Week 2

Association Rules

2.1 Market Basket Assessment

Companies use loyalty cards to gain market advantage by leveraging this data along with your personal demographic information. When you make purchases at many stores, the information on what you bought is recorded and stored. Analyses involving such information are often referred to as *Customer Relationship Management* (CRM) models. *Association rule mining* is a method that seeks to find products that are frequently purchased together.

We can see an example of the use of association rule mining at Amazon.com. When you search for a particular book, you also are presented with a list of books that other people, who have bought the book that you are looking at, have purchased. Amazon.com is grouping together products that have a high probability of appearing in the same ”market basket” in an attempt to get you to add them to *your* basket.

Grocery stores are another example of the use of association rule mining. Placement of products in the store and the offers of weekly specials are designed to encourage you to add items to your shopping basket. As an example of association rule mining, we are going to simulate and then analyze the shopping baskets of 2733 shoppers whose grocery carts can contain (for simplicity) only items from the list of Milk, Peanut.Butter, Bread, Cereal, Jelly.. The following code simulates these shopping baskets:

```
n.baskets <- 2733
numb <- 5                          # Number of items in baskets
   # Simulation of baskets
set.seed(12345, ”default”)         # Try to make replication possible
   # Create a matrix to represent the basket (holds 5 items)
baskets <- matrix(0, n.baskets, numb)
heading <- c(”Milk”, ”Peanut.Butter”, ”Bread”, ”Cereal”, ”Jelly”)
dimnames(baskets) <- list (NULL, heading)
   # Put ”Peanut Butter”, ”Bread”,”Jelly” in baskets 1-82
baskets[1:82, c(”Peanut Butter”, ”Bread”,”Jelly”)] <- 1
   # and ”Peanut Butter”,”Jelly” in 83 to 100
baskets[83:100,c(”Peanut Butter”,”Jelly”)] <- 1
   # (P)eanut Butter & (J)elly can only appear together
   #         in the first 100 baskets
   # n.@, b.@, e.@ represent the number of the item @,
   #     the starting and ending baskets
   # sample(100,n.@) produces n.@ random integers from 1-100
   # Scatter some (M)ilk and (C)ereal in the first 100 baskets
n.M <- 60
baskets[sample(100, n.M), c(”Milk”)] <- 1
n.C <- 55
```

```
baskets[sample(100, n.C), c("Cereal")] <- 1
   # sample(e.@ - b.@, n.@) + b.@
   # e.@ - b.@ gives the range [1, e.@ - b.@] from which to choose n.J
   # adding b.@ makes the ramge [b.@+1, e.@ - b.@ + b.@]
n.J <- 800; b.J <- 100;  e.J <- 1503
baskets[sample(e.J - b.J, n.J) + b.J, c("Jelly")] <- 1
n.P <- 700; b.P <- 1514;  e.P <- 2733
baskets[sample(e.P - b.P, n.P) + b.P, c("Peanut.Butter")] <- 1
   # Now put more Milk, Cereal, and Bread in some baskets
n.M <- 500; b.M <- 100;  e.M <- 1103
baskets[sample(e.M - b.M, n.M) + b.M, c("Milk")] <- 1
n.M <- 450; b.M <- 200;  e.M <- 2733
baskets[sample(e.M - b.M, n.M) + b.M, c("Milk")] <- 1
n.C <- 780; b.C <- 600; e.C <- 1600
baskets[sample(e.C - b.C, n.C) + b.C, c("Cereal")] <- 1
n.B <- 1200; b.B <- 100;  e.B <- 2733
baskets[sample(e.B - b.B, n.B) + b.B, c("Bread")] <- 1
   # End of simulation
```

We can find out the total for each commodity across the baskets:

```
(item.in.basket <- apply (baskets, 2, sum))
        Milk Peanut.Butter     Bread        Cereal      Jelly
          932          800          1282          835          900
```

or the percentage of baskets containing each commodity:

```
(percent.in.basket <- round(item.in.basket/n.baskets*100, 2))
        Milk Peanut.Butter     Bread        Cereal      Jelly
        34.10         29.27        46.90        30.55        32.93
```

For analyzing this data, we want to have all possible pairs, triples, and quadruples of the 5 items so:

All unique pairs

```
n.comb.2 <- choose(numb,2)
double <- matrix(0, n.comb.2, 2)
ind <- 1
for (i in 1:numb) {
  j <- i+1
  while (j <= numb) {  # for can run 6:5 ...
      double[ind,] <- c(i,j)
      ind <- ind + 1
      j <- j + 1
  }
}
# double
matrix(heading[double], n.comb.2, 2)
      [,1]             [,2]
 [1,] "Milk"           "Peanut.Butter"
 [2,] "Milk"           "Bread"
 [3,] "Milk"           "Cereal"
 [4,] "Milk"           "Jelly"
```

```
 [5,] "Peanut.Butter" "Bread"
 [6,] "Peanut.Butter" "Cereal"
 [7,] "Peanut.Butter" "Jelly"
 [8,] "Bread"         "Cereal"
 [9,] "Bread"         "Jelly"
[10,] "Cereal"        "Jelly"
```

Count the number of times each pair occurs:

```
double.counts <- matrix(0, n.comb.2, 3)
for (i in 1:dim(double)[1]){
double.counts[i,] <-c(double[i,],sum(floor(apply(baskets[,double[i,]],1,sum)/2)))
}
matrix(c(heading[double.counts[,1:2]], double.counts[,3]), n.comb.2, 3)
      [,1]            [,2]            [,3]
 [1,] "Milk"          "Peanut.Butter" "191"
 [2,] "Milk"          "Bread"         "445"
 [3,] "Milk"          "Cereal"        "322"
 [4,] "Milk"          "Jelly"         "419"
 [5,] "Peanut.Butter" "Bread"         "396"
 [6,] "Peanut.Butter" "Cereal"         "88"
 [7,] "Peanut.Butter" "Jelly"         "100"
 [8,] "Bread"         "Cereal"        "393"
 [9,] "Bread"         "Jelly"         "438"
[10,] "Cereal"        "Jelly"         "479"
```

Notice that a matrix has been used to store the information. Because *some* items are strings, the matrix treats **all** the items as strings.

All unique triples

```
n.comb.3 <- choose(numb,3)
triple <- matrix(0, n.comb.3, 3)
ind <- 1
for (i in 1:numb) {
   j <- i+1
   while (j <= numb) {  # for can run 6:5 ...
      k <- j + 1
      while (k <= numb) {
         triple[ind, ] <- c(i,j,k)
         ind <- ind + 1
         k <- k + 1
      }
      j <- j + 1
   }
}
# triple
matrix(heading[triple], n.comb.3, 3)
      [,1]            [,2]            [,3]
[1,] "Milk"          "Peanut.Butter" "Bread"
[2,] "Milk"          "Peanut.Butter" "Cereal"
[3,] "Milk"          "Peanut.Butter" "Jelly"
[4,] "Milk"          "Bread"         "Cereal"
[5,] "Milk"          "Bread"         "Jelly"
```

```
[6,] "Milk"           "Cereal"        "Jelly"
[7,] "Peanut.Butter" "Bread"         "Cereal"
[8,] "Peanut.Butter" "Bread"         "Jelly"
[9,] "Peanut.Butter" "Cereal"        "Jelly"
[10,] "Bread"         "Cereal"        "Jelly"
```

Count the number of times each triple occurs:

```
triple.counts <- matrix(0, n.comb.3, 4)
for (i in 1:dim(triple)[1]){
   triple.counts[i,] <-c(triple[i,],sum(floor(apply(baskets[,triple[i,]],1,sum)/3)))
}
matrix(c(heading[triple.counts[,1:3]],triple.counts[,4]), n.comb.3, 4)
      [,1]            [,2]            [,3]     [,4]
 [1,] "Milk"          "Peanut.Butter" "Bread"  "113"
 [2,] "Milk"          "Peanut.Butter" "Cereal"  "42"
 [3,] "Milk"          "Peanut.Butter" "Jelly"   "60"
 [4,] "Milk"          "Bread"         "Cereal" "154"
 [5,] "Milk"          "Bread"         "Jelly"  "217"
 [6,] "Milk"          "Cereal"        "Jelly"  "197"
 [7,] "Peanut.Butter" "Bread"         "Cereal"  "58"
 [8,] "Peanut.Butter" "Bread"         "Jelly"   "82"
 [9,] "Peanut.Butter" "Cereal"        "Jelly"   "55"
[10,] "Bread"         "Cereal"        "Jelly"  "231"
```

Count the number of times each quad occurs.

All unique quads

```
n.comb.4 <- choose(numb,4)
quad <- matrix(0, n.comb.4, 4)
ind <- 1
for (i in 1:numb) {
   j <- i+1
   while (j <= numb) {  # for can run 6:5 ...
      k <- j + 1
      while (k <= numb) {
         l <- k + 1
         while (l <= numb) {
            quad[ind, ] <- c(i,j,k,l)
            ind <- ind + 1
            l <- l + 1
          }
          k <- k + 1
      }
      j <- j + 1
   }
}
# quads
matrix(heading[quad], n.comb.4, 5)
quad.counts <- matrix(0, n.comb.4, 5)
for (i in 1:dim(quad)[1]){
```

```
   quad.counts[i,] <-c(quad[i,],sum(floor(apply(baskets[,quad[i,]],1,sum)/4)))
}
matrix(c(heading[quad.counts[,1:4]],quad.counts[,5]), n.comb.4, 5)
     [,1]            [,2]            [,3]     [,4]     [,5]
[1,] "Milk"          "Peanut Butter" "Bread"  "Cereal" "33"
[2,] "Milk"          "Peanut Butter" "Bread"  "Jelly"  "50"
[3,] "Milk"          "Peanut Butter" "Cereal" "Jelly"  "37"
[4,] "Milk"          "Bread"         "Cereal" "Jelly"  "102"
[5,] "Peanut Butter" "Bread"         "Cereal" "Jelly"  "46"
```

The hope of association rule mining analysis is that you may find that the presence of one product (say A) in a shopping basket infers, with high probability, that some other product (say C) will also be in the basket.

An **association rule** is a rule such as ”If a customer buys A and B, he/she often buys C as well”.

i.e. A and B $\Rightarrow$ C

The concept can be used to place products closer together to suggest to shoppers that these items that may be of interest. (At the same time, commonly purchased staples may be spread across the outside of the store to encourage impulse shopping by getting you to walk past many displays of other items you would not have been thinking about buying.) (It has been stated that Thomas Blischok (working for NCR at the time) spotted a correlation in the purchase of beer and diapers between 5pm and 7pm. The usual follow-up is that by moving beer and diapers closer together, sales boomed. *Forbes* magazine indicated that, while the former is true, the rearrangement did not happen.)

How do you determine if a rule is a good rule?

With all the possible products available, the potential combinations of itemsets that one could have in their ”basket” would be nearly impossible to inspect (we have looked at some simple combinations above). So how do we determine good rules? One algorithm for this is the *Apriori* algorithm (Rakesh Agrawal, Ramakrishnan Srikant, ”Fast Algorithms for Mining Association Rules” (1994),Proc. 20th Int. Conf. Very Large Data Bases, VLDB) which seeks to look at only the *most probable* sets and combine them.

Thus one concept that is used is the idea of **support** :

$$\begin{aligned}\text{Support}(A, B, C) &= \left(\frac{\text{Number of items containing } A, B, \text{ and } C}{\text{Total number of baskets}}\right) \times 100\% \\ &= P(A \cap B \cap C)\end{aligned}$$

Only those sets exceeding a specified level of support are retained as *candidate* sets for further consideration. this will reduce the number of combinations under consideration.

A second concept is the **confidence** of a *rule* of the form A and B ⇒ C:

$$\begin{aligned}\text{Confidence}(\mathit{rule}) &= \left(\frac{\text{Support}(A,B,C)}{\text{Support}(A,B)}\right) \times 100\% \\ &= P(C|A \cap B)\end{aligned}$$

By setting the confidence of a rule fairly high (and restricting ourselves to candidate sets $A \cap B$ that have sufficient support), we still want a rule that has high confidence of actually holding.

The third concept is that of **lift** of a *rule* of the form A and B ⇒ C:

$$\begin{aligned}\text{Lift}(\mathit{rule}) &= \left(\frac{\text{Confidence}(A,B,C)}{\text{Confidence}(C)}\right) \times 100\% \\ &= \frac{P(C|A \cap B)}{P(C)}\end{aligned}$$

So after restricting ourselves to things that occur often enough (support) and looking at how confident we are that the rule holds (confidence), we are interested in those rules that really result in significant ”improvement” (lift).
We wish to determine if the purchase of some items influences the purchase of others, and if so, which ones influence the most.

We start by looking at the **support for single items**. This is simply the percentages in the baskets (as seen before):

```
        Milk Peanut.Butter         Bread         Cereal         Jelly
        34.10         29.27         46.90         30.55         32.93
```

Now we look at **support for pairs**:

```
for (i in 1:length(item.in.basket)){    # Run through all the singles
   for (j in 1:nrow(triple)){      # For each single, look at each double
      one.in <- i == double[j,]
      if (any(one.in)) {               # Test to see if single in current double
         s.c <- item.in.basket[i]     # It is, so we get the appropriate counts.
         d.c <- double.counts[j, 3]
         item <- double[j, which(!one.in)]
         other <- heading[i]
         cat("When ", other,
               " is purchased (", s.c, "/", n.baskets, " = ",
               round(100*s.c/n.baskets, 2),"%) Support(", other, ")\n   ",
               heading[item], " was purchased (", d.c, "/", s.c,
               " = ", round(100*d.c/s.c, 2),"%)  Confidence\n", sep = "")
         cat("   Overall ", heading[item], " purchase rate is ",
percent.in.basket[item],
```

```
                  ”%)  Support(”, heading[item], ”),\n”, sep = ””)
         cat(”   Lift ”, round(10000*d.c/s.c, 0)/percent.in.basket[item], ”%\n”, sep =
””)
      }
   }
}
```

```
Milk is purchased (932/2733 = 34.1%) Support(Milk)
   Peanut.Butter was also purchased (191/932 = 20.49%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 70.00342%

When Milk is purchased (932/2733 = 34.1%) Support(Milk)
   Bread was also purchased (445/932 = 47.75%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 101.7907%

When Milk is purchased (932/2733 = 34.1%) Support(Milk)
   Cereal was also purchased (322/932 = 34.55%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 113.0933%

When Milk is purchased (932/2733 = 34.1%) Support(Milk)
   Jelly was also purchased (419/932 = 44.96%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 136.5320%


When Peanut.Butter is purchased (800/2733 = 29.27%) Support(Peanut.Butter)
   Milk was alsopurchased (191/800 = 23.88%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 70.02933%
When Peanut.Butter is purchased (800/2733 = 29.27%) Support(Peanut.Butter)
   Bread was also purchased (396/800 = 49.5%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 105.5212%
When Peanut.Butter is purchased (800/2733 = 29.27%) Support(Peanut.Butter)
   Cereal was also purchased (88/800 = 11%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 36.00655%
When Peanut.Butter is purchased (800/2733 = 29.27%) Support(Peanut.Butter)
   Jelly was also purchased (100/800 = 12.5%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 37.95931%


When Bread is purchased (1282/2733 = 46.91%) Support(Bread)
   Milk was also purchased (445/1282 = 34.71%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 101.7889%
When Bread is purchased (1282/2733 = 46.91%) Support(Bread)
   Peanut.Butter was also purchased (396/1282 = 30.89%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 105.5347%
When Bread is purchased (1282/2733 = 46.91%) Support(Bread)
   Cereal was also purchased (393/1282 = 30.66%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 100.3601%
When Bread is purchased (1282/2733 = 46.91%) Support(Bread)
   Jelly was also purchased (438/1282 = 34.17%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
```

```
   Lift 103.7656%


When Cereal is purchased (835/2733 = 30.55%) Support(Cereal)
   Milk was also purchased (322/835 = 38.56%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 113.0792%
When Cereal is purchased (835/2733 = 30.55%) Support(Cereal)
   Peanut.Butter was also purchased (88/835 = 10.54%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 36.00957%
When Cereal is purchased (835/2733 = 30.55%) Support(Cereal)
   Bread was also purchased (393/835 = 47.07%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 100.3411%
When Cereal is purchased (835/2733 = 30.55%) Support(Cereal)
   Jelly was also purchased (479/835 = 57.37%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 174.2180%


When Jelly is purchased (900/2733 = 32.93%) Support(Jelly)
   Milk was also purchased (419/900 = 46.56%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 136.5396%
When Jelly is purchased (900/2733 = 32.93%) Support(Jelly)
   Peanut.Butter was also purchased (100/900 = 11.11%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 37.95695%
When Jelly is purchased (900/2733 = 32.93%) Support(Jelly)
   Bread was also purchased (438/900 = 48.67%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 103.7519%
When Jelly is purchased (900/2733 = 32.93%) Support(Jelly)
   Cereal was also purchased (479/900 = 53.22%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 174.2062%
```

Now we look at **support for triples**:

```
for (i in 1:nrow(double)){    # Run through all the doubles
   for (j in 1:nrow(triple)){ # For each double, look at each triple
      one.in <- double[i,1] == triple[j,]
      two.in <- double[i,2] == triple[j,]
      if (sum(two.in*1 + one.in*1) == 2) { # Test to see if double in current triple
         d.c <- double.counts[i,3]     # It is, so we get the appropriate counts.
         t.c <- triple.counts[j,4]
         item <- triple[j,which(!(two.in|one.in))]
         others <- paste(heading[double[i,1]],heading[double[i,2]], sep = "/")
         cat("When ", others,
               " are purchased (", d.c, "/", n.baskets, " = ",
               round(100*d.c/n.baskets, 2),"%) Support(", others, ")\n   ",
               heading[item], " was purchased (", t.c,"/",d.c,
               " = ", round(100*t.c/d.c, 2),"%)  Confidence\n", sep = "")
         cat("   Overall ", heading[item], " purchase rate is ",
percent.in.basket[item],
                  "%)  Support(", heading[item], "),\n", sep = "")
         cat("   Lift ", round(10000*t.c/d.c, 0)/percent.in.basket[item], "%\n", sep =
"")
```

```
    }
  }
}
```

In the first pairing below, we consider purchases containing milk and peanut butter (i.e. Milk / Peanut.Butter where "/" indicates "and");'they were purchased together 6.99% of the time so the **support** is 6.99%. In combination with Milk / Peanut.Butter, we find that Bread was purchased 59.16% of the time but overall Bread only was purchased 46.91% of the time. This illustrates the **lift**. High values of lift indicate groupings of interest.

```
When Milk/Peanut.Butter are purchased (191/2733 = 6.99%) Support(Milk/Peanut.Butter)
   Bread was also purchased (113/191 = 59.16%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 126.1138%

When Milk/Peanut.Butter are purchased (191/2733 = 6.99%) Support(Milk/Peanut.Butter)
   Cereal was also purchased (42/191 = 21.99%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 71.98036%

When Milk/Peanut.Butter are purchased (191/2733 = 6.99%) Support(Milk/Peanut.Butter)
   Jelly was also purchased (60/191 = 31.41%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 95.38415%


When Milk/Bread are purchased (445/2733 = 16.28%) Support(Milk/Bread)
   Peanut.Butter was also purchased (113/445 = 25.39%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 86.7441%

When Milk/Bread are purchased (445/2733 = 16.28%) Support(Milk/Bread)
   Cereal was also purchased (154/445 = 34.61%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 113.2897%

When Milk/Bread are purchased (445/2733 = 16.28%) Support(Milk/Bread)
   Jelly was also purchased (217/445 = 48.76%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 148.0717%


When Milk/Cereal are purchased (322/2733 = 11.78%) Support(Milk/Cereal)
   Peanut.Butter was also purchased (42/322 = 13.04%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 44.55073%
When Milk/Cereal are purchased (322/2733 = 11.78%) Support(Milk/Cereal)
   Bread was also purchased (154/322 = 47.83%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 101.9612%
When Milk/Cereal are purchased (322/2733 = 11.78%) Support(Milk/Cereal)
   Jelly was also purchased (197/322 = 61.18%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 185.7880%


When Milk/Jelly are purchased (419/2733 = 15.33%) Support(Milk/Jelly)
   Peanut.Butter was also purchased (60/419 = 14.32%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 48.92381%
```

```
When Milk/Jelly are purchased (419/2733 = 15.33%) Support(Milk/Jelly)
   Bread was also purchased (217/419 = 51.79%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 110.4029%
When Milk/Jelly are purchased (419/2733 = 15.33%) Support(Milk/Jelly)
   Cereal was also purchased (197/419 = 47.02%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 153.9116%


When Peanut.Butter/Bread are purchased (396/2733 = 14.49%) Support(Peanut.Butter/Bread)
   Milk was also purchased (113/396 = 28.54%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 83.69501%
When Peanut.Butter/Bread are purchased (396/2733 = 14.49%) Support(Peanut.Butter/Bread)
   Cereal was also purchased (58/396 = 14.65%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 47.95417%
When Peanut.Butter/Bread are purchased (396/2733 = 14.49%) Support(Peanut.Butter/Bread)
   Jelly was also purchased (82/396 = 20.71%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 62.89098%


When Peanut.Butter/Cereal are purchased (88/2733 = 3.22%) Support(Peanut.Butter/Cereal)
   Milk was also purchased (42/88 = 47.73%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 139.9707%
When Peanut.Butter/Cereal are purchased (88/2733 = 3.22%) Support(Peanut.Butter/Cereal)
   Bread was also purchased (58/88 = 65.91%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 140.5031%
When Peanut.Butter/Cereal are purchased (88/2733 = 3.22%) Support(Peanut.Butter/Cereal)
   Jelly was also purchased (55/88 = 62.5%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 189.7965%


When Peanut.Butter/Jelly are purchased (100/2733 = 3.66%) Support(Peanut.Butter/Jelly)
   Milk was also purchased (60/100 = 60%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 175.9531%
When Peanut.Butter/Jelly are purchased (100/2733 = 3.66%) Support(Peanut.Butter/Jelly)
   Bread was also purchased (82/100 = 82%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 174.8028%
When Peanut.Butter/Jelly are purchased (100/2733 = 3.66%) Support(Peanut.Butter/Jelly)
   Cereal was also purchased (55/100 = 55%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 180.0327%


When Bread/Cereal are purchased (393/2733 = 14.38%) Support(Bread/Cereal)
   Milk was also purchased (154/393 = 39.19%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 114.9267%
When Bread/Cereal are purchased (393/2733 = 14.38%) Support(Bread/Cereal)
   Peanut.Butter was also purchased (58/393 = 14.76%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 50.42706%
When Bread/Cereal are purchased (393/2733 = 14.38%) Support(Bread/Cereal)
   Jelly was also purchased (231/393 = 58.78%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
```

```
   Lift 178.4998%


When Bread/Jelly are purchased (438/2733 = 16.03%) Support(Bread/Jelly)
   Milk was also purchased (217/438 = 49.54%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 145.2786%
When Bread/Jelly are purchased (438/2733 = 16.03%) Support(Bread/Jelly)
   Peanut.Butter was also purchased (82/438 = 18.72%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 63.95627%
When Bread/Jelly are purchased (438/2733 = 16.03%) Support(Bread/Jelly)
   Cereal was also purchased (231/438 = 52.74%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 172.6350%


When Cereal/Jelly are purchased (479/2733 = 17.53%) Support(Cereal/Jelly)
   Milk was also purchased (197/479 = 41.13%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 120.6158%
When Cereal/Jelly are purchased (479/2733 = 17.53%) Support(Cereal/Jelly)
   Peanut.Butter was also purchased (55/479 = 11.48%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 39.22105%
When Cereal/Jelly are purchased (479/2733 = 17.53%) Support(Cereal/Jelly)
   Bread was also purchased (231/479 = 48.23%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 102.8139%
```

```
for (i in 1:nrow(triple)){    # Run through all the triples
   for (j in 1:nrow(quad)){ # For each triple, look at each quad
      one.in <- triple[i, 1] == quad[j,]
      two.in <- triple[i, 2] == quad[j,]
      three.in <- triple[i, 3] == quad[j,]
      if (sum(two.in*1 + one.in*1 + three.in*1) == 3) { # Test to see if triple in
current quad
         t.c <- triple.counts[i, 4]     # It is, so we get the appropriate counts.
         q.c <- quad.counts[j, 5]
         item <- quad[j, which(!(three.in|two.in|one.in))]
         others <- paste(heading[triple[i,]], collapse = "/")
         cat("When ", others,
               " are purchased (", t.c, "/", n.baskets, " = ",
               round(100*t.c/n.baskets, 2),"%) Support(", others, ")\n   ",
               heading[item], " was purchased (", q.c,"/", t.c,
               " = ", round(100*q.c/t.c, 2),"%)  Confidence\n", sep = "")
         cat("   Overall ", heading[item], " purchase rate is ",
percent.in.basket[item],
                  "%)  Support(", heading[item], "),\n", sep = "")
         cat("   Lift ", round(10000*q.c/t.c, 0)/percent.in.basket[item], "%\n", sep =
"")
      }
   }
}
When Milk/Peanut.Butter/Bread are purchased (113/2733 = 4.13%) Support(Milk/Peanut.Butter
   Cereal was also purchased (33/113 = 29.2%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 95.58101%
```

```
When Milk/Peanut.Butter/Bread are purchased (113/2733 = 4.13%) Support(Milk/Peanut.Butter
   Jelly was also purchased (50/113 = 44.25%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 134.3759%


When Milk/Peanut.Butter/Cereal are purchased (42/2733 = 1.54%) Support(Milk/Peanut.Butter
   Bread was also purchased (33/42 = 78.57%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 167.4909%
When Milk/Peanut.Butter/Cereal are purchased (42/2733 = 1.54%) Support(Milk/Peanut.Butter
   Jelly was also purchased (37/42 = 88.1%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 267.5372%


When Milk/Peanut.Butter/Jelly are purchased (60/2733 = 2.2%) Support(Milk/Peanut.Butter/
   Bread was also purchased (50/60 = 83.33%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 177.6380%
When Milk/Peanut.Butter/Jelly are purchased (60/2733 = 2.2%) Support(Milk/Peanut.Butter/
   Cereal was also purchased (37/60 = 61.67%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 201.8658%


When Milk/Bread/Cereal are purchased (154/2733 = 5.63%) Support(Milk/Bread/Cereal)
   Peanut.Butter was also purchased (33/154 = 21.43%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 73.2149%
When Milk/Bread/Cereal are purchased (154/2733 = 5.63%) Support(Milk/Bread/Cereal)
   Jelly was also purchased (102/154 = 66.23%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
   Lift 201.1236%


When Milk/Bread/Jelly are purchased (217/2733 = 7.94%) Support(Milk/Bread/Jelly)
   Peanut.Butter was also purchased (50/217 = 23.04%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 78.71541%
When Milk/Bread/Jelly are purchased (217/2733 = 7.94%) Support(Milk/Bread/Jelly)
   Cereal was also purchased (102/217 = 47%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 153.8462%


When Milk/Cereal/Jelly are purchased (197/2733 = 7.21%) Support(Milk/Cereal/Jelly)
   Peanut.Butter was also purchased (37/197 = 18.78%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 64.16126%
When Milk/Cereal/Jelly are purchased (197/2733 = 7.21%) Support(Milk/Cereal/Jelly)
   Bread was also purchased (102/197 = 51.78%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 110.3816%


When Peanut.Butter/Bread/Cereal are purchased (58/2733 = 2.12%) Support(Peanut.Butter/Bread
   Milk was also purchased (33/58 = 56.9%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 166.8622%
When Peanut.Butter/Bread/Cereal are purchased (58/2733 = 2.12%) Support(Peanut.Butter/Bread
   Jelly was also purchased (46/58 = 79.31%)  Confidence
   Overall Jelly purchase rate is 32.93%)  Support(Jelly),
```

```
   Lift 240.8442%


When Peanut.Butter/Bread/Jelly are purchased (82/2733 = 3%) Support(Peanut.Butter/Bread/
   Milk was also purchased (50/82 = 60.98%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 178.8270%
When Peanut.Butter/Bread/Jelly are purchased (82/2733 = 3%) Support(Peanut.Butter/Bread/
   Cereal was also purchased (46/82 = 56.1%)  Confidence
   Overall Cereal purchase rate is 30.55%)  Support(Cereal),
   Lift 183.6334%


When Peanut.Butter/Cereal/Jelly are purchased (55/2733 = 2.01%) Support(Peanut.Butter/Cerea
   Milk was also purchased (37/55 = 67.27%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 197.2727%
When Peanut.Butter/Cereal/Jelly are purchased (55/2733 = 2.01%) Support(Peanut.Butter/Cerea
   Bread was also purchased (46/55 = 83.64%)  Confidence
   Overall Bread purchase rate is 46.91%)  Support(Bread),
   Lift 178.2989%


When Bread/Cereal/Jelly are purchased (231/2733 = 8.45%) Support(Bread/Cereal/Jelly)
   Milk was also purchased (102/231 = 44.16%)  Confidence
   Overall Milk purchase rate is 34.1%)  Support(Milk),
   Lift 129.5015%
When Bread/Cereal/Jelly are purchased (231/2733 = 8.45%) Support(Bread/Cereal/Jelly)
   Peanut.Butter was also purchased (46/231 = 19.91%)  Confidence
   Overall Peanut.Butter purchase rate is 29.27%)  Support(Peanut.Butter),
   Lift 68.02187%
```
etc.

The basic idea of this type of analysis is to determine if knowledge of some item(s) purchased will help us predict what else may also be purchased. If we can discover this knowledge (i.e. if there is an increased probability of purchasing certain items when other particular items are purchased) it can lead to a better positioning of such items in a physical store (perhaps to generate an “improved” traffic flow) or to increase sales revenue.

A major problem with the “brute force” method used above is that the number of itemsets grows very quickly. We can *prune* ones that have low support. A paper (R. Agrawal, T. Imielinski, and A. Swami (1993) ”Mining association rules between sets of items in large databases”, in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207-216, Washington, D.C.) looked at a method for keeping the size (and hence the computing time) down.

Suppose that we look back at the preceding output and extract the support for the various groupings and decide that we require a **minimum support of 5**% . (We would actually want support in the ≥80% region, but our simulated data has no sets at that level).

Singles	Support	Pairs	Support	Triples	Support
Bread	46.90%	Cereal/Jelly	17.53%	Bread/Cereal/Jelly	8.45%
Milk	34.10%	Milk/Bread	16.28%	Milk/Bread/Jelly	7.94%
Jelly	32.93%	Bread/Jelly	16.03%	Milk/Cereal/Jelly	7.21%
Cereal	30.55%	Milk/Jelly	15.33%	Milk/Bread/Cereal	5.63%
Peanut.Butter	29.27%	Peanut.Butter/Bread	14.49%	-------------------------	
		Bread/Cereal	14.38%	Milk/Peanut.Butter/Bread	4.13%
		Milk/Cereal	11.78%	Peanut.Butter/Bread/Jelly	3.00%
		Milk/Peanut.Butter	6.99%	Milk/Peanut.Butter/Jelly	2.20%
		----------------------		Peanut.Butter/Bread/Cereal	2.12%
		Peanut.Butter/Cereal	3.22%	Peanut.Butter/Cereal/Jelly	2.01%
		Peanut.Butter/Jelly	3.66%	Milk/Peanut.Butter/Cereal	1.54%

In the pairs, we see that the Peanut.Butter/Cereal and Peanut.Butter/Jelly fall below our threshold of 5%. When we look at the triples, we see that anything with those combinations falls below the threshold This suggests that we do not need to use all the pairs in the creation of the triples - we could **prune** to the top 8 pairs in the pairs table above.. Then for the triples (note -the last 5 triples would not now be generated). we would further prune Milk/Peanut.Butter/Bread. In this way, we would have to generate fewer sets of size 4, 5, ... and might be able to keep the complexity down. This is the basic idea behind the *Apriori* algorithmn described by Agrawal et al. (see below in these notes.)

(It should be noted that the sets were not produced in the way suggested by the above.) We do not want to generate triples based on pairs that should be pruned. Once we have the pairs that we have *not pruned*, we would use *those* pairs as the basis for generating the triples. In this way we eliminate the issue of throwing out cases that would reintroduce combinations that have previously been eliminated.

There is a package in R that does **association rules** (**Note that before using this package you will have to load it**. To do so,

```
install.packages ("arules")
library(arules)
```

The `apriori` function requires that the **columns of our dataset be factors**. (Note that we do not want to prune here because we want to compare the output of this routine to our previous output, so we set the support and confidence thresholds to a very low level.)

```
rules <- apriori(apply(baskets, 2, as.numeric),
                    parameter = list(supp = 0.01, conf = 0.01, target = "rules"))
parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
       0.01    0.1    1 none FALSE            TRUE    0.01      1     10  rules FALSE
algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[5 item(s), 2733 transaction(s)] done [0.00s].
```

```
sorting and recoding items ... [5 item(s)] done [0.00s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [80 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

summary(rules)
set of 80 rules
rule length distribution (lhs + rhs):
 1  2  3  4  5
 5 20 30 20  5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       2       3       3       4       5
summary of quality measures:
    support           confidence          lift
 Min.   :0.01134   Min.   :0.1054   Min.   :0.3600
 1st Qu.:0.01793   1st Qu.:0.2904   1st Qu.:0.9323
 Median :0.03696   Median :0.4673   Median :1.1174
 Mean   :0.07384   Mean   :0.4386   Mean   :1.2458
 3rd Qu.:0.08452   3rd Qu.:0.5772   3rd Qu.:1.7420
 Max.   :0.46908   Max.   :0.9394   Max.   :2.8526
```

Recall (again) the `percent.in.basket`

```
        Milk Peanut.Butter        Bread        Cereal        Jelly
        34.10         29.27        46.90         30.55        32.93
```

and we see these values as the **support**, **confidence** and **lift** values in rules 1 through 5.

Rule 6. Support(P,C) = P(P∩C)= (Baskets with Peanut.Butter & Cereal)/Baskets

= 88/2733 = 0.032199

Support(P) = P(P) = (Baskets with Peanut.Butter)/Baskets

= 800/2733 = 0.29272

Confidence = P(C|P) = Support(P,C)/Support(P)

= 0.032199/0.29272 = 0.11

Lift = P(C|P)/P(C) = 0.11/(835/2733) = 0.36004

Rule 26. Support(P,C,M) = P(P∩C∩M)

= (Baskets with Peanut.Butter, Cereal, & Milk)/Baskets

= 42/2733 = 0.015368

Support(P,C) = P(P∩C) = 0.032199 (see above)

Confidence = P(M|P∩C) = Support(P,C,M)/Support(P,C)

= 0.015368/0.032199 = 0.47728

Lift = P(M|P∩C)/P(M) = 0.47728/(932/2733) = 1.3996

The other rules are evaluated in a similar way

```
inspect(rules)
   lhs                  rhs             support confidence lift
1  {}                => {Peanut.Butter} 0.29271862  0.2927186 1.0000000
```

```
2   {}                 => {Cereal}        0.30552506  0.3055251 1.0000000
3   {}                 => {Milk}          0.34101720  0.3410172 1.0000000
4   {}                 => {Jelly}         0.32930845  0.3293085 1.0000000
5   {}                 => {Bread}         0.46908160  0.4690816 1.0000000
6   {Peanut.Butter}    => {Cereal}        0.03219905  0.1100000 0.3600359
7   {Cereal}           => {Peanut.Butter} 0.03219905  0.1053892 0.3600359
8   {Peanut.Butter}    => {Milk}          0.06988657  0.2387500 0.7001113
9   {Milk}             => {Peanut.Butter} 0.06988657  0.2049356 0.7001113
10 {Peanut.Butter}     => {Jelly}         0.03658983  0.1250000 0.3795833
11 {Jelly}             => {Peanut.Butter} 0.03658983  0.1111111 0.3795833
12 {Peanut.Butter}     => {Bread}         0.14489572  0.4950000 1.0552535
13 {Bread}             => {Peanut.Butter} 0.14489572  0.3088924 1.0552535
14 {Cereal}            => {Milk}          0.11781925  0.3856287 1.1308190
15 {Milk}              => {Cereal}        0.11781925  0.3454936 1.1308190
16 {Cereal}            => {Jelly}         0.17526528  0.5736527 1.7419920
17 {Jelly}             => {Cereal}        0.17526528  0.5322222 1.7419920
18 {Cereal}            => {Bread}         0.14379802  0.4706587 1.0033621
19 {Bread}             => {Cereal}        0.14379802  0.3065523 1.0033621
20 {Milk}              => {Jelly}         0.15331138  0.4495708 1.3651967
21 {Jelly}             => {Milk}          0.15331138  0.4655556 1.3651967
22 {Milk}              => {Bread}         0.16282473  0.4774678 1.0178779
23 {Bread}             => {Milk}          0.16282473  0.3471139 1.0178779
24 {Jelly}             => {Bread}         0.16026345  0.4866667 1.0374883
25 {Bread}             => {Jelly}         0.16026345  0.3416537 1.0374883
26 {Peanut.Butter,
    Cereal}            => {Milk}          0.01536773  0.4772727 1.3995562
27 {Milk,
    Peanut.Butter}     => {Cereal}        0.01536773  0.2198953 0.7197291
28 {Milk,
    Cereal}            => {Peanut.Butter} 0.01536773  0.1304348 0.4455978
29 {Peanut.Butter,
    Cereal}            => {Jelly}         0.02012441  0.6250000 1.8979167
30 {Peanut.Butter,
    Jelly}             => {Cereal}        0.02012441  0.5500000 1.8001796
31 {Cereal,
    Jelly}             => {Peanut.Butter} 0.02012441  0.1148225 0.3922625
32 {Peanut.Butter,
    Cereal}            => {Bread}         0.02122210  0.6590909 1.4050667
33 {Peanut.Butter,
    Bread}             => {Cereal}        0.02122210  0.1464646 0.4793867
34 {Bread,
    Cereal}            => {Peanut.Butter} 0.02122210  0.1475827 0.5041794
35 {Milk,
    Peanut.Butter}     => {Jelly}         0.02195390  0.3141361 0.9539267
36 {Peanut.Butter,
    Jelly}             => {Milk}          0.02195390  0.6000000 1.7594421
37 {Milk,
    Jelly}             => {Peanut.Butter} 0.02195390  0.1431981 0.4892005
38 {Milk,
    Peanut.Butter}     => {Bread}         0.04134651  0.5916230 1.2612369
39 {Peanut.Butter,
    Bread}             => {Milk}          0.04134651  0.2853535 0.8367717
40 {Milk,
    Bread}             => {Peanut.Butter} 0.04134651  0.2539326 0.8674972
41 {Peanut.Butter,
    Jelly}             => {Bread}         0.03000366  0.8200000 1.7480967
42 {Peanut.Butter,
    Bread}             => {Jelly}         0.03000366  0.2070707 0.6288047
43 {Bread,
    Jelly}             => {Peanut.Butter} 0.03000366  0.1872146 0.6395719
44 {Milk,
    Cereal}            => {Jelly}         0.07208196  0.6118012 1.8578364
45 {Cereal,
```

```
     Jelly}          => {Milk}          0.07208196  0.4112735 1.2060198
46 {Milk,
     Jelly}          => {Cereal}        0.07208196  0.4701671 1.5388821
47 {Milk,
     Cereal}         => {Bread}         0.05634834  0.4782609 1.0195686
48 {Bread,
     Cereal}         => {Milk}          0.05634834  0.3918575 1.1490843
49 {Milk,
     Bread}          => {Cereal}        0.05634834  0.3460674 1.1326973
50 {Cereal,
     Jelly}          => {Bread}         0.08452250  0.4822547 1.0280828
51 {Bread,
     Cereal}         => {Jelly}         0.08452250  0.5877863 1.7849109
52 {Bread,
     Jelly}          => {Cereal}        0.08452250  0.5273973 1.7261997
53 {Milk,
     Jelly}          => {Bread}         0.07939993  0.5178998 1.1040718
54 {Milk,
     Bread}          => {Jelly}         0.07939993  0.4876404 1.4808015
55 {Bread,
     Jelly}          => {Milk}          0.07939993  0.4954338 1.4528117
56 {Milk,
     Peanut.Butter,
     Cereal}         => {Jelly}         0.01353824  0.8809524 2.6751587
57 {Peanut.Butter,
     Cereal,
     Jelly}          => {Milk}          0.01353824  0.6727273 1.9727078
58 {Milk,
     Peanut.Butter,
     Jelly}          => {Cereal}        0.01353824  0.6166667 2.0183832
59 {Milk,
     Cereal,
     Jelly}          => {Peanut.Butter} 0.01353824  0.1878173 0.6416307
60 {Milk,
     Peanut.Butter,
     Cereal}         => {Bread}         0.01207464  0.7857143 1.6750056
61 {Peanut.Butter,
     Bread,
     Cereal}         => {Milk}          0.01207464  0.5689655 1.6684364
62 {Milk,
     Peanut.Butter,
     Bread}          => {Cereal}        0.01207464  0.2920354 0.9558476
63 {Milk,
     Bread,
     Cereal}         => {Peanut.Butter} 0.01207464  0.2142857 0.7320536
64 {Peanut.Butter,
     Cereal,
     Jelly}          => {Bread}         0.01683132  0.8363636 1.7829811
65 {Peanut.Butter,
     Bread,
     Cereal}         => {Jelly}         0.01683132  0.7931034 2.4083908
66 {Peanut.Butter,
     Bread,
     Jelly}          => {Cereal}        0.01683132  0.5609756 1.8361034
67 {Bread,
     Cereal,
     Jelly}          => {Peanut.Butter} 0.01683132  0.1991342 0.6802922
68 {Milk,
     Peanut.Butter,
     Jelly}          => {Bread}         0.01829491  0.8333333 1.7765211
69 {Milk,
     Peanut.Butter,
     Bread}          => {Jelly}         0.01829491  0.4424779 1.3436578
```

```
70 {Peanut.Butter,
     Bread,
     Jelly}          => {Milk}          0.01829491  0.6097561 1.7880509
71 {Milk,
     Bread,
     Jelly}          => {Peanut.Butter} 0.01829491  0.2304147 0.7871544
72 {Milk,
     Cereal,
     Jelly}          => {Bread}         0.03732162  0.5177665 1.1037877
73 {Milk,
     Bread,
     Cereal}         => {Jelly}         0.03732162  0.6623377 2.0112987
74 {Bread,
     Cereal,
     Jelly}          => {Milk}          0.03732162  0.4415584 1.2948275
75 {Milk,
     Bread,
     Jelly}          => {Cereal}        0.03732162  0.4700461 1.5384862
76 {Milk,
     Peanut.Butter,
     Cereal,
     Jelly}          => {Bread}         0.01134285  0.8378378 1.7861239
77 {Milk,
     Peanut.Butter,
     Bread,
     Cereal}         => {Jelly}         0.01134285  0.9393939 2.8526263
78 {Peanut.Butter,
     Bread,
     Cereal,
     Jelly}          => {Milk}          0.01134285  0.6739130 1.9761849
79 {Milk,
     Peanut.Butter,
     Bread,
     Jelly}          => {Cereal}        0.01134285  0.6200000 2.0292934
80 {Milk,
     Bread,
     Cereal,
     Jelly}          => {Peanut.Butter} 0.01134285  0.3039216 1.0382721
```

Note: Stand-alone programs for frequent pattern mining are provided by Christian Borgeit at http://www.borgelt.net/software.html

Recall

```
(item.in.basket <- apply (baskets, 2, sum))
        Milk Peanut.Butter      Bread        Cereal       Jelly
          932           800       1282          835          900

(percent.in.basket <- floor(apply (baskets, 2, sum)/.3)/100)
        Milk Peanut.Butter     Bread        Cereal       Jelly
        34.10         29.27      46.90        30.55        32.93
```

We see 30.6%/835 (=the percentage of baskets with cereal)/(number of baskets with cereal)) so the **first number** is the **support** for cereal.

```
Cereal <-  (30.6%/835, 100.0%/2733, 30.6%, 100.0%)
Peanut_Butter <-  (29.3%/800, 100.0%/2733, 29.3%, 100.0%)
Milk <-  (34.1%/932, 100.0%/2733, 34.1%, 100.0%)
Jelly <-  (32.9%/900, 100.0%/2733, 32.9%, 100.0%)
Bread <-  (46.9%/1282, 100.0%/2733, 46.9%, 100.0%)
```

```
Milk <- Peanut_Butter  (7.0%/191, 29.3%/800, 23.9%, 70.0%)
```

From the pairs:

```
 [6,] "Peanut.Butter" "Cereal"         "88"
[10,] "Cereal"        "Jelly"         "479"
```

so (see below)

- 17.5% is the **support** of Cereal and Jelly and 479 is the number of baskets in which that combination appears;

- 30.6% is the **support** for Cereal alone (as above) and 835 is the **number of times** cereal occurs;

so P(C,J)/P(C) = 479/835 = 57.4% = the **confidence** of Cereal ⇒Jelly;

and{P(C,J)/P(C)}/P(J) = {(479/2733)/(835/2733)}/(900/2733) = 174.2% is the **lift** of Cereal ⇒Jelly.

```
Jelly <- Cereal  (17.5%/479, 30.6%/835, 57.4%, 174.2%)
Cereal <- Jelly  (17.5%/479, 32.9%/900, 53.2%, 174.2%)
Milk <- Cereal  (11.8%/322, 30.6%/835, 38.6%, 113.1%)
Cereal <- Milk  (11.8%/322, 34.1%/932, 34.5%, 113.1%)
Peanut_Butter <- Milk  (7.0%/191, 34.1%/932, 20.5%, 70.0%)
Bread <- Peanut_Butter  (14.5%/396, 29.3%/800, 49.5%, 105.5%)
Peanut_Butter <- Bread  (14.5%/396, 46.9%/1282, 30.9%, 105.5%)
Bread <- Cereal  (14.4%/393, 30.6%/835, 47.1%, 100.3%)
Cereal <- Bread  (14.4%/393, 46.9%/1282, 30.7%, 100.3%)
Jelly <- Milk  (15.3%/419, 34.1%/932, 45.0%, 136.5%)
Milk <- Jelly  (15.3%/419, 32.9%/900, 46.6%, 136.5%)
Bread <- Milk  (16.3%/445, 34.1%/932, 47.7%, 101.8%)
Milk <- Bread  (16.3%/445, 46.9%/1282, 34.7%, 101.8%)
Bread <- Jelly  (16.0%/438, 32.9%/900, 48.7%, 103.7%)
Jelly <- Bread  (16.0%/438, 46.9%/1282, 34.2%, 103.7%)
```

From the triples -

```
 [9,] "Peanut.Butter" "Cereal"        "Jelly"   "55"
```

so (see below)

- 2.0% is the **support** of Cereal, Peanut.Butter and Jelly and 55 is the number of baskets in which that combination appears;
- 3.2% is the **support** and 88 is the **count** for Cereal and Peanut.Butter together in baskets(see above [6,]) - note that this does not appear in the output above because the minimum confidence was set at 20%;

so P(C,J,PB)/P(C,PB) = 55/88 = 62.5% is the **confidence** of Peanut.Butter and Cereal⇒Jelly

and P(C,J,PB)/P(C,PB)}/P(J) = {(55/2733)/(88/2733}/(900/2733) = 189.8% is the **lift** Peanut.Butter and Cereal⇒Jelly.

```
Jelly <- Peanut_Butter Cereal  (2.0%/55, 3.2%/88, 62.5%, 189.8%)
Cereal <- Peanut_Butter Jelly  (2.0%/55, 3.7%/100, 55.0%, 180.0%)
Milk <- Peanut_Butter Cereal  (1.5%/42, 3.2%/88, 47.7%, 140.0%)
Cereal <- Peanut_Butter Milk  (1.5%/42, 7.0%/191, 22.0%, 72.0%)
Bread <- Peanut_Butter Cereal  (2.1%/58, 3.2%/88, 65.9%, 140.5%)
Jelly <- Peanut_Butter Milk  (2.2%/60, 7.0%/191, 31.4%, 95.4%)
Milk <- Peanut_Butter Jelly  (2.2%/60, 3.7%/100, 60.0%, 175.9%)
Bread <- Peanut_Butter Milk  (4.1%/113, 7.0%/191, 59.2%, 126.1%)
Milk <- Peanut_Butter Bread  (4.1%/113, 14.5%/396, 28.5%, 83.7%)
Peanut_Butter <- Milk Bread  (4.1%/113, 16.3%/445, 25.4%, 86.7%)
```

```
Bread <- Peanut_Butter Jelly  (3.0%/82, 3.7%/100, 82.0%, 174.8%)
Jelly <- Peanut_Butter Bread  (3.0%/82, 14.5%/396, 20.7%, 62.9%)
Jelly <- Cereal Milk  (7.2%/197, 11.8%/322, 61.2%, 185.8%)
Milk <- Cereal Jelly  (7.2%/197, 17.5%/479, 41.1%, 120.6%)
Cereal <- Milk Jelly  (7.2%/197, 15.3%/419, 47.0%, 153.9%)
Bread <- Cereal Milk  (5.6%/154, 11.8%/322, 47.8%, 102.0%)
Milk <- Cereal Bread  (5.6%/154, 14.4%/393, 39.2%, 114.9%)
Cereal <- Milk Bread  (5.6%/154, 16.3%/445, 34.6%, 113.3%)
Bread <- Cereal Jelly  (8.5%/231, 17.5%/479, 48.2%, 102.8%)
Jelly <- Cereal Bread  (8.5%/231, 14.4%/393, 58.8%, 178.5%)
Cereal <- Jelly Bread  (8.5%/231, 16.0%/438, 52.7%, 172.6%)
Bread <- Milk Jelly  (7.9%/217, 15.3%/419, 51.8%, 110.4%)
Jelly <- Milk Bread  (7.9%/217, 16.3%/445, 48.8%, 148.1%)
Milk <- Jelly Bread  (7.9%/217, 16.0%/438, 49.5%, 145.3%)
Jelly <- Peanut_Butter Cereal Milk  (1.4%/37, 1.5%/42, 88.1%, 267.5%)
Milk <- Peanut_Butter Cereal Jelly  (1.4%/37, 2.0%/55, 67.3%, 197.3%)
Cereal <- Peanut_Butter Milk Jelly  (1.4%/37, 2.2%/60, 61.7%, 201.8%)
Bread <- Peanut_Butter Cereal Milk  (1.2%/33, 1.5%/42, 78.6%, 167.5%)
Milk <- Peanut_Butter Cereal Bread  (1.2%/33, 2.1%/58, 56.9%, 166.8%)
Cereal <- Peanut_Butter Milk Bread  (1.2%/33, 4.1%/113, 29.2%, 95.6%)
Peanut_Butter <- Cereal Milk Bread  (1.2%/33, 5.6%/154, 21.4%, 73.2%)
Bread <- Peanut_Butter Cereal Jelly  (1.7%/46, 2.0%/55, 83.6%, 178.3%)
Jelly <- Peanut_Butter Cereal Bread  (1.7%/46, 2.1%/58, 79.3%, 240.8%)
Cereal <- Peanut_Butter Jelly Bread  (1.7%/46, 3.0%/82, 56.1%, 183.6%)
Bread <- Peanut_Butter Milk Jelly  (1.8%/50, 2.2%/60, 83.3%, 177.7%)
Jelly <- Peanut_Butter Milk Bread  (1.8%/50, 4.1%/113, 44.2%, 134.4%)
Milk <- Peanut_Butter Jelly Bread  (1.8%/50, 3.0%/82, 61.0%, 178.8%)
Peanut_Butter <- Milk Jelly Bread  (1.8%/50, 7.9%/217, 23.0%, 78.7%)
Bread <- Cereal Milk Jelly  (3.7%/102, 7.2%/197, 51.8%, 110.4%)
Jelly <- Cereal Milk Bread  (3.7%/102, 5.6%/154, 66.2%, 201.1%)
Milk <- Cereal Jelly Bread  (3.7%/102, 8.5%/231, 44.2%, 129.5%)
Cereal <- Milk Jelly Bread  (3.7%/102, 7.9%/217, 47.0%, 153.8%)
Bread <- Peanut_Butter Cereal Milk Jelly  (1.1%/31, 1.4%/37, 83.8%, 178.6%)
Jelly <- Peanut_Butter Cereal Milk Bread  (1.1%/31, 1.2%/33, 93.9%, 285.3%)
Milk <- Peanut_Butter Cereal Jelly Bread  (1.1%/31, 1.7%/46, 67.4%, 197.6%)
Cereal <- Peanut_Butter Milk Jelly Bread  (1.1%/31, 1.8%/50, 62.0%, 202.9%)
Peanut_Butter <- Cereal Milk Jelly Bread  (1.1%/31, 3.7%/102, 30.4%, 103.8%)
```