support of 2, because only two transactions included red, white, and green faceplates.

> In R, the user will input data using the package arules in the transactions database format. The package only creates rules with one item as the consequent. It calls the consequent the *right-hand side (RHS)* of the rule, and the antecedent the *left-hand-side (LHS)* of the rule.

The Process of Rule Selection

The process of selecting strong rules is based on generating all association rules that meet stipulated support and confidence requirements. This is done in two stages. The first stage, described earlier, consists of finding all "frequent" itemsets, those itemsets that have a requisite support. In the second stage, we generate, from the frequent itemsets, association rules that meet a confidence requirement. The first step is aimed at removing item combinations that are rare in the database. The second stage then filters the remaining rules and selects only those with high confidence. For most association analysis data, the computational challenge is the first stage, as described in the discussion of the Apriori algorithm.

The computation of confidence in the second stage is simple. Since any subset (e.g., {red} in the phone faceplate example) must occur at least as frequently as the set it belongs to (e.g., {red, white}), each subset will also be in the list. It is then straightforward to compute the confidence as the ratio of the support for the itemset to the support for each subset of the itemset. We retain the corresponding association rule only if it exceeds the desired cutoff value for confidence. For example, from the itemset {red, white, green} in the phone faceplate purchases, we get the following single-consequent association rules, confidence values, and lift values:

Rule	Confidence	Lift
{red, white} ⇒ {green}	$\frac{\text{support of \{red, white, green\}}}{\text{support of \{red, white\}}} = 2/4 = 50\%$	$\frac{\text{confidence of rule}}{\text{benchmark confidence}} = \frac{50\%}{20\%} = 2.5$
{green} ⇒ {red}	$\frac{\text{support of \{green, red\}}}{\text{support of \{green\}}} = 2/2 = 100\%$	$\frac{\text{confidence of rule}}{\text{benchmark confidence}} = \frac{100\%}{60\%} = 1.67$
{white, green} ⇒ {red}	$\frac{\text{support of \{white, green, red\}}}{\text{support of \{white, green\}}} = 2/2 = 100\%$	$\frac{\text{confidence of rule}}{\text{benchmark confidence}} = \frac{100\%}{60\%} = 1.67$

If the desired minimum confidence is 70%, we would report only the second and third rules.

We can generate association rules in R by coercing the binary incidence matrix in Table 14.2 into a transaction database like in Table 14.1. We specify the minimum support (20%) and minimum confidence level percentage (50%). Table 14.4 shows the output. The output includes information on each rule and its support, confidence, and lift (Note that here we consider all possible itemsets, not just {red, white, green} as above.).

Table 14.4 Binary Incidence Matrix, Transactions Database, and Rules for Faceplate Example

code for running the Apriori algorithm

```
fp.df <- read.csv("Faceplate.csv")

# remove first column and convert to matrix
fp.mat <- as.matrix(fp.df[, -1])

# convert the binary incidence matrix into a transactions database
fp.trans <- as(fp.mat, "transactions")
inspect(fp.trans)

## get rules
# when running apriori(), include the minimum support, minimum confidence, and target
# as arguments.
rules <- apriori(fp.trans, parameter = list(supp = 0.2, conf = 0.5, target = "rules"))

# inspect the first six rules, sorted by their lift
inspect(head(sort(rules, by = "lift"), n = 6))
inspect(head(sort(rules, by = "lift"), n = 6))

Output
> fp.mat
      Red White Blue Orange Green Yellow
 [1,]   1     1    0      0     1      0
 [2,]   0     1    0      1     0      0
 [3,]   0     1    1      0     0      0
 [4,]   1     1    0      1     0      0
 [5,]   1     0    1      0     0      0
 [6,]   0     1    1      0     0      0
 [7,]   1     0    1      0     0      0
 [8,]   1     1    1      0     1      0
 [9,]   1     1    1      0     0      0
[10,]   0     0    0      0     0      1

> inspect(fp.trans)
   items
1  {Red,White,Green}
2  {White,Orange}
3  {White,Blue}
4  {Red,White,Orange}
5  {Red,Blue}
6  {White,Blue}
7  {Red,Blue}
8  {Red,White,Blue,Green}
9  {Red,White,Blue}
10 {Yellow}

> inspect(head(sort(rules, by = "lift"), n = 6))
   lhs              rhs     support confidence lift
15 {Red,White}   => {Green} 0.2     0.5        2.500000
5  {Green}       => {Red}   0.2     1.0        1.666667
14 {White,Green} => {Red}   0.2     1.0        1.666667
4  {Orange}      => {White} 0.2     1.0        1.428571
6  {Green}       => {White} 0.2     1.0        1.428571
13 {Red,Green}   => {White} 0.2     1.0        1.428571
```

Interpreting the Results

We can translate each of the rules from Table 14.5 into an understandable sentence that provides information about performance. For example, we can read rule #4 as follows:

> If orange is purchased, then with confidence 100% white will also be purchased. This rule has a lift ratio of 1.43.

In interpreting results, it is useful to look at the various measures. The support for the rule indicates its impact in terms of overall size: How many transactions are affected? If only a small number of transactions are affected, the rule may be of little use (unless the consequent is very valuable and/or the rule is very efficient in finding it).

The lift ratio indicates how efficient the rule is in finding consequents, compared to random selection. A very efficient rule is preferred to an inefficient rule, but we must still consider support: A very efficient rule that has very low support may not be as desirable as a less efficient rule with much greater support.

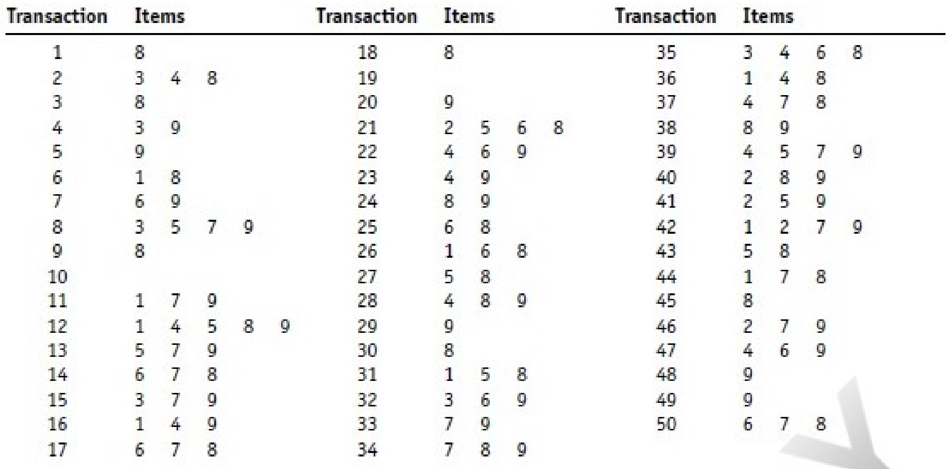
The confidence tells us at what rate consequents will be found, and is useful in determining the business or operational usefulness of a rule: A rule with low confidence may find consequents at too low a rate to be worth the cost of (say) promoting the consequent in all the transactions that involve the antecedent.

Rules and Chance

What about confidence in the nontechnical sense? How sure can we be that the rules we develop are meaningful? Considering the matter from a statistical perspective, we can ask: Are we finding associations that are really just chance occurrences?

Let us examine the output from an application of this algorithm to a small database of 50 transactions, where each of the nine items is assigned randomly to each transaction. The data are shown in Table 14.5, and the association rules generated are shown in Table 14.6. In looking at these tables, remember that “lhs” and “rhs” refer to itemsets, not records.

Table 14.5 Fifty Transactions of Randomly Assigned Items



Transaction	Items	Transaction	Items	Transaction	Items
1	8	18	8	35	3 4 6 8
2	3 4 8	19		36	1 4 8
3	8	20	9	37	4 7 8
4	3 9	21	2 5 6 8	38	8 9
5	9	22	4 6 9	39	4 5 7 9
6	1 8	23	4 9	40	2 8 9
7	6 9	24	8 9	41	2 5 9
8	3 5 7 9	25	6 8	42	1 2 7 9
9	8	26	1 6 8	43	5 8
10		27	5 8	44	1 7 8
11	1 7 9	28	4 8 9	45	8
12	1 4 5 8 9	29	9	46	2 7 9
13	5 7 9	30	8	47	4 6 9
14	6 7 8	31	1 5 8	48	9
15	3 7 9	32	3 6 9	49	9
16	1 4 9	33	7 9	50	6 7 8
17	6 7 8	34	7 8 9		

Table 14.6 Association Rules Output for Random Data

Min. Support:	**math 2 = 4%**
Min. Conf. % :	**math 70**

```
> rules.tbl[rules.tbl$support >= 0.04 & rules.tbl$confidence >= 0.7,]
                  lhs         rhs support confidence      lift
[18]         {item.2} => {item.9}    0.08        0.8 1.481481
[89]  {item.2,item.7} => {item.9}    0.04        1.0 1.851852
[104] {item.3,item.4} => {item.8}    0.04        1.0 1.851852
[105] {item.3,item.8} => {item.4}    0.04        1.0 5.000000
[113] {item.3,item.7} => {item.9}    0.04        1.0 1.851852
[119] {item.1,item.5} => {item.8}    0.04        1.0 1.851852
[149] {item.4,item.5} => {item.9}    0.04        1.0 1.851852
[155] {item.5,item.7} => {item.9}    0.06        1.0 1.851852
[176] {item.6,item.7} => {item.8}    0.06        1.0 1.851852
```

R code is provided in the next example

In this example, the lift ratios highlight Rule [105] as most interesting, as it suggests that purchase of item 4 is almost five times as likely when items 3 and 8 are purchased than if item 4 was not associated with the itemset {3,8}. Yet we know there is no fundamental association underlying these data—they were generated randomly.

Two principles can guide us in assessing rules for possible spuriousness due to chance effects:

1. The more records the rule is based on, the more solid the conclusion.
2. The more distinct rules we consider seriously (perhaps consolidating multiple rules that deal with the same items), the more likely it is that at least some will be based on chance sampling results. For one person to toss a coin 10 times and get 10 heads would be quite surprising. If 1000 people toss a coin 10 times each, it would not be nearly so surprising to have one get 10 heads. Formal adjustment of "statistical significance" when multiple comparisons are made is a complex subject in its own right, and beyond the scope of this book. A reasonable approach is to consider rules from the top-down in terms of business or operational applicability, and not consider more than what can reasonably be incorporated in a human decision-making process. This will impose a rough constraint on the dangers that arise from an automated review of hundreds or thousands of rules in search of "something interesting."

We now consider a more realistic example, using a larger database and real transactional data.

Example 2: Rules for Similar Book Purchases

The following example (drawn from the Charles Book Club case; see Chapter 21) examines associations among transactions involving various types of books. The database includes 2000 transactions, and there are 11 different types of books. The data, in binary incidence matrix form, are shown in Table 14.7.

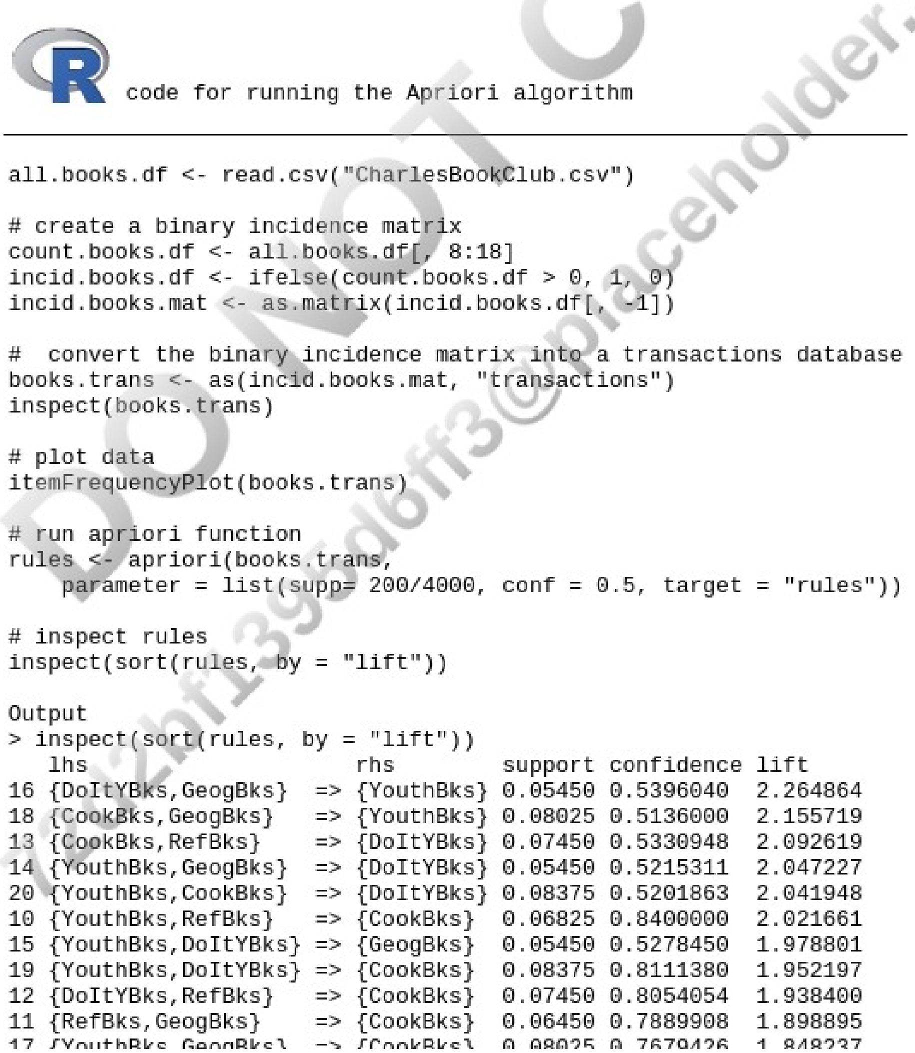
Table 14.7 Subset of book purchase transactions in binary matrix format

Table 14.7 Subset of book purchase transactions in binary matrix format

ChildBks	YouthBks	CookBks	DoItYBks	cefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt	Florence
0	1	0	1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	1	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

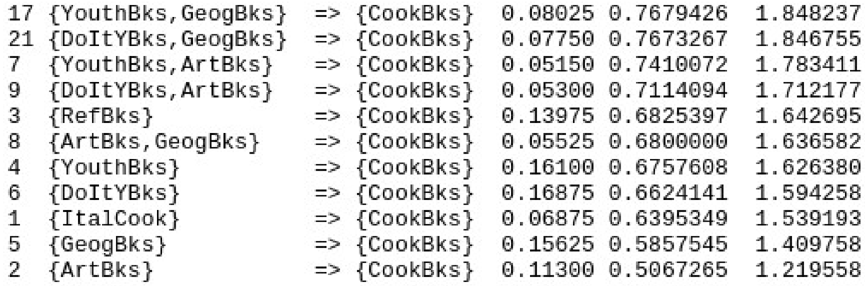
For instance, the first transaction included *YouthBks* (youth books) *DoItYBks* (do-it-yourself books), and *GeogBks* (geography books). Table 14.8 shows some of the rules generated for these data, given that we specified a minimal support of 200 transactions (out of 4000 transactions) and a minimal confidence of 50%. This resulted in 21 rules (see Table 14.8).

Table 14.8 Rules for book purchase transactions



code for running the Apriori algorithm

```
all.books.df <- read.csv("CharlesBookClub.csv")

# create a binary incidence matrix
count.books.df <- all.books.df[, 8:18]
incid.books.df <- ifelse(count.books.df > 0, 1, 0)
incid.books.mat <- as.matrix(incid.books.df[, -1])

#  convert the binary incidence matrix into a transactions database
books.trans <- as(incid.books.mat, "transactions")
inspect(books.trans)

# plot data
itemFrequencyPlot(books.trans)

# run apriori function
rules <- apriori(books.trans,
    parameter = list(supp= 200/4000, conf = 0.5, target = "rules"))

# inspect rules
inspect(sort(rules, by = "lift"))

Output
> inspect(sort(rules, by = "lift"))
   lhs                     rhs         support confidence lift
16 {DoItYBks,GeogBks}  => {YouthBks} 0.05450 0.5396040  2.264864
18 {CookBks,GeogBks}   => {YouthBks} 0.08025 0.5136000  2.155719
13 {CookBks,RefBks}    => {DoItYBks} 0.07450 0.5330948  2.092619
14 {YouthBks,GeogBks}  => {DoItYBks} 0.05450 0.5215311  2.047227
20 {YouthBks,CookBks}  => {DoItYBks} 0.08375 0.5201863  2.041948
10 {YouthBks,RefBks}   => {CookBks}  0.06825 0.8400000  2.021661
15 {YouthBks,DoItYBks} => {GeogBks}  0.05450 0.5278450  1.978801
19 {YouthBks,DoItYBks} => {CookBks}  0.08375 0.8111380  1.952197
12 {DoItYBks,RefBks}   => {CookBks}  0.07450 0.8054054  1.938400
11 {RefBks,GeogBks}    => {CookBks}  0.06450 0.7889908  1.898895
17 {YouthBks,GeogBks}  => {CookBks}  0.08025 0.7679426  1.848237
```



```
17 {YouthBks,GeogBks}  => {CookBks}  0.08025 0.7679426  1.848237
21 {DoItYBks,GeogBks}  => {CookBks}  0.07750 0.7673267  1.846755
7  {YouthBks,ArtBks}   => {CookBks}  0.05150 0.7410072  1.783411
9  {DoItYBks,ArtBks}   => {CookBks}  0.05300 0.7114094  1.712177
3  {RefBks}            => {CookBks}  0.13975 0.6825397  1.642695
8  {ArtBks,GeogBks}    => {CookBks}  0.05525 0.6800000  1.636582
4  {YouthBks}          => {CookBks}  0.16100 0.6757608  1.626380
6  {DoItYBks}          => {CookBks}  0.16875 0.6624141  1.594258
1  {ItalCook}          => {CookBks}  0.06875 0.6395349  1.539193
5  {GeogBks}           => {CookBks}  0.15625 0.5857545  1.409758
2  {ArtBks}            => {CookBks}  0.11300 0.5067265  1.219558
```