803  1.0061554 259
[30] {FINISH}                => {GREAT}      0.6600398 0.9880952  0.9980159 332
[31] {GREAT}                 => {FINISH}     0.6600398 0.6666667  0.9980159 332
[32] {FRAME,LONG}            => {GREAT}      0.2524851 0.9921875  1.0021492 127
[33] {GREAT,LONG}            => {FRAME}      0.2524851 0.6864865  1.3280873 127
[34] {FRAME,GREAT}           => {LONG}       0.2524851 0.4903475  1.3260472 127
[35] {FINISH,LONG}           => {GREAT}      0.2564612 0.9923077  1.0022706 129
[36] {GREAT,LONG}            => {FINISH}     0.2564612 0.6972973  1.0438707 129
[37] {FINISH,GREAT}          => {LONG}       0.2564612 0.3885542  1.0507676 129
[38] {BLACKBERRY,FINISH}     => {GREAT}      0.2445328 0.9919355  1.0018947 123
[39] {BLACKBERRY,GREAT}      => {FINISH}     0.2445328 0.6577540  0.9846734 123
[40] {FINISH,GREAT}          => {BLACKBERRY} 0.2445328 0.3704819  0.9912362 123
[41] {FINISH,FRUIT}          => {GREAT}      0.2524851 0.9844961  0.9943806 127
[42] {FRUIT,GREAT}           => {FINISH}     0.2524851 0.6256158  0.9365617 127
[43] {FINISH,GREAT}          => {FRUIT}      0.2524851 0.3825301  0.9385983 127
[44] {FINISH,FRAME}          => {GREAT}      0.2942346 0.9932886  1.0032614 148
[45] {FRAME,GREAT}           => {FINISH}     0.2942346 0.5714286  0.8554422 148
[46] {FINISH,GREAT}          => {FRAME}      0.2942346 0.4457831  0.8624189 148
```

```
In [26]: # Scatter plot
         plot(rules_sup1_conf80, measure=c("support","lift"), shading="confidence")
```



In [27]:

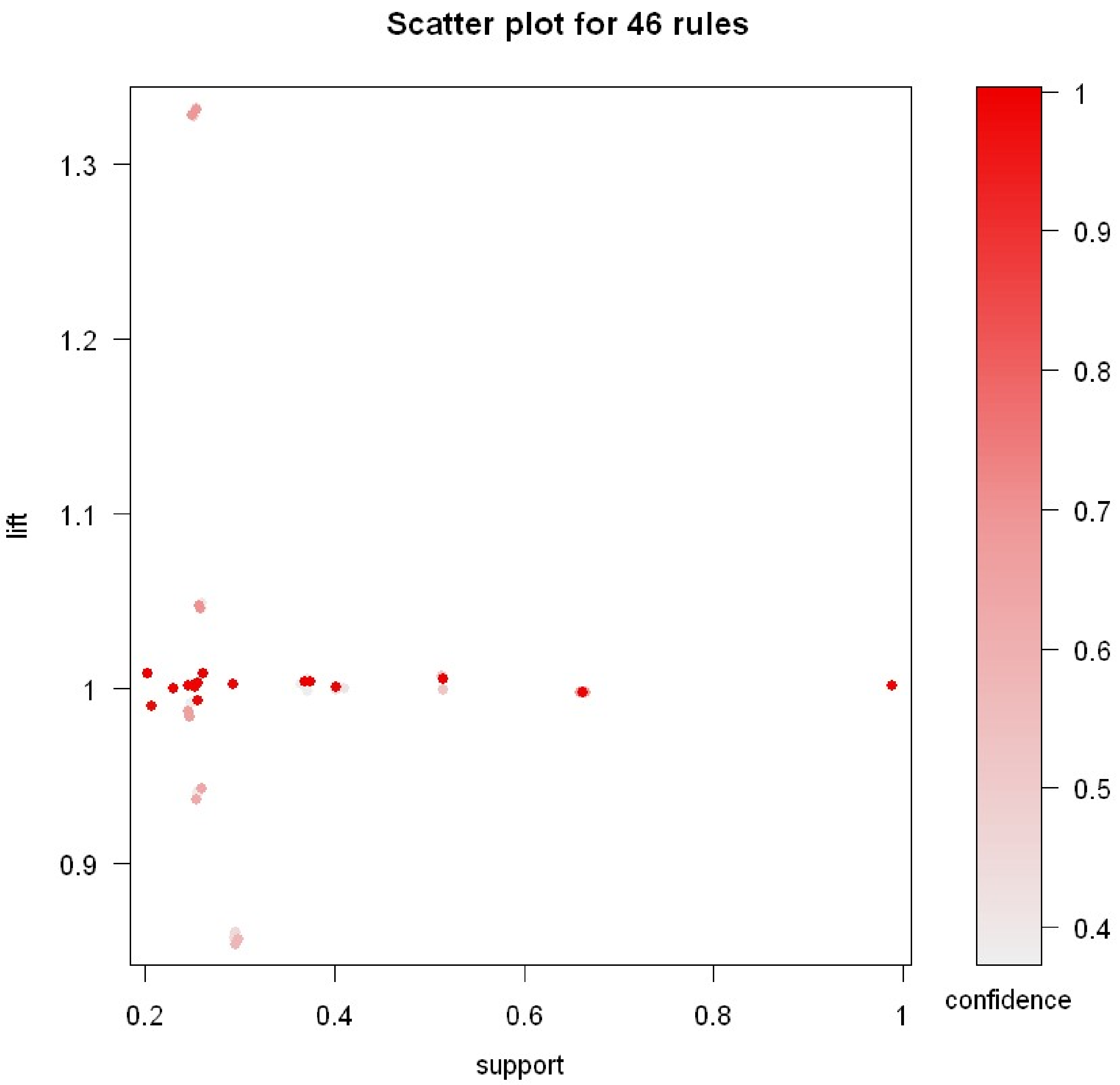
```
# Scatter plot
plot(rules_sup1_conf50, measure=c("support","lift"), shading="confidence")
```



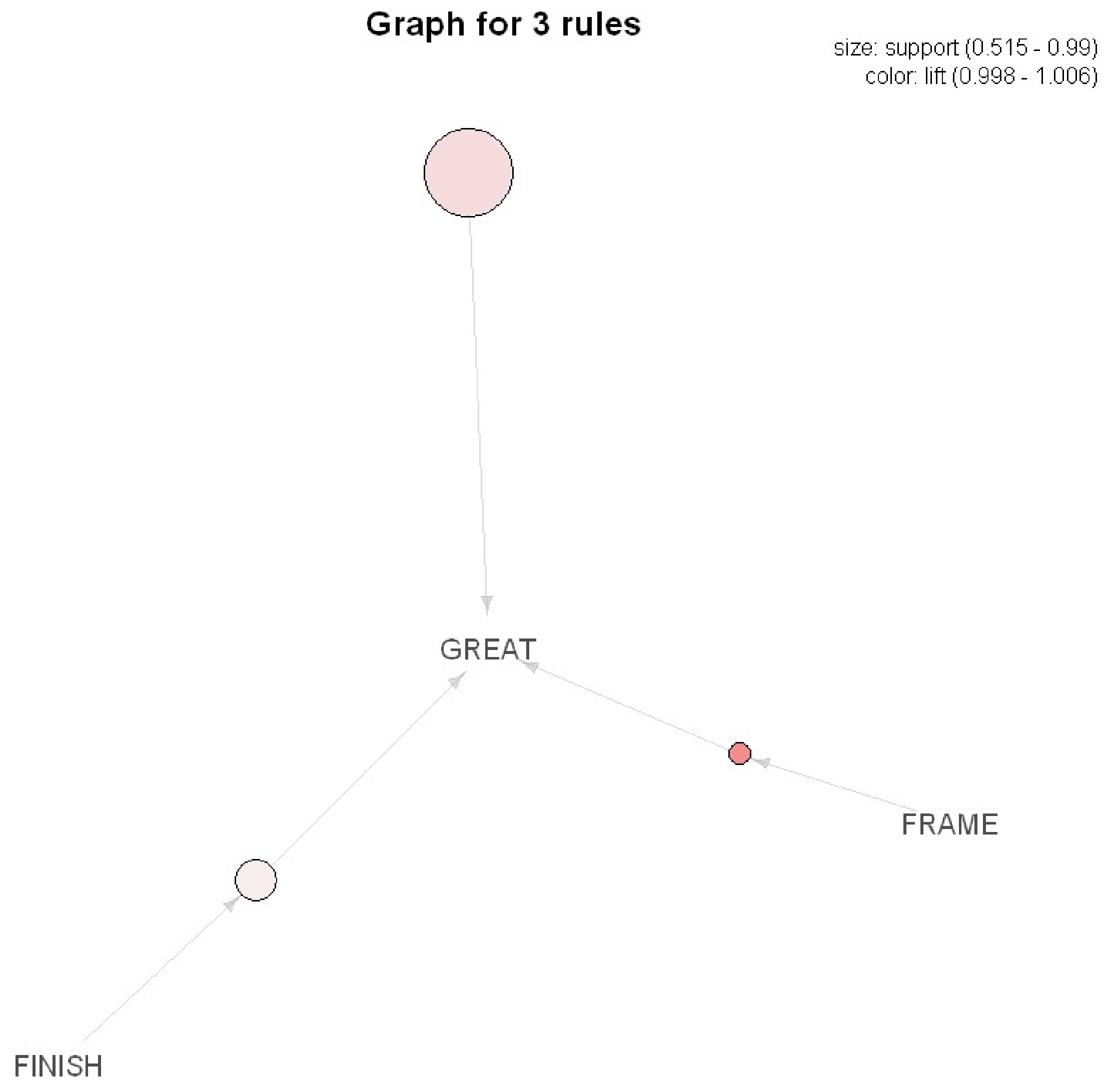
In [28]:

```
# Scatter plot
plot(rules_sup1_conf30, measure=c("support","lift"), shading="confidence")
```

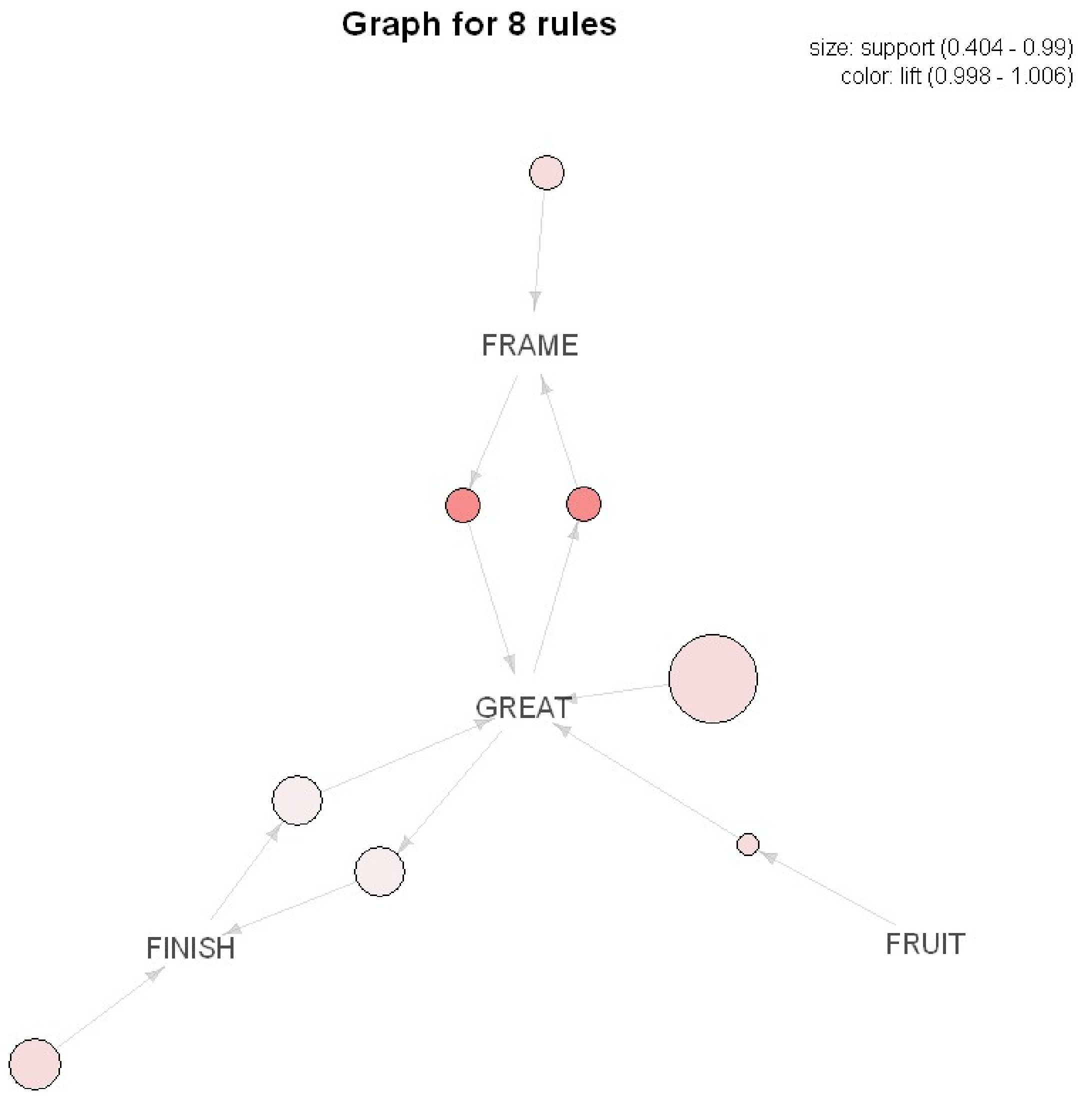
```
To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



```
In [29]: plot(rules_sup1_conf80, method="graph")
```

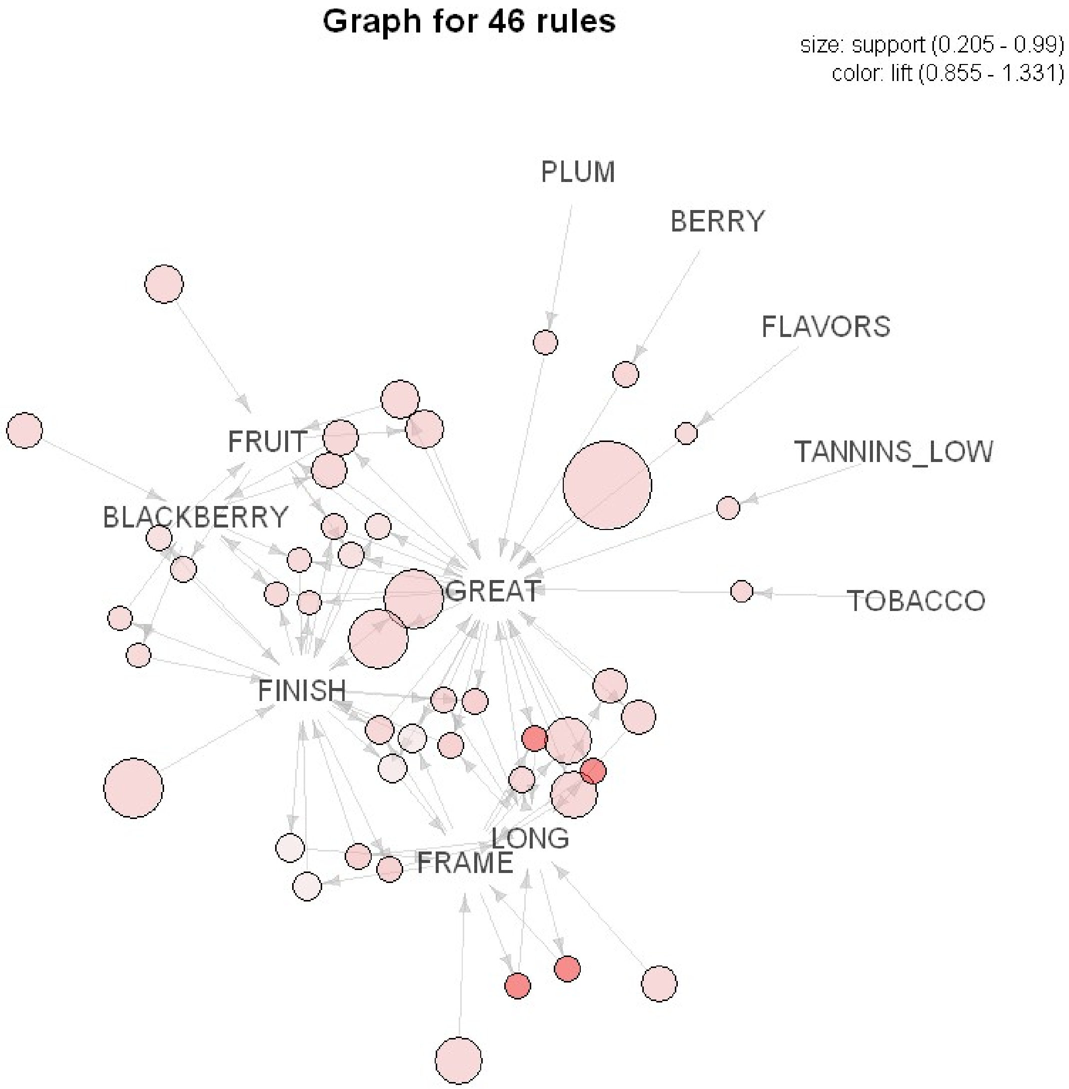


```
In [30]: plot(rules_sup1_conf50, method="graph")
```



In [31]:

```
plot(rules_sup1_conf30, method="graph")
```



In [ ]: