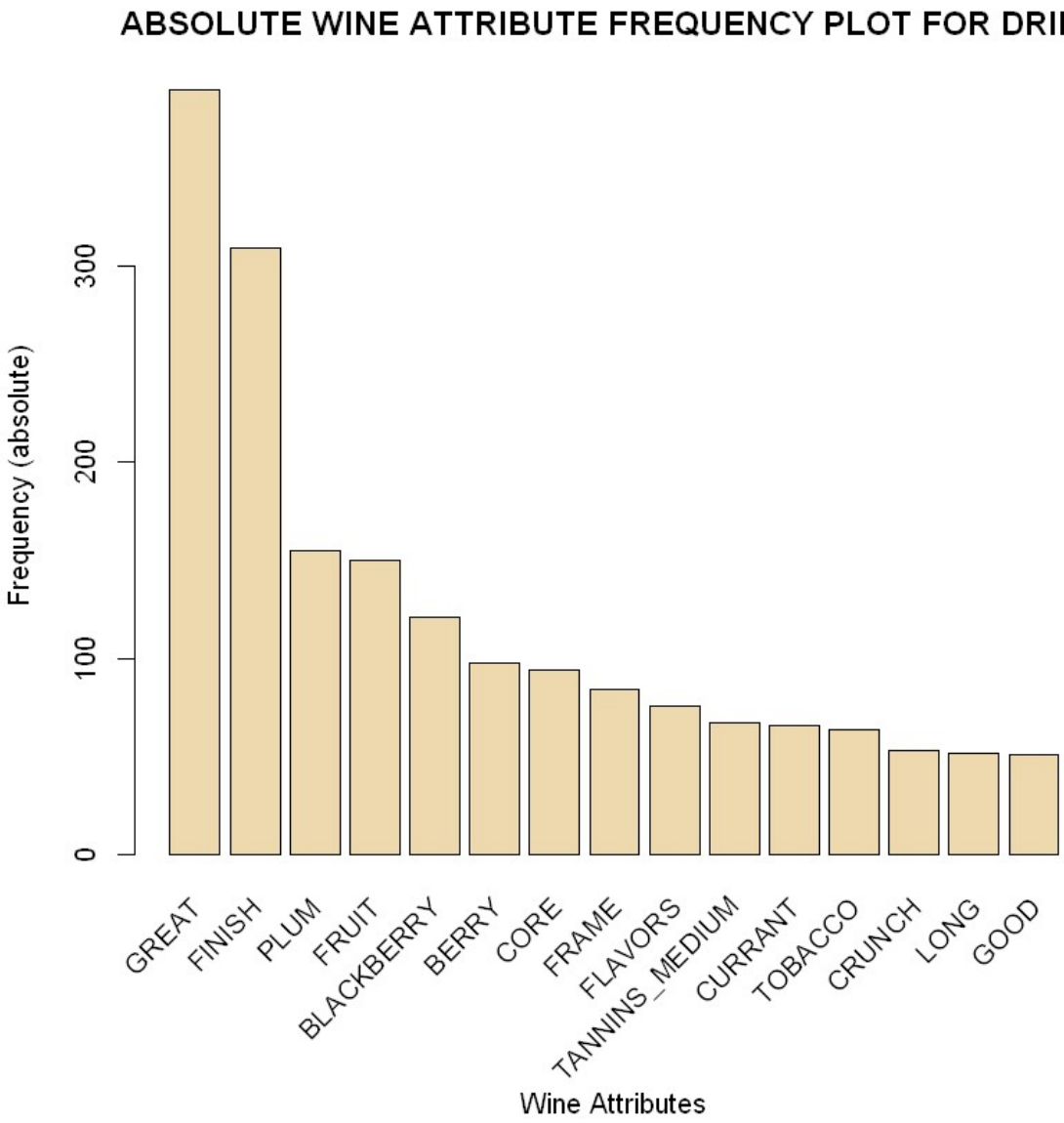
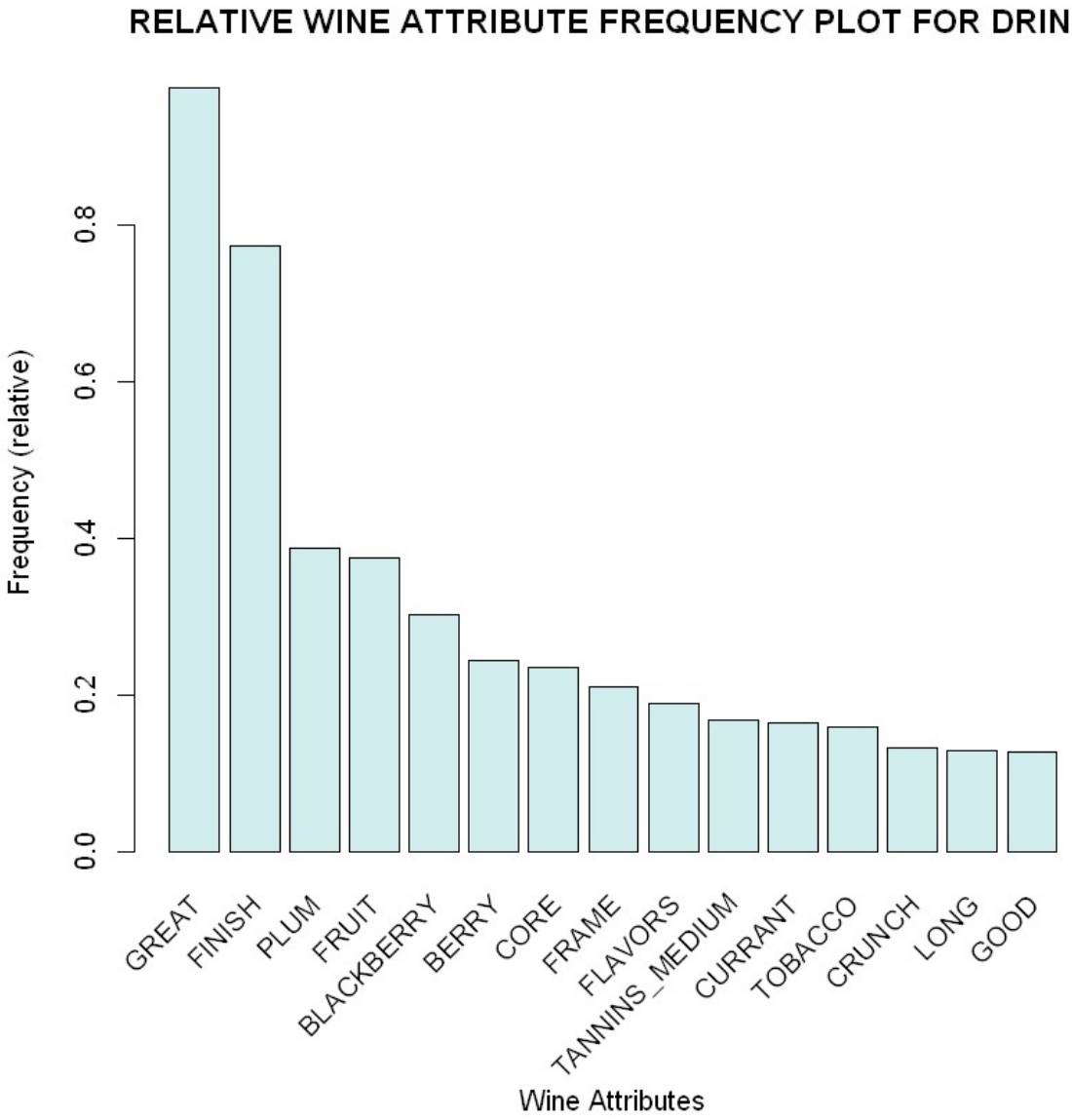
In [56]:
```
# Load libraries
library(tidyverse) # data manipulation
library(arules) # mining association rules and frequent itemsets
library(arulesViz) # visualization techniques for association rules
library(knitr) # dynamic report generation
library(gridExtra) # provides a number of user-level functions to work with "grid"
graphics
library(lubridate) # work with dates and times

# Read the data
trans <- read.transactions("./drink_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
 761 items (columns)
```

In [57]:
```
# Absolute Item Frequency Plot
itemFrequencyPlot(trans, topN=15, type="absolute", col="wheat2",xlab="Wine Attributes",
                  ylab="Frequency (absolute)", main="ABSOLUTE WINE ATTRIBUTE FREQUENCY PLOT FOR DRINK")
```



In [43]:
```
itemFrequencyPlot(trans, topN=15, type="relative", col="lightcyan2", xlab="Wine Attributes",
                  ylab="Frequency (relative)", main="RELATIVE WINE ATTRIBUTE FREQUENCY PLOT FOR DRINK")
```



In [58]:
```
# Support and confidence values
supportLevels <- c(0.7, 0.4, 0.2)
confidenceLevels <- c(0.8, 0.5, 0.3)

# Empty integers
rules_sup70 <- integer(length=9)
rules_sup40 <- integer(length=9)
rules_sup20 <- integer(length=9)

# Apriori algorithm with a support level of 70 and confidence of 80%

rules_sup70 <- length(apriori(trans, parameter=list(sup=supportLevels[1],
                                    conf=confidenceLevels[1], target="rules")))
rules_sup70

# 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     0.7      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: 280

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [2 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

2

```
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: 160

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [2 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [4 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

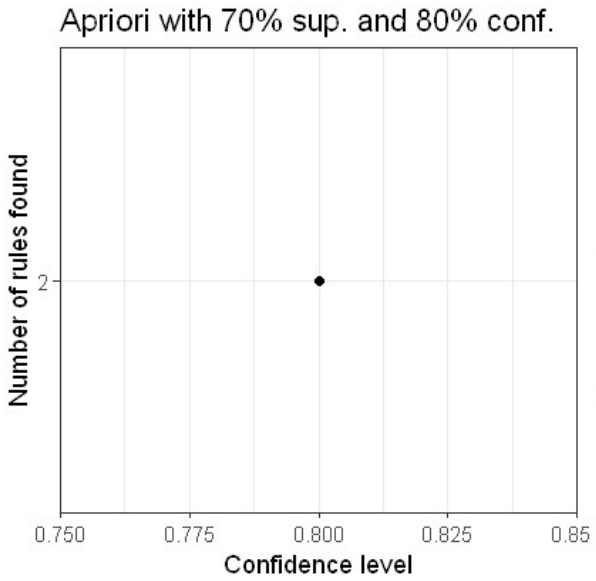
4

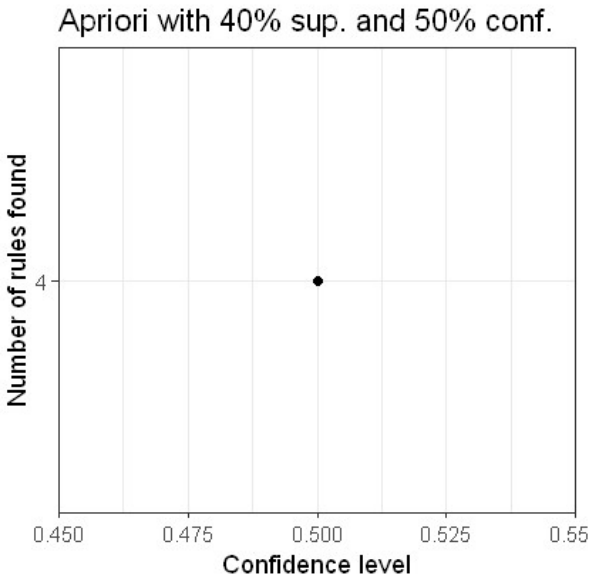
```
Apriori

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: 80

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [30 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

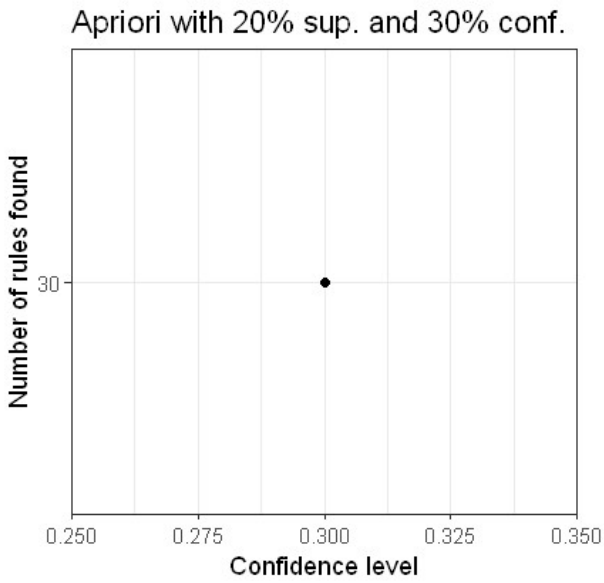
30

In [55]:
```
# Number of rules found with a support level of 70%
plot1 <- qplot(confidenceLevels[1], rules_sup70, geom=c("point", "line"),
               xlab="Confidence level", ylab="Number of rules found",
               main="Apriori with 70% sup. and 80% conf.") +   scale_y_continuous(b
reaks=seq(0, 10, 2))+
  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(0, 50, 10)) +
  theme_bw()


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

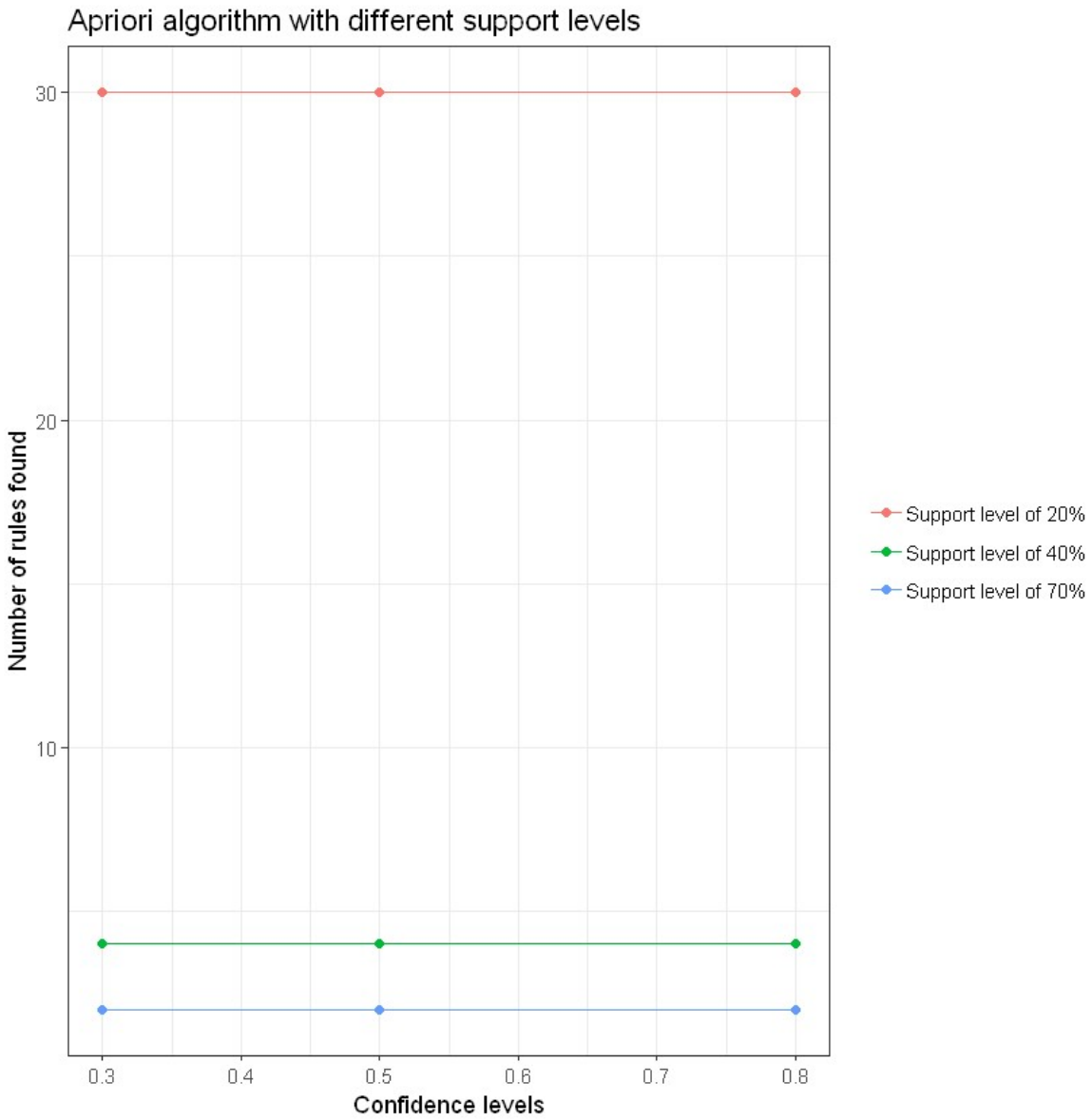
```
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?
geom_path: Each group consists of only one observation. Do you need to adjust
the group aesthetic?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







In [46]:
```
# Data frame
num_rules <- data.frame(rules_sup70, 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_sup70, colour="Support level of 70%")) +
  geom_point(aes(y=rules_sup70, colour="Support level of 70%")) +

  # 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
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 70%

In [62]:
```
# Apriori algorithm execution with a support level of 7% and a confidence level of
80%
rules_sup1_conf80 <- apriori(trans, parameter=list(sup=0.7,
                             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     0.7      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: 280

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [2 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 2 rules
```

In [63]:
```
# 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: 160

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [2 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [4 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 4 rules
```

In [64]:
```
# 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: 80

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[761 item(s), 400 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [30 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

set of 30 rules
```

In [65]:
```
# Inspect association rules
inspect(rules_sup1_conf80)
```

```
    lhs          rhs     support confidence lift     count
[1] {}       => {GREAT} 0.9750  0.9750000  1.000000 390
[2] {FINISH} => {GREAT} 0.7575  0.9805825  1.005726 303
```

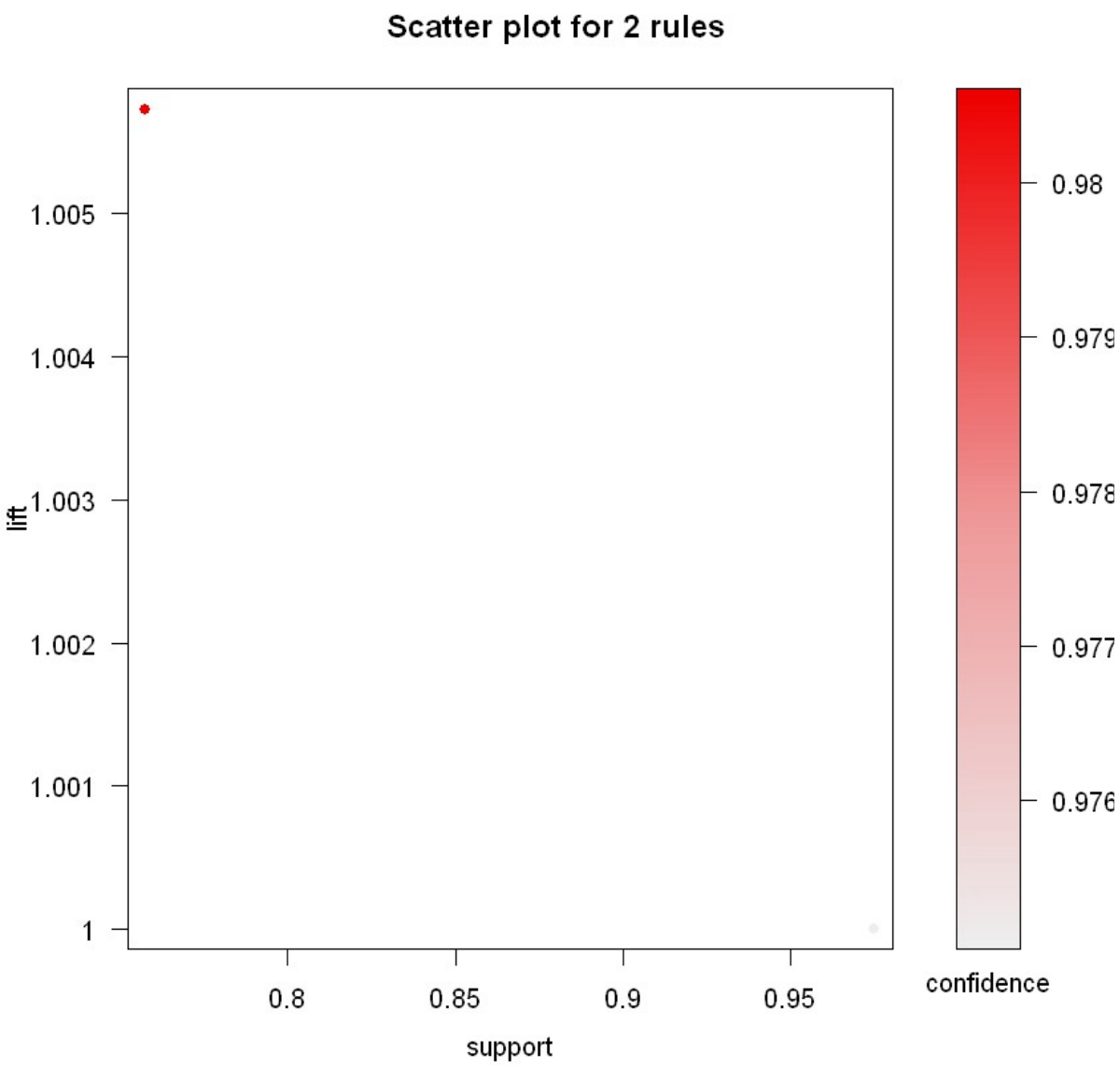
In [66]:
```
# Inspect association rules
inspect(rules_sup1_conf50)
```

```
    lhs          rhs      support confidence lift     count
[1] {}       => {FINISH} 0.7725  0.7725000  1.000000 309
[2] {}       => {GREAT}  0.9750  0.9750000  1.000000 390
[3] {FINISH} => {GREAT}  0.7575  0.9805825  1.005726 303
[4] {GREAT}  => {FINISH} 0.7575  0.7769231  1.005726 303
```

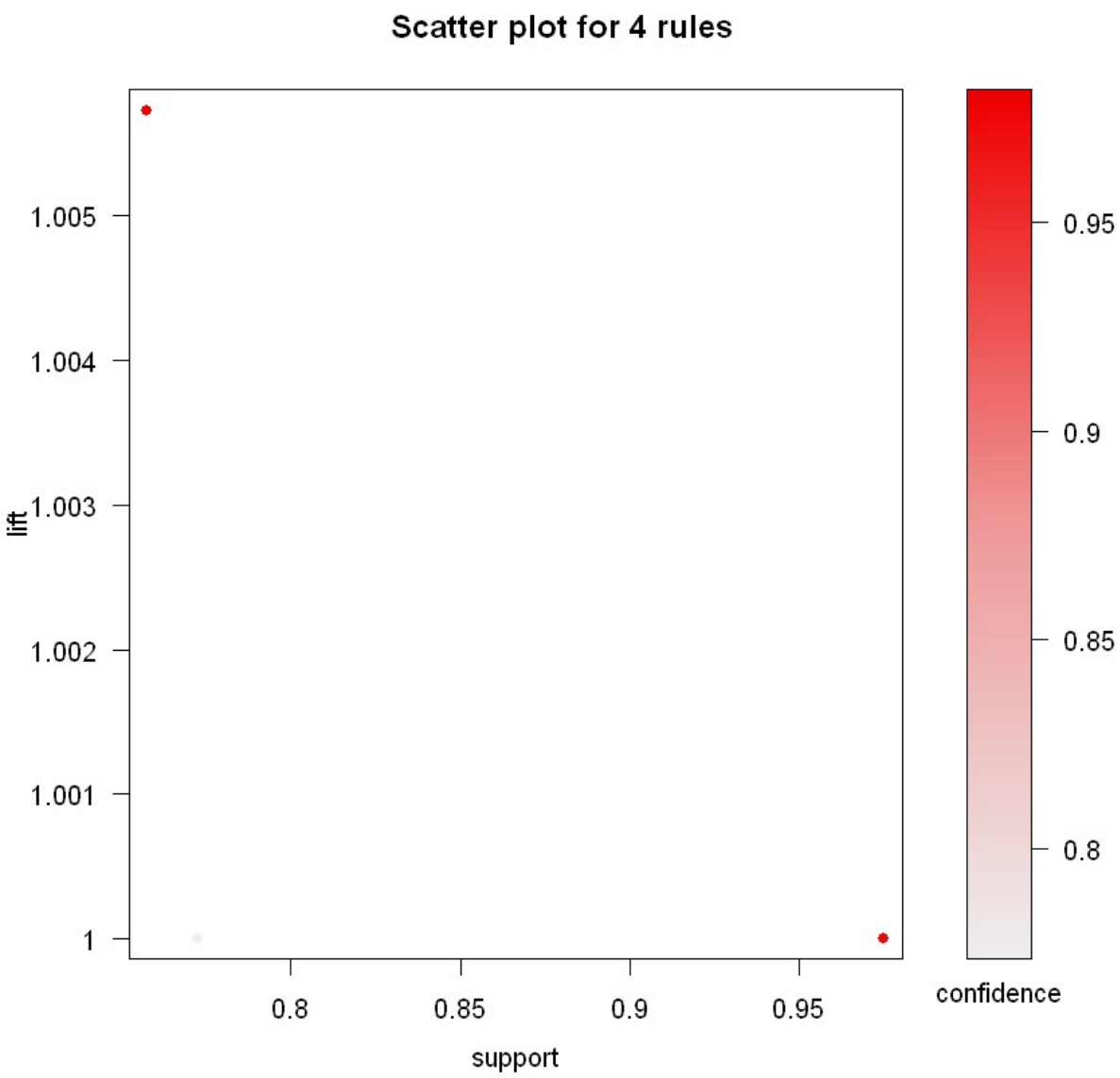
In [67]:
```
# Inspect association rules
inspect(rules_sup1_conf30)
```

```
     lhs                     rhs          support confidence lift      count
[1]  {}                   => {BLACKBERRY} 0.3025  0.3025000  1.0000000 121
[2]  {}                   => {FRUIT}      0.3750  0.3750000  1.0000000 150
[3]  {}                   => {PLUM}       0.3875  0.3875000  1.0000000 155
[4]  {}                   => {FINISH}     0.7725  0.7725000  1.0000000 309
[5]  {}                   => {GREAT}      0.9750  0.9750000  1.0000000 390
[6]  {FRAME}              => {GREAT}      0.2050  0.9761905  1.0012210  82
[7]  {BERRY}              => {GREAT}      0.2425  0.9897959  1.0151753  97
[8]  {CORE}               => {GREAT}      0.2325  0.9893617  1.0147300  93
[9]  {BLACKBERRY}         => {FINISH}     0.2400  0.7933884  1.0270400  96
[10] {FINISH}             => {BLACKBERRY} 0.2400  0.3106796  1.0270400  96
[11] {BLACKBERRY}         => {GREAT}      0.2900  0.9586777  0.9832592 116
[12] {FRUIT}              => {FINISH}     0.2975  0.7933333  1.0269687 119
[13] {FINISH}             => {FRUIT}      0.2975  0.3851133  1.0269687 119
[14] {FRUIT}              => {GREAT}      0.3650  0.9733333  0.9982906 146
[15] {GREAT}              => {FRUIT}      0.3650  0.3743590  0.9982906 146
[16] {PLUM}               => {FINISH}     0.3200  0.8258065  1.0690051 128
[17] {FINISH}             => {PLUM}       0.3200  0.4142395  1.0690051 128
[18] {PLUM}               => {GREAT}      0.3775  0.9741935  0.9991729 151
[19] {GREAT}              => {PLUM}       0.3775  0.3871795  0.9991729 151
[20] {FINISH}             => {GREAT}      0.7575  0.9805825  1.0057257 303
[21] {GREAT}              => {FINISH}     0.7575  0.7769231  1.0057257 303
[22] {BLACKBERRY,FINISH} => {GREAT}       0.2300  0.9583333  0.9829060  92
[23] {BLACKBERRY,GREAT}  => {FINISH}      0.2300  0.7931034  1.0266711  92
[24] {FINISH,GREAT}       => {BLACKBERRY} 0.2300  0.3036304  1.0037367  92
[25] {FINISH,FRUIT}       => {GREAT}      0.2875  0.9663866  0.9911657 115
[26] {FRUIT,GREAT}        => {FINISH}     0.2875  0.7876712  1.0196391 115
[27] {FINISH,GREAT}       => {FRUIT}      0.2875  0.3795380  1.0121012 115
[28] {FINISH,PLUM}        => {GREAT}      0.3125  0.9765625  1.0016026 125
[29] {GREAT,PLUM}         => {FINISH}     0.3125  0.8278146  1.0716046 125
[30] {FINISH,GREAT}       => {PLUM}       0.3125  0.4125413  1.0646226 125
```

In [69]:
```
# Scatter plot
plot(rules_sup1_conf80, measure=c("support","lift"), shading="confidence")
```

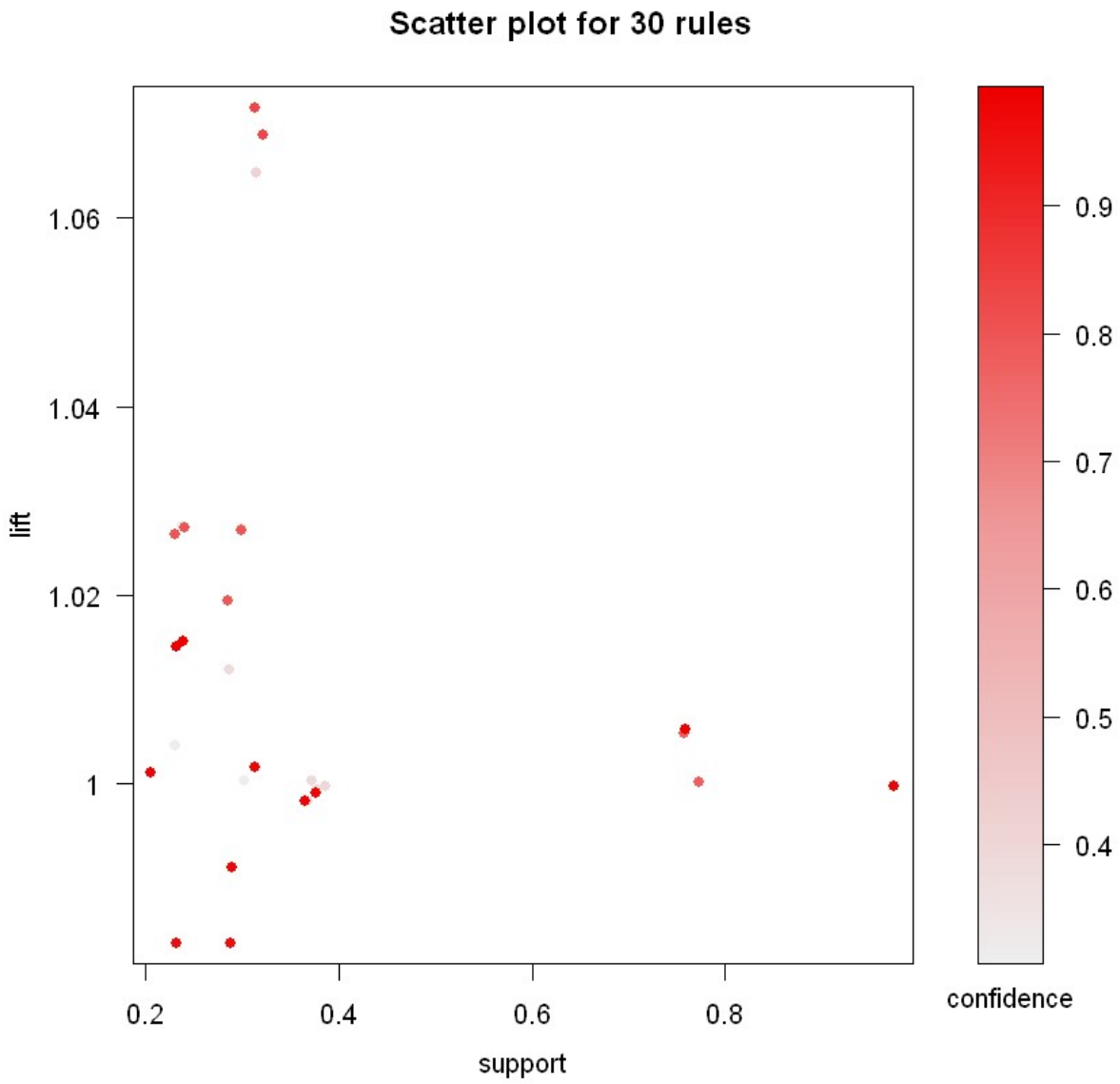


In [70]:
```
# Scatter plot
plot(rules_sup1_conf50, measure=c("support","lift"), shading="confidence")
```

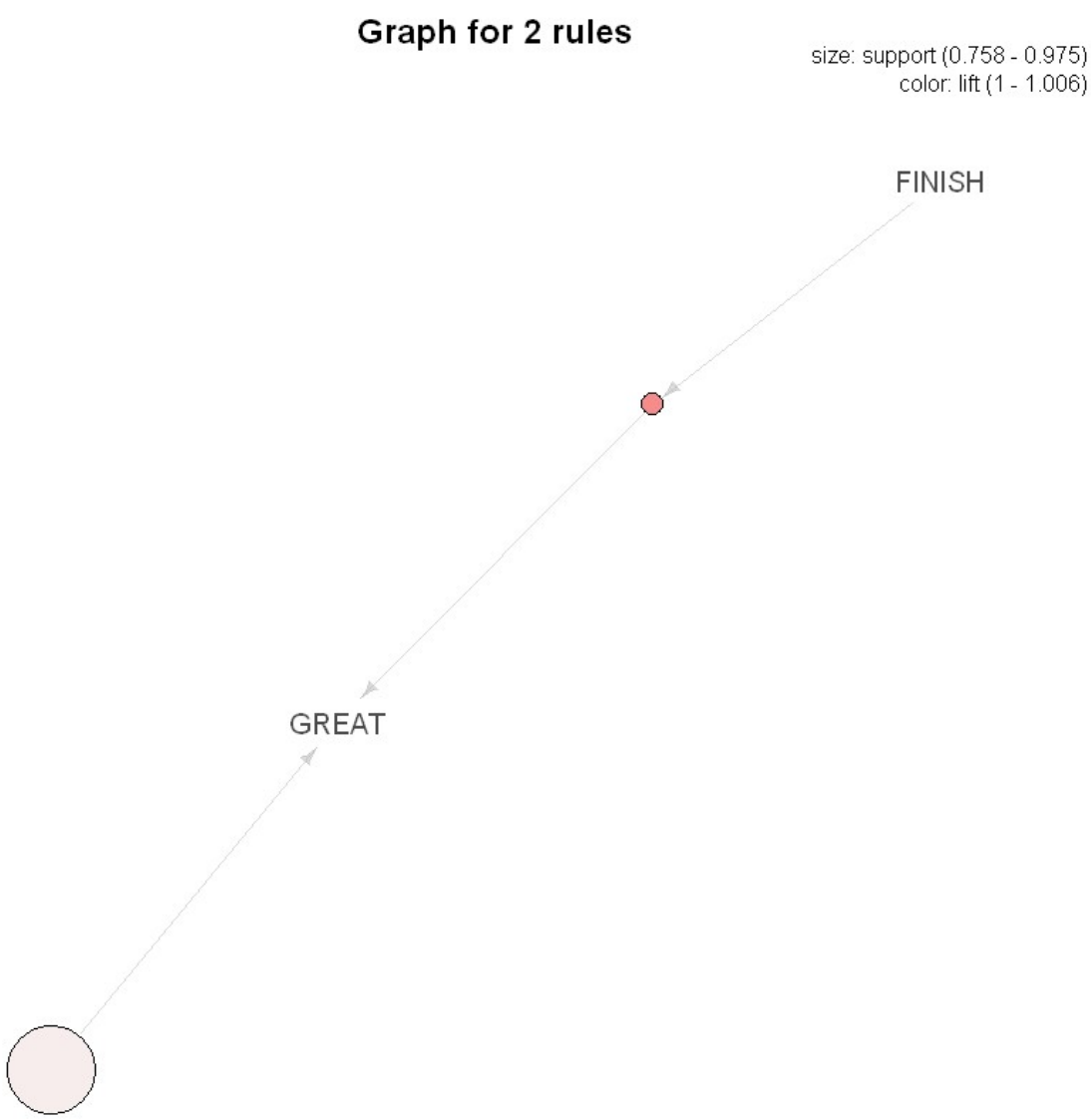


In [71]:
```
# Scatter plot
plot(rules_sup1_conf30, measure=c("support","lift"), shading="confidence")
```

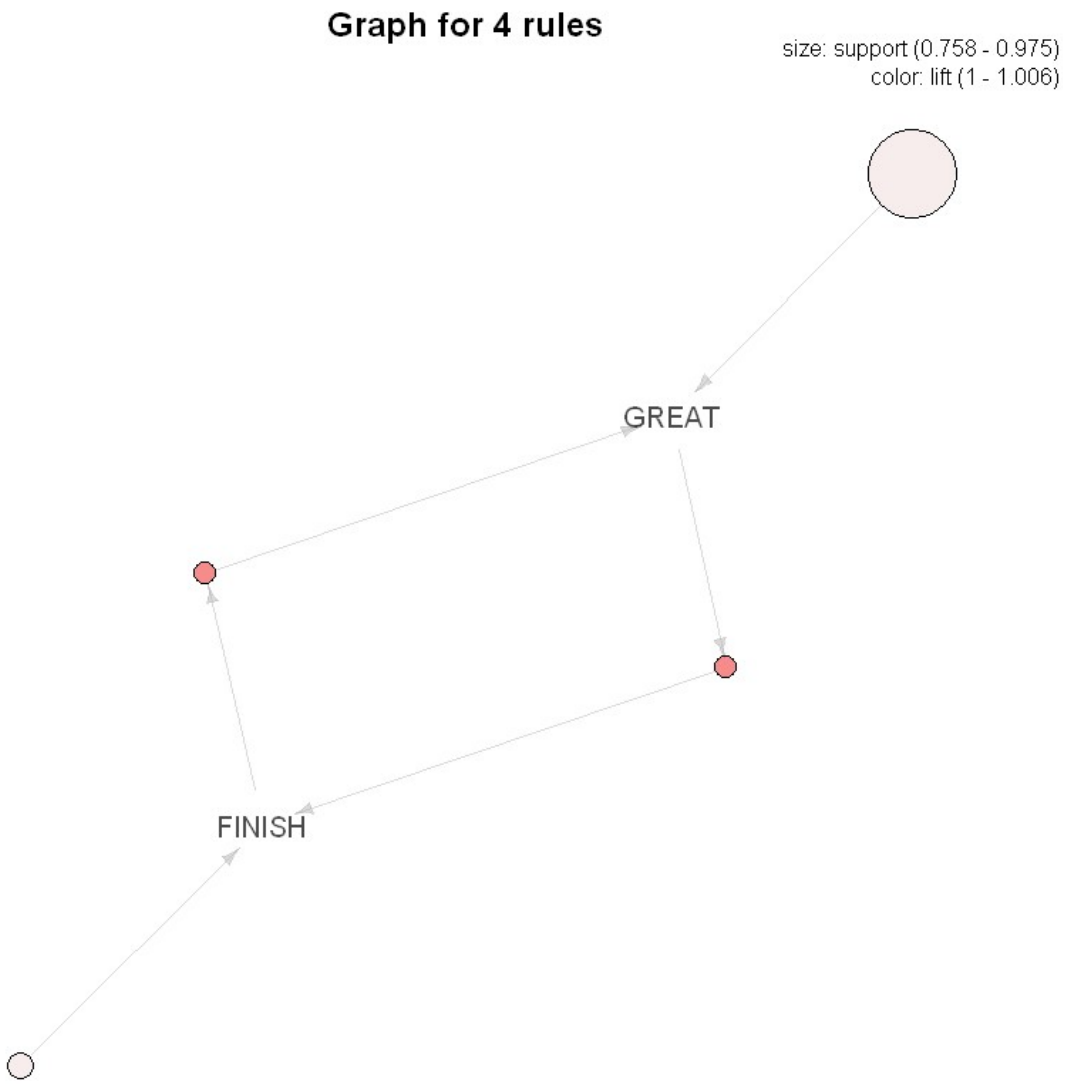
```
To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



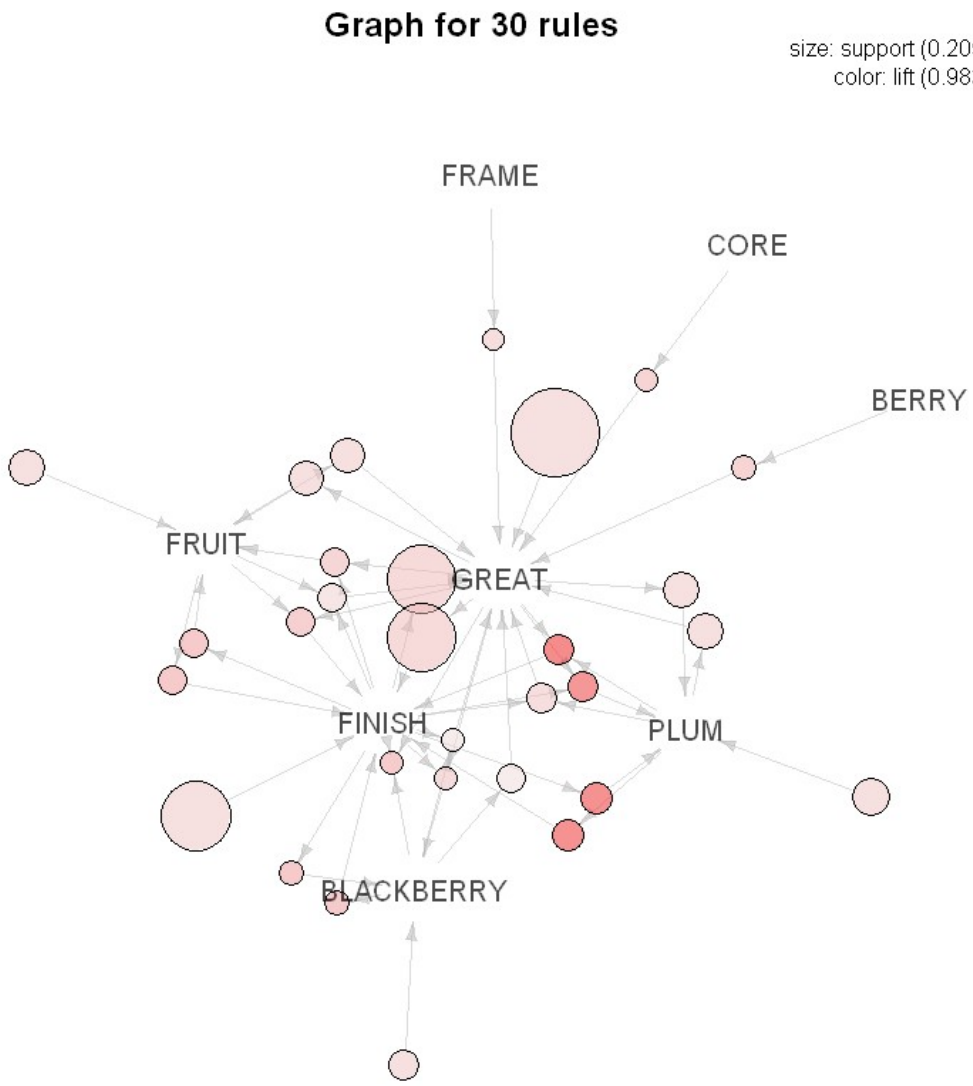
In [72]:
```
plot(rules_sup1_conf80, method="graph")
```



In [73]:
```
plot(rules_sup1_conf50, method="graph")
```



In [74]:
```
plot(rules_sup1_conf30, method="graph")
```



In [ ]: