

執行前說明

本作業使用 R 語言 (3.6.3) 進行資料前處理。

In [48]:

```
library(RODBC)
library(RWeka)
library(arules)
library(knitr)

db <- odbcDriverConnect("Driver={Microsoft Access Driver (*.mdb, *.accdb)};
                        DBQ=.\foodmart2000.mdb")
```

題目 1

請利用Weka 中的Apriori 演算法，從Foodmart資料庫的交易資料中，探勘符合Minimum Support = 0.0001且Minimum Confidence = 0.9的Association Rules，並列出Confidence最高的前10條Rules。若無法跑出結果，請簡述其原因。

資料前處理

In [39]:

```
sql <- "
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_1998 as ft, product as pd, product_class as pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id
union all
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_dec_1998 as ft, product as pd, product_class as pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id;"

ft <- as.data.frame(sqlQuery(db, sql))
ft[, 1] <- as.character(ft[, 1])
ft[, 2] <- as.character(ft[, 2])

head(ft)
```

In []:

```
tids <- unique(ft$tid)
print(paste('tid length:', length(tids)))

max_len <- 0
for (tid in tids){
  items <- unique(ft[ft$tid==tid, 2])
  item_len <- length(items)
  max_len <- ifelse(item_len > max_len, item_len, max_len)
}
print(paste('max_len:', max_len))

header <- paste('I', 1:max_len, sep = '')
result_frame <- rbind(NULL)

for (tid in tids){
  items <- unique(ft[ft$tid==tid, 2])
  items_len <- length(items)
  diff_len <- max_len - items_len
  row <- c(items, rep('', diff_len))
  result_frame <- rbind(result_frame, row)
}

result_frame <- rbind(header, result_frame)
head(result_frame)

write.arff(result_frame, file = 'q1_apriori.arff')
write.table(
  result_frame,
  'q1_apriori.csv',
  row.names = FALSE,
  col.names = FALSE,
  sep = ",",
  quote = TRUE)
```

Weka 結果

根據上述程式做出之資料集進行 mining 後，得到以下十條 rule:

Idx	LHS	RHS	Conf	Lift
1.	Magazines ,Candy ,Dairy	Snack Foods	1	9.85
2.	Magazines ,Dairy ,Snack Foods	Bread	1	35.05
3.	Magazines ,Bread ,Snack Foods	Dairy	1	20.95
4.	Snack Foods ,Beer and Wine ,Meat	Vegetables	1	8.26
5.	Jams and Jellies ,Snack Foods ,Hot Beverages	Dairy	1	20.95
6.	Dairy ,Snack Foods ,Hygiene	Vegetables	1	8.26
7.	Vegetables ,Dairy ,Hygiene	Snack Foods	1	17.8
8.	Beer and Wine ,Vegetables ,Fruit	Snack Foods	1	8.87
9.	Starchy Foods ,Breakfast Foods ,Magazines	Snack Foods	1	9.85
10.	Starchy Foods ,Snack Foods ,Magazines	Breakfast Foods	1	38.82

Weka 原始輸出

以下是 Weka 輸出結果：

=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 1.0
E-4 -U 1.0 -M 1.0E-4 -S -1.0 -c -1
Relation: q1_apriori
Instances: 37851
Attributes: 18
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
I16
I17
I18

=== Associator model (full training set) ===

Apriori

=====

Minimum support: 0 (8 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 9998

Generated sets of large itemsets:

Size of set of large itemsets L(1): 448

Size of set of large itemsets L(2): 11004

Size of set of large itemsets L(3): 2403

Size of set of large itemsets L(4): 28

Best rules found:

題目 2

請利用Weka 中的FP-Growth演算法，從Foodmart資料庫的交易資料中，探勘符合Minimum Support = 0.0001 且Minimum Confidence = 0.9的Association Rules，並列出Confidence最高的前10條Rules。若無法跑出結果，請簡述其原因。

資料前處理

In []:

```
sql2 <- "
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_1998 as ft, product as pd, product_class as pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id
union all
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_dec_1998 as ft, product as pd, product_class as pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id;"

ft <- as.data.frame(sqlQuery(db, sql2))
ft[, 1] <- as.character(ft[, 1])
ft[, 2] <- as.character(ft[, 2])

head(ft)
```

In []:

```
sql_pc <- "select distinct product_category from product_class; "

pc <- as.data.frame(sqlQuery(db, sql_pc))
pc[,1] <- as.character(pc[,1])

columns <- c("tid", as.vector(t(pc)[1,]))
len <- length(columns)
tids <- unique(ft$tid)
df <- as.data.frame(matrix(ncol = length(columns),
                           nrow=0,
                           dimnames = list(NULL,columns)))

ridx <- 1

for(tid in tids){
  df[ridx, 1] = tid
  df[ridx, 2:len] = 0

  pcs <- unique(as.vector(ft[ft$tid==tid,2]))

  for(pc in pcs){
    df[ridx, pc] = 1
  }

  ridx <- ridx + 1
}

for (idx in 1:length(columns)){
  df[,idx] <- as.factor(df[,idx])
}

head(df)

write.arff(df[,2:len], file = 'q2_fpgrowth.arff')
write.table(
  df[,2:len],
  'q2_fpgrowth.csv',
  row.names = FALSE,
  col.names = TRUE,
  sep = ",",
  quote = FALSE)
```

Weka 結果

根據上述程式做出之資料集進行 mining 後，得到以下十條 rule:

Idx	LHS	RHS	Conf	Lift
1.	Fruit, Pain Relievers, Miscellaneous	Vegetables	1	2.29
2.	Frozen Desserts, Kitchen Products, Cold Remedies	Vegetables	1	2.29
3.	Jams and Jellies, Starchy Foods, Canned Oysters	Snack Foods	1	2.38
4.	Vegetables, Baking Goods, Frozen Desserts, Canned Tuna	Snack Foods	1	2.38
5.	Meat, Baking Goods, Electrical, Pizza	Vegetables	1	2.29
6.	Breakfast Foods, Baking Goods, Canned Soup, Bathroom Products	Snack Foods	1	2.38
7.	Vegetables, Dairy, Canned Soup, Bathroom Products, Kitchen Products	Snack Foods	1	2.38
8.	Snack Foods, Dairy, Jams and Jellies, Beer and Wine, Magazines	Vegetables	94%	2.16
9.	Meat, Breakfast Foods, Electrical, Pizza	Vegetables	94%	2.16
10.	Baking Goods, Pain Relievers, Frozen Entrees	Vegetables	94%	2.15

Weka 原始輸出

=== Run information ===

Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -
C 0.9 -D 1.0E-4 -U 1.0 -M 1.0E-4

Relation: R_data_frame

Instances: 37851

Attributes: 47

Baking Goods
Bathroom Products
Beer and Wine
Bread
Breakfast Foods
Candles
Candy
Canned Anchovies
Canned Clams
Canned Oysters
Canned Sardines
Canned Shrimp
Canned Soup
Canned Tuna
Carbonated Beverages
Cleaning Supplies
Cold Remedies
Dairy
Decongestants
Drinks
Dry Goods
Eggs
Electrical
Frozen Desserts
Frozen Entrees
Fruit
Hardware
Hot Beverages
Hygiene
Jams and Jellies
Kitchen Products
Magazines
Meat
Miscellaneous
Packaged Soup
Packaged Vegetables
Pain Relievers
Paper Products
Pizza
Plastic Products
Pure Juice Beverages
Seafood
Side Dishes
Snack Foods
Specialty

```
Starchy Foods
Vegetables
=== Associator model (full training set) ===
```

```
FPGrowth found 60 rules (displaying top 10)
```

```
1. [Fruit=1, Pain Relievers=1, Miscellaneous=1]: 12 ==> [Vegeta
```

題目 3

有時候我們有興趣的資料不只有產品間的資訊，也會想要由User Profile探勘顧客的基本資料。請運用Weka，給定Minimum Support = 0.05且Minimum Confidence= 0.9的條件下，探勘Foodmart顧客基本資料的屬性{State_Province, Yearly_Income , Gender , Total_Children , Num_Children_at_Home , Education , Occupation, Houseowner , Num_cars,owned } 間的association rule。(列出10條)

資料前處理

In []:

```
db <- odbcDriverConnect("Driver={Microsoft Access Driver (*.mdb, *.accdb)};
                        DBQ=.\foodmart2000.mdb")

sql3 <- "
select distinct
  ct.customer_id,
  ct.state_province,
  ct.yearly_income,
  ct.gender,
  ct.total_children,
  ct.num_children_at_home,
  ct.education,
  ct.occupation,
  ct.houseowner,
  ct.num_cars_owned
from customer ct, (
  select
    customer_id
  from sales_fact_1998
  union
  select
    customer_id
  from sales_fact_dec_1998) as tx
where ct.customer_id = tx.customer_id ;
"

ct <- as.data.frame(sqlQuery(db, sql3))

head(ct)

for(idx in 2:10){
  ct[,idx] <- as.factor(ct[, idx])
}

write.arff(ct[,2:10], file = 'q3_customer.arff')
write.table(
  ct[,2:10],
  'q3_customer.csv',
  row.names = FALSE,
  col.names = TRUE,
  sep = ",",
  quote = FALSE)
```

Weka 結果

根據上述程式做出之資料集進行 mining 後，得到了許多結果，但排序在前段的有一些是沒有必要的，故過濾之後的前七項如下：

Idx	LHS	RHS	Conf	Lift
3.	num_cars_owned=0	yearly_income=10K–30K	1	4.66%
29.	yearly_income=10K–30K	education=Partial High School	92%	3.13%
11.	yearly_income=10K–30K occupation=Manual	education=Partial High School	96%	3.26
13.	yearly_income=10K–30K occupation=Skilled Manual	education=Partial High School	95%	3.25
22.	yearly_income=10K–30K houseowner=Y	education=Partial High School	93%	3.16
15.	yearly_income=50K–70K occupation=Professional	education=Bachelors Degree	95%	3.69
41.	education=High School Degree occupation=Manual	yearly_income=30K–50K	90%	2.75

無法做出前十項結果的原因在於，mining 結果有許多是集合的重疊，因此我選擇了較具識別性，也比較寬鬆的條件做為結果。

Weka 原始輸出

Weka 的原始輸出如下：

=== Run information ===

Scheme: weka.associations.Apriori -N 50 -T 0 -C 0.9 -D 0.0
5 -U 1.0 -M 0.05 -S -1.0 -c -1
Relation: R_data_frame
Instances: 8060
Attributes: 9
state_province
yearly_income
gender
total_children
num_children_at_home
education
occupation
houseowner
num_cars_owned

=== Associator model (full training set) ===

Apriori

=====

Minimum support: 0.05 (403 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 19

Generated sets of large itemsets:

Size of set of large itemsets L(1): 36

Size of set of large itemsets L(2): 244

Size of set of large itemsets L(3): 279

Size of set of large itemsets L(4): 47

Best rules found:

1. total_children=0 812 ==> num_children_at_home=0 812 <conf:(1)> lift:(1.58) lev:(0.04) [298] conv:(298.81)
2. total_children=0 houseowner=Y 499 ==> num_children_at_home=0 499 <conf:(1)> lift:(1.58) lev:(0.02) [183] conv:(183.63)
3. num_cars_owned=0 469 ==> yearly_income=\$10K - \$30K 469 <conf:(1)> lift:(4.63) lev:(0.05) [367] conv:(367.64)
4. gender=M total_children=0 438 ==> num_children_at_home=0 438 <conf:(1)> lift:(1.58) lev:(0.02) [161] conv:(161.18)
5. education=Partial High School num_cars_owned=0 432 ==> yearly_income=\$10K - \$30K 432 <conf:(1)> lift:(4.63) lev:(0.04) [338] conv:(338.63)
6. yearly_income=\$10K - \$30K gender=M occupation=Skilled Manual 419 ==> education=Partial High School 407 <conf:(0.97)> lift:(3.29) lev:(0.04) [283] conv:(22.71)

7. yearly_income=\$10K - \$30K occupation=Manual houseowner=Y 496
 ==> education=Partial High School 479 <conf:(0.97)> lift:(3.2
 7) lev:(0.04) [332] conv:(19.41)

8. yearly_income=\$10K - \$30K occupation=Skilled Manual houseown
 er=Y 479 ==> education=Partial High School 462 <conf:(0.96)>
 lift:(3.26) lev:(0.04) [320] conv:(18.75)

9. yearly_income=\$10K - \$30K num_children_at_home=0 occupation=
 Manual 530 ==> education=Partial High School 511 <conf:(0.96)
 > lift:(3.26) lev:(0.04) [354] conv:(18.67)

10. yearly_income=\$10K - \$30K occupation=Manual 837 ==> educatio
 n=Partial High School 803 <conf:(0.96)> lift:(3.25) lev:(0.0
 7) [555] conv:(16.85)

11. yearly_income=\$10K - \$30K gender=M occupation=Manual 436 ==>
 education=Partial High School 418 <conf:(0.96)> lift:(3.24) l
 ev:(0.04) [289] conv:(16.17)

12. yearly_income=\$10K - \$30K occupation=Skilled Manual num_cars
 _owned=1 433 ==> education=Partial High School 414 <conf:(0.9
 6)> lift:(3.24) lev:(0.04) [286] conv:(15.25)

13. yearly_income=\$10K - \$30K occupation=Skilled Manual 843 ==>
 education=Partial High School 806 <conf:(0.96)> lift:(3.24)
 lev:(0.07) [556] conv:(15.63)

14. yearly_income=\$10K - \$30K num_children_at_home=0 occupation=
 Skilled Manual 518 ==> education=Partial High School 495 <con
 f:(0.96)> lift:(3.23) lev:(0.04) [341] conv:(15.2)

15. yearly_income=\$50K - \$70K occupation=Professional 817 ==> ed
 ucation=Bachelors Degree 777 <conf:(0.95)> lift:(3.72) lev:
 (0.07) [568] conv:(14.83)

16. yearly_income=\$50K - \$70K occupation=Professional houseowner
 =Y 459 ==> education=Bachelors Degree 433 <conf:(0.94)> lift:
 (3.69) lev:(0.04) [315] conv:(12.66)

17. yearly_income=\$50K - \$70K num_children_at_home=0 occupation=
 Professional 532 ==> education=Bachelors Degree 501 <conf:(0.
 94)> lift:(3.68) lev:(0.05) [365] conv:(12.38)

18. education=Partial High School occupation=Skilled Manual num_
 cars_owned=1 440 ==> yearly_income=\$10K - \$30K 414 <conf:(0.9
 4)> lift:(4.35) lev:(0.04) [318] conv:(12.77)

19. yearly_income=\$10K - \$30K num_children_at_home=0 houseowner=
 Y 516 ==> education=Partial High School 485 <conf:(0.94)> lif
 t:(3.18) lev:(0.04) [332] conv:(11.36)

20. yearly_income=\$10K - \$30K houseowner=Y num_cars_owned=1 494
 ==> education=Partial High School 463 <conf:(0.94)> lift:(3.
 17) lev:(0.04) [317] conv:(10.88)

21. yearly_income=\$10K - \$30K gender=M houseowner=Y 522 ==> educ
 ation=Partial High School 489 <conf:(0.94)> lift:(3.17) lev:
 (0.04) [334] conv:(10.82)

22. yearly_income=\$10K - \$30K houseowner=Y 1009 ==> education=Pa
 rtial High School 941 <conf:(0.93)> lift:(3.16) lev:(0.08) [6
 42] conv:(10.3)

23. yearly_income=\$10K - \$30K gender=M 888 ==> education=Partial
 High School 825 <conf:(0.93)> lift:(3.14) lev:(0.07) [562] co
 nv:(9.77)

24. yearly_income=\$10K - \$30K gender=F houseowner=Y 487 ==> educ

ation=Partial High School 452 <conf:(0.93)> lift:(3.14) lev:
 (0.04) [308] conv:(9.53)
 25. yearly_income=\$10K - \$30K gender=F num_children_at_home=0 52
 6 ==> education=Partial High School 488 <conf:(0.93)> lift:
 (3.14) lev:(0.04) [332] conv:(9.5)
 26. yearly_income=\$10K - \$30K num_children_at_home=0 1086 ==> ed
 ucation=Partial High School 1006 <conf:(0.93)> lift:(3.13) le
 v:(0.08) [685] conv:(9.45)
 27. yearly_income=\$10K - \$30K num_cars_owned=1 859 ==> education
 =Partial High School 795 <conf:(0.93)> lift:(3.13) lev:(0.07)
 [541] conv:(9.31)
 28. yearly_income=\$10K - \$30K gender=M num_children_at_home=0 56
 0 ==> education=Partial High School 518 <conf:(0.93)> lift:
 (3.13) lev:(0.04) [352] conv:(9.17)
 29. yearly_income=\$10K - \$30K 1742 ==> education=Partial High Sc
 hool 1609 <conf:(0.92)> lift:(3.13) lev:(0.14) [1094] conv:
 (9.16)
 30. num_cars_owned=0 469 ==> education=Partial High School 432
 <conf:(0.92)> lift:(3.12) lev:(0.04) [293] conv:(8.69)
 31. yearly_income=\$10K - \$30K num_cars_owned=0 469 ==> education
 =Partial High School 432 <conf:(0.92)> lift:(3.12) lev:(0.04)
 [293] conv:(8.69)
 32. num_cars_owned=0 469 ==> yearly_income=\$10K - \$30K education
 =Partial High School 432 <conf:(0.92)> lift:(4.61) lev:(0.04)
 [338] conv:(9.88)
 33. yearly_income=\$10K - \$30K num_children_at_home=0 num_cars_ow
 ned=1 526 ==> education=Partial High School 484 <conf:(0.92)>
 lift:(3.11) lev:(0.04) [328] conv:(8.62)
 34. yearly_income=\$10K - \$30K gender=F 854 ==> education=Partial

題目 4

請運用Weka探勘Foodmart資料庫中，顧客背景資料與其交易資料之間的關係(Quantitative Association Rules)。例如80%女性顧客常買保養品。請自行嘗試設定Minimum Support Minimum Confidence，找出10條你覺得有意義的Rules。請說明你的作法及相關參數設定。

資料前處理

In []:

```
db <- odbcDriverConnect("Driver={Microsoft Access Driver (*.mdb, *.accdb)};
                        DBQ=.\foodmart2000.mdb")

sql4 <- "
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  ct.yearly_income,
  ct.gender,
  ct.occupation,
  pc.product_category as item
from sales_fact_1998 as ft, product as pd, product_class as pc, customer as ct
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id
and ft.customer_id = ct.customer_id
union all
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-' &
  trim(str(ft.store_id)) as tid,
  ct.yearly_income,
  ct.gender,
  ct.occupation,
  pc.product_category as item
from sales_fact_dec_1998 as ft, product as pd, product_class as pc,
  customer as ct
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id
and ft.customer_id = ct.customer_id;"

max_columns <- 5

ft <- as.data.frame(sqlQuery(db, sql4))

for(ci in 1:max_columns){
  ft[,ci] <- as.character(ft[,ci])
}

head(ft)
```

In []:

```
columns <- c("tid")
cate_sql <- "select distinct product_category from product_class;"
cust_attrs <- c("yearly_income", "gender", "education", "occupation")
attr_sqls = paste("select distinct", cust_attrs, "from customer;")
all_attr_sqls = c(attr_sqls, cate_sql)

for( sql in all_attr_sqls){
  pc <- as.data.frame(sqlQuery(db, sql))
  pc[,1] <- as.character(pc[,1])
  columns <- c(columns, as.vector(t(pc)[1,]))
}

len <- length(columns)
# tids <- head(unique(ft$tid), n=5000)
tids <- unique(ft$tid)
df <- as.data.frame(matrix(ncol = length(columns),
                           nrow=0,
                           dimnames = list(NULL,columns)))

ridx <- 1
for(tid in tids){
  df[ridx, 1] = tid
  df[ridx, 2:len] = 0

  cust_indices <- 2:(length(cust_attrs) + 1)
  for(cust_idx in cust_indices){
    ca <- unique(as.vector(ft[ft$tid==tid, cust_idx]))
    df[ridx, ca] = 1
  }

  prod_idx <- max_columns
  pcs <- unique(as.vector(ft[ft$tid==tid, prod_idx]))
  for(pc in pcs){
    df[ridx, pc] = 1
  }

  ridx <- ridx + 1
}

for (idx in 1:length(columns)){
  df[,idx] <- as.factor(df[,idx])
}

head(df)

write.arff(df[,2:len], file = 'q4_customer_tx.arff')
write.table(
  df[,2:len],
  'q4_customer_tx.csv',
  row.names = FALSE,
  col.names = TRUE,
  sep = ",",
  quote = FALSE)
```

關聯性分析

根據以上的程式，使用 FPGrowth 演算法，可以 mining 出許多的結果，我選擇前 100 項，再比對後發現以下的規則：

1. 這超市的女性客戶，大多是進行文書工作(Clerical)，常買的幾項有
 - 現切水果果汁
 - 小吃
 - 電子類產品
 - 肉類
 - 浴室用品
2. 而擴大來說，文書類工作的人，薪水大概都在 30K - 50K 上下，常買的有
 - 肉類
 - 果醬或果凍
 - 蔬菜
 - 小吃
 - 啤酒與紅酒
 - 奶製品
3. 另一方面來說，這超市的男性客戶，大多是進行專業類工作，常買的幾項有
 - 肉類
 - 麵包
 - 烘焙用品
 - 小吃
 - 水果
4. 擴大來說，專業類工作的人，薪水大概都在 130K - 150K 上下，常買的有
 - 奶製品
 - 硬體類 (像工具之類的)
 - 麵包
 - 紙製品
 - 罐頭
5. 而另一方面，來光臨的勞工們，大多會買
 - 披薩
 - 清潔用品
 - 消炎止痛藥
 - 小吃
 - 水果
 - 紙製品
6. 超市應該是女性客戶偏多，因為有 mining 出單以女性為條件的 rule，常買的有
 - 浴室用品
 - 個人衛生用品
 - 麵包
 - 冷凍食品
 - 清潔用品
7. 另外，有許多專業工作者會在超市買早餐，大多是：
 - 澱粉類
 - 水果
 - 奶製品
 - 果醬類

Weka 原始輸出

以下是 Weka 原始輸出結果：

=== Run information ===

Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 100 -T 0
-C 0.9 -D 5.0E-4 -U 1.0 -M 5.0E-4

Relation: R_data_frame

Instances: 37851

Attributes: 67

\$10K - \$30K

\$110K - \$130K

\$130K - \$150K

\$150K +

\$30K - \$50K

\$50K - \$70K

\$70K - \$90K

\$90K - \$110K

F

M

Bachelors Degree

Graduate Degree

High School Degree

Partial College

Partial High School

Clerical

Management

Manual

Professional

Skilled Manual

Baking Goods

Bathroom Products

Beer and Wine

Bread

Breakfast Foods

Candles

Candy

Canned Anchovies

Canned Clams

Canned Oysters

Canned Sardines

Canned Shrimp

Canned Soup

Canned Tuna

Carbonated Beverages

Cleaning Supplies

Cold Remedies

Dairy

Decongestants

Drinks

Dry Goods

Eggs

Electrical

Frozen Desserts

Frozen Entrees

Fruit
 Hardware
 Hot Beverages
 Hygiene
 Jams and Jellies
 Kitchen Products
 Magazines
 Meat
 Miscellaneous
 Packaged Soup
 Packaged Vegetables
 Pain Relievers
 Paper Products
 Pizza
 Plastic Products
 Pure Juice Beverages
 Seafood
 Side Dishes
 Snack Foods
 Specialty
 Starchy Foods
 Vegetables

=== Associator model (full training set) ===

FPGrowth found 82 rules (displaying top 82)

1. [F=1, Pure Juice Beverages=1, Clerical=1]: 24 ==> [\$30K - \$50K=1]: 24 <conf:(1)> lift:(3.05) lev:(0) conv:(16.13)
2. [Meat=1, Jams and Jellies=1, Clerical=1]: 25 ==> [\$30K - \$50K=1]: 25 <conf:(1)> lift:(3.05) lev:(0) conv:(16.81)
3. [Meat=1, Beer and Wine=1, Clerical=1]: 22 ==> [\$30K - \$50K=1]: 22 <conf:(1)> lift:(3.05) lev:(0) conv:(14.79)
4. [F=1, Snack Foods=1, Electrical=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 21 <conf:(1)> lift:(3.05) lev:(0) conv:(14.12)
5. [Vegetables=1, Snack Foods=1, Meat=1, Clerical=1]: 29 ==> [\$30K - \$50K=1]: 29 <conf:(1)> lift:(3.05) lev:(0) conv:(19.5)
6. [Vegetables=1, Dairy=1, Jams and Jellies=1, Clerical=1]: 19 ==> [\$30K - \$50K=1]: 19 <conf:(1)> lift:(3.05) lev:(0) conv:(12.77)
7. [F=1, Electrical=1, Clerical=1]: 37 ==> [\$30K - \$50K=1]: 36 <conf:(0.97)> lift:(2.97) lev:(0) conv:(12.44)
8. [F=1, Snack Foods=1, Meat=1, Clerical=1]: 29 ==> [\$30K - \$50K=1]: 28 <conf:(0.97)> lift:(2.95) lev:(0) conv:(9.75)
9. [Dairy=1, Bread=1, Clerical=1]: 27 ==> [\$30K - \$50K=1]: 26 <conf:(0.96)> lift:(2.94) lev:(0) conv:(9.08)
10. [Dairy=1, Bathroom Products=1, Clerical=1]: 24 ==> [\$30K - \$50K=1]: 23 <conf:(0.96)> lift:(2.92) lev:(0) conv:(8.07)
11. [F=1, Snack Foods=1, Bathroom Products=1, Clerical=1]: 24 ==> [\$30K - \$50K=1]: 23 <conf:(0.96)> lift:(2.92) lev:(0) conv:(8.07)
12. [F=1, Candy=1, Clerical=1]: 23 ==> [\$30K - \$50K=1]: 22 <co

nf:(0.96)> lift:(2.92) lev:(0) conv:(7.73)
 13. [Snack Foods=1, Hardware=1, Clerical=1]: 23 ==> [\$30K - \$50K=1]: 22 <conf:(0.96)> lift:(2.92) lev:(0) conv:(7.73)
 14. [Dairy=1, \$130K - \$150K=1, Hardware=1]: 23 ==> [Professional=1]: 22 <conf:(0.96)> lift:(2.95) lev:(0) conv:(7.78)
 15. [Bread=1, Paper Products=1, \$110K - \$130K=1]: 23 ==> [Professional=1]: 22 <conf:(0.96)> lift:(2.95) lev:(0) conv:(7.78)
 16. [F=1, Dairy=1, Meat=1, Clerical=1]: 23 ==> [\$30K - \$50K=1]: 22 <conf:(0.96)> lift:(2.92) lev:(0) conv:(7.73)
 17. [F=1, Bathroom Products=1, Clerical=1]: 45 ==> [\$30K - \$50K=1]: 43 <conf:(0.96)> lift:(2.92) lev:(0) conv:(10.08)
 18. [F=1, Jams and Jellies=1, Clerical=1]: 44 ==> [\$30K - \$50K=1]: 42 <conf:(0.95)> lift:(2.91) lev:(0) conv:(9.86)
 19. [\$110K - \$130K=1, Canned Clams=1]: 21 ==> [Professional=1]: 20 <conf:(0.95)> lift:(2.94) lev:(0) conv:(7.1)
 20. [F=1, Kitchen Products=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 20 <conf:(0.95)> lift:(2.91) lev:(0) conv:(7.06)
 21. [F=1, Snack Foods=1, Jams and Jellies=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 20 <conf:(0.95)> lift:(2.91) lev:(0) conv:(7.06)
 22. [F=1, Meat=1, Fruit=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 20 <conf:(0.95)> lift:(2.91) lev:(0) conv:(7.06)
 23. [F=1, Bread=1, Clerical=1]: 39 ==> [\$30K - \$50K=1]: 37 <conf:(0.95)> lift:(2.89) lev:(0) conv:(8.74)
 24. [F=1, Baking Goods=1, Clerical=1]: 54 ==> [\$30K - \$50K=1]: 51 <conf:(0.94)> lift:(2.88) lev:(0) conv:(9.08)
 25. [F=1, Vegetables=1, Meat=1, Clerical=1]: 36 ==> [\$30K - \$50K=1]: 34 <conf:(0.94)> lift:(2.88) lev:(0) conv:(8.07)
 26. [Vegetables=1, Meat=1, Clerical=1]: 70 ==> [\$30K - \$50K=1]: 66 <conf:(0.94)> lift:(2.88) lev:(0) conv:(9.41)
 27. [M=1, Vegetables=1, Meat=1, Clerical=1]: 34 ==> [\$30K - \$50K=1]: 32 <conf:(0.94)> lift:(2.87) lev:(0) conv:(7.62)
 28. [Snack Foods=1, Meat=1, Clerical=1]: 60 ==> [\$30K - \$50K=1]: 56 <conf:(0.93)> lift:(2.85) lev:(0) conv:(8.07)
 29. [Snack Foods=1, Paper Products=1, Clerical=1]: 30 ==> [\$30K - \$50K=1]: 28 <conf:(0.93)> lift:(2.85) lev:(0) conv:(6.72)
 30. [Vegetables=1, Dairy=1, Meat=1, Clerical=1]: 29 ==> [\$30K - \$50K=1]: 27 <conf:(0.93)> lift:(2.84) lev:(0) conv:(6.5)
 31. [F=1, Meat=1, Clerical=1]: 72 ==> [\$30K - \$50K=1]: 67 <conf:(0.93)> lift:(2.84) lev:(0) conv:(8.07)
 32. [Dairy=1, Meat=1, Clerical=1]: 42 ==> [\$30K - \$50K=1]: 39 <conf:(0.93)> lift:(2.83) lev:(0) conv:(7.06)
 33. [M=1, Meat=1, Bread=1, \$110K - \$130K=1]: 27 ==> [Professional=1]: 25 <conf:(0.93)> lift:(2.86) lev:(0) conv:(6.09)
 34. [F=1, Frozen Desserts=1, Clerical=1]: 26 ==> [\$30K - \$50K=1]: 24 <conf:(0.92)> lift:(2.82) lev:(0) conv:(5.83)
 35. [Electrical=1, Paper Products=1, \$110K - \$130K=1]: 25 ==> [Professional=1]: 23 <conf:(0.92)> lift:(2.84) lev:(0) conv:(5.63)
 36. [Snack Foods=1, Fruit=1, Paper Products=1, \$90K - \$110K=1]: 25 ==> [Professional=1]: 23 <conf:(0.92)> lift:(2.84) lev:(0) conv:(5.63)

37. [F=1, Paper Products=1, Clerical=1]: 37 ==> [\$30K - \$50K=1]: 34 <conf:(0.92)> lift:(2.8) lev:(0) conv:(6.22)

38. [M=1, Baking Goods=1, \$150K +=1]: 48 ==> [Professional=1]: 44 <conf:(0.92)> lift:(2.83) lev:(0) conv:(6.49)

39. [Jams and Jellies=1, Hygiene=1, Hardware=1]: 24 ==> [Snack Foods=1]: 22 <conf:(0.92)> lift:(2.18) lev:(0) conv:(4.64)

40. [Fruit=1, Frozen Desserts=1, \$150K +=1]: 24 ==> [Professional=1]: 22 <conf:(0.92)> lift:(2.83) lev:(0) conv:(5.41)

41. [F=1, Snack Foods=1, Dairy=1, \$70K - \$90K=1, Eggs=1]: 24 ==> [Professional=1]: 22 <conf:(0.92)> lift:(2.83) lev:(0) conv:(5.41)

42. [Vegetables=1, Electrical=1, Clerical=1]: 35 ==> [\$30K - \$50K=1]: 32 <conf:(0.91)> lift:(2.79) lev:(0) conv:(5.88)

43. [\$10K - \$30K=1, Kitchen Products=1, Decongestants=1]: 23 ==> [Vegetables=1]: 21 <conf:(0.91)> lift:(2.09) lev:(0) conv:(4.32)

44. [Fruit=1, Hygiene=1, Hardware=1]: 23 ==> [Snack Foods=1]: 21 <conf:(0.91)> lift:(2.17) lev:(0) conv:(4.44)

45. [Meat=1, Bathroom Products=1, Clerical=1]: 23 ==> [\$30K - \$50K=1]: 21 <conf:(0.91)> lift:(2.79) lev:(0) conv:(5.15)

46. [F=1, Fruit=1, Breakfast Foods=1, \$90K - \$110K=1]: 23 ==> [Professional=1]: 21 <conf:(0.91)> lift:(2.82) lev:(0) conv:(5.18)

47. [M=1, Snack Foods=1, Fruit=1, \$70K - \$90K=1, Candy=1]: 23 ==> [Professional=1]: 21 <conf:(0.91)> lift:(2.82) lev:(0) conv:(5.18)

48. [Vegetables=1, Jams and Jellies=1, Clerical=1]: 57 ==> [\$30K - \$50K=1]: 52 <conf:(0.91)> lift:(2.78) lev:(0) conv:(6.39)

49. [F=1, Fruit=1, Clerical=1]: 68 ==> [\$30K - \$50K=1]: 62 <conf:(0.91)> lift:(2.78) lev:(0) conv:(6.53)

50. [Meat=1, Fruit=1, Clerical=1]: 34 ==> [\$30K - \$50K=1]: 31 <conf:(0.91)> lift:(2.78) lev:(0) conv:(5.71)

51. [Snack Foods=1, Bathroom Products=1, Clerical=1]: 45 ==> [\$30K - \$50K=1]: 41 <conf:(0.91)> lift:(2.78) lev:(0) conv:(6.05)

52. [F=1, Pain Relievers=1, Clerical=1]: 22 ==> [\$30K - \$50K=1]: 20 <conf:(0.91)> lift:(2.77) lev:(0) conv:(4.93)

53. [F=1, Skilled Manual=1, Pizza=1, Cleaning Supplies=1]: 22 ==> [Vegetables=1]: 20 <conf:(0.91)> lift:(2.08) lev:(0) conv:(4.13)

54. [M=1, Snack Foods=1, Baking Goods=1, \$150K +=1]: 22 ==> [Professional=1]: 20 <conf:(0.91)> lift:(2.81) lev:(0) conv:(4.96)

55. [Vegetables=1, \$10K - \$30K=1, Magazines=1, Pain Relievers=1]: 22 ==> [Snack Foods=1]: 20 <conf:(0.91)> lift:(2.16) lev:(0) conv:(4.25)

56. [Snack Foods=1, Bread=1, Bathroom Products=1, Drinks=1]: 22 ==> [Vegetables=1]: 20 <conf:(0.91)> lift:(2.08) lev:(0) conv:(4.13)

57. [M=1, Snack Foods=1, Manual=1, Paper Products=1, Pain Relievers=1]: 22 ==> [Vegetables=1]: 20 <conf:(0.91)> lift:(2.08) lev:(0) conv:(4.13)

58. [M=1, Snack Foods=1, Fruit=1, Paper Products=1, Pain Relievers=1]: 22 ==> [Vegetables=1]: 20 <conf:(0.91)> lift:(2.08) lev:(0) conv:(4.13)

v:(0) conv:(4.13)

59. [Snack Foods=1, \$30K - \$50K=1, Fruit=1, Canned Soup=1, Starchy Foods=1]: 22 ==> [Vegetables=1]: 20 <conf:(0.91)> lift:(2.08) lev:(0) conv:(4.13)

60. [Vegetables=1, Snack Foods=1, Meat=1, \$70K - \$90K=1, Frozen Desserts=1]: 22 ==> [Professional=1]: 20 <conf:(0.91)> lift:(2.81) lev:(0) conv:(4.96)

61. [F=1, Beer and Wine=1, Clerical=1]: 43 ==> [\$30K - \$50K=1]: 39 <conf:(0.91)> lift:(2.77) lev:(0) conv:(5.78)

62. [Meat=1, Clerical=1]: 139 ==> [\$30K - \$50K=1]: 126 <conf:(0.91)> lift:(2.77) lev:(0) conv:(6.67)

63. [Electrical=1, Clerical=1]: 85 ==> [\$30K - \$50K=1]: 77 <conf:(0.91)> lift:(2.76) lev:(0) conv:(6.35)

64. [Pure Juice Beverages=1, Clerical=1]: 42 ==> [\$30K - \$50K=1]: 38 <conf:(0.9)> lift:(2.76) lev:(0) conv:(5.65)

65. [F=1, Vegetables=1, Clerical=1]: 126 ==> [\$30K - \$50K=1]: 114 <conf:(0.9)> lift:(2.76) lev:(0) conv:(6.52)

66. [Bread=1, Hygiene=1, Frozen Entrees=1]: 21 ==> [F=1]: 19 <conf:(0.9)> lift:(1.79) lev:(0) conv:(3.46)

67. [Beer and Wine=1, Eggs=1, Cleaning Supplies=1]: 21 ==> [F=1]: 19 <conf:(0.9)> lift:(1.79) lev:(0) conv:(3.46)

68. [M=1, Hardware=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 19 <conf:(0.9)> lift:(2.76) lev:(0) conv:(4.71)

69. [Fruit=1, Breakfast Foods=1, \$150K +=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

70. [Bathroom Products=1, Pure Juice Beverages=1, \$90K - \$110K=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

71. [F=1, Snack Foods=1, Canned Soup=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 19 <conf:(0.9)> lift:(2.76) lev:(0) conv:(4.71)

72. [F=1, \$50K - \$70K=1, Bathroom Products=1, Hygiene=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

73. [M=1, Jams and Jellies=1, Breakfast Foods=1, \$110K - \$130K=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

74. [M=1, Breakfast Foods=1, Starchy Foods=1, \$130K - \$150K=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

75. [Meat=1, Breakfast Foods=1, Baking Goods=1, Specialty=1]: 21 ==> [Vegetables=1]: 19 <conf:(0.9)> lift:(2.07) lev:(0) conv:(3.94)

76. [Snack Foods=1, Dairy=1, Jams and Jellies=1, Clerical=1]: 21 ==> [\$30K - \$50K=1]: 19 <conf:(0.9)> lift:(2.76) lev:(0) conv:(4.71)

77. [Dairy=1, Jams and Jellies=1, Breakfast Foods=1, \$90K - \$110K=1]: 21 ==> [Professional=1]: 19 <conf:(0.9)> lift:(2.79) lev:(0) conv:(4.73)

78. [Cold Remedies=1, Clerical=1]: 31 ==> [\$30K - \$50K=1]: 28 <conf:(0.9)> lift:(2.76) lev:(0) conv:(5.21)

79. [M=1, Snack Foods=1, Meat=1, Clerical=1]: 31 ==> [\$30K - \$50K=1]: 28 <conf:(0.9)> lift:(2.76) lev:(0) conv:(5.21)

```
80. [F=1, Clerical=1]: 309 ==> [$30K - $50K=1]: 279 <conf:(0.9)> lift:(2.76) lev:(0) conv:(6.7)
81. [F=1, Snack Foods=1, Clerical=1]: 123 ==> [$30K - $50K=1]: 111 <conf:(0.9)> lift:(2.75) lev:(0) conv:(6.36)
82. [Snack Foods=1, Electrical=1, Clerical=1]: 41 ==> [$30K - $50K=1]: 37 <conf:(0.9)> lift:(2.75) lev:(0) conv:(5.51)
```

題目 5

在美國由於聖誕節，12月是購物的旺季。請探勘分析比較12月與1~11月的顧客購物行為。有哪些相似的地方，有哪些差異的地方？

資料前處理

本題的資料前處理使用 FP-Growth 的做法，只是區別兩個不同的 SQL：

```
sql5_xmas <- "
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-'
  & trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_dec_1998 as ft, product as pd, product_class as
  pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id;"
```

```
sql5_normal <- "
select
  trim(str(ft.customer_id)) & '-' & trim(str(ft.time_id)) & '-'
  & trim(str(ft.store_id)) as tid,
  pc.product_category as item
from sales_fact_1998 as ft, product as pd, product_class as pc
where pd.product_id = ft.product_id
and pd.product_class_id = pc.product_class_id;"
```

以取得兩個不同的資料集，再以同樣的參數進行 mining。

聖誕月銷售分析

根據上述程式做出之資料集進行 mining 後，得到以下十條 rule:

Idx	LHS	RHS	Conf	Lift
1.	Hygiene, Side Dishes	Vegetables	1	2.27
2.	Hardware, Canned Tuna	Vegetables	1	2.27
3.	Meat, Bathroom Products, Pure Juice Beverages	Vegetables	1	2.27
4.	Jams and Jellies, Canned Soup, Pure Juice Beverages	Vegetables	1	2.27
5.	Breakfast Foods, Bathroom Products, Pure Juice Beverages	Vegetables	1	2.27
6.	Snack Foods, Meat, Baking Goods, Paper Products	Vegetables	1	2.27
7.	Fruit, Baking Goods, Pizza	Snack Foods	91%	2.16
8.	Breakfast Foods, Baking Goods, Canned Soup	Snack Foods	91%	2.16
9.	Vegetables, Meat, Paper Products, Frozen Desserts	Snack Foods	91%	2.16
10.	Baking Goods, Electrical, Hot Beverages	Snack Foods	90%	2.14

=== Run information ===

Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -
C 0.9 -D 1.0E-4 -U 1.0 -M 1.0E-4

Relation: R_data_frame

Instances: 3781

Attributes: 47

Baking Goods
Bathroom Products
Beer and Wine
Bread
Breakfast Foods
Candles
Candy
Canned Anchovies
Canned Clams
Canned Oysters
Canned Sardines
Canned Shrimp
Canned Soup
Canned Tuna
Carbonated Beverages
Cleaning Supplies
Cold Remedies
Dairy
Decongestants
Drinks
Dry Goods
Eggs
Electrical
Frozen Desserts
Frozen Entrees
Fruit
Hardware
Hot Beverages
Hygiene
Jams and Jellies
Kitchen Products
Magazines
Meat
Miscellaneous
Packaged Soup
Packaged Vegetables
Pain Relievers
Paper Products
Pizza
Plastic Products
Pure Juice Beverages
Seafood
Side Dishes

Snack Foods
Specialty
Starchy Foods
Vegetables

=== Associator model (full training set) ===

FPGrowth found 11 rules (displaying top 10)

1. [Hygiene=1, Side Dishes=1]: 9 ==> [Vegetables=1]: 9 <conf:(1)> lift:(2.27) lev:(0) conv:(5.03)
2. [Hardware=1, Canned Tuna=1]: 8 ==> [Vegetables=1]: 8 <conf:(1)> lift:(2.27) lev:(0) conv:(4.47)
3. [Meat=1, Bathroom Products=1, Pure Juice Beverages=1]: 8 ==> [Vegetables=1]: 8 <conf:(1)> lift:(2.27) lev:(0) conv:(4.47)
4. [Jams and Jellies=1, Canned Soup=1, Pure Juice Beverages=1]: 9 ==> [Vegetables=1]: 9 <conf:(1)> lift:(2.27) lev:(0) conv:(5.03)
5. [Breakfast Foods=1, Bathroom Products=1, Pure Juice Beverages=1]: 8 ==> [Vegetables=1]: 8 <conf:(1)> lift:(2.27) lev:(0) conv:(4.47)
6. [Snack Foods=1, Meat=1, Baking Goods=1, Paper Products=1]: 9 ==> [Vegetables=1]: 9 <conf:(1)> lift:(2.27) lev:(0) conv:(5.03)
7. [Fruit=1, Baking Goods=1, Pizza=1]: 11 ==> [Snack Foods=1]: 10 <conf:(0.91)> lift:(2.16) lev:(0) conv:(3.19)
8. [Breakfast Foods=1, Baking Goods=1, Canned Soup=1]: 11 ==> [Snack Foods=1]: 10 <conf:(0.91)> lift:(2.16) lev:(0) conv:(3.19)
9. [Vegetables=1, Meat=1, Paper Products=1, Frozen Desserts=1]: 11 ==> [Snack Foods=1]: 10 <conf:(0.91)> lift:(2.16) lev:(0) conv:(3.19)
10. [Baking Goods=1, Electrical=1, Hot Beverages=1]: 10 ==> [Snack Foods=1]: 9 <conf:(0.9)> lift:(2.14) lev:(0) conv:(2.9)

平常月銷售分析

根據上述程式做出之資料集進行 mining 後，得到以下十條 rule:

Idx	LHS	RHS	Conf	Lift
1.	Meat, Fruit, Candy, Kitchen Products	Vegetables	1	2.29
2.	Meat, Breakfast Foods, Baking Goods, Specialty	Vegetables	94%	2.17
3.	Jams and Jellies, Beer and Wine, Paper Products, Kitchen Products	Vegetables	94%	2.16
4.	Snack Foods, Dairy, Jams and Jellies, Beer and Wine, Magazines	Vegetables	94%	2.16
5.	Snack Foods, Fruit, Pain Relievers, Pizza	Vegetables	94%	2.15
6.	Dairy, Meat, Baking Goods, Frozen Entrees	Vegetables	94%	2.15
7.	Vegetables, Dairy, Canned Soup, Candy, Beer and Wine	Snack Foods	94%	2.23
8.	Snack Foods, Meat, Jams and Jellies, Miscellaneous	Vegetables	93%	2.14
9.	Meat, Breakfast Foods, Electrical, Pizza	Vegetables	93%	2.14
10.	Fruit, Candy, Electrical, Hot Beverages	Vegetables	93%	2.14

Weka 原始輸出

以下是 Weka 輸出結果：

=== Run information ===

Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -
C 0.9 -D 1.0E-4 -U 1.0 -M 1.0E-4

Relation: R_data_frame

Instances: 34070

Attributes: 47

Baking Goods
Bathroom Products
Beer and Wine
Bread
Breakfast Foods
Candles
Candy
Canned Anchovies
Canned Clams
Canned Oysters
Canned Sardines
Canned Shrimp
Canned Soup
Canned Tuna
Carbonated Beverages
Cleaning Supplies
Cold Remedies
Dairy
Decongestants
Drinks
Dry Goods
Eggs
Electrical
Frozen Desserts
Frozen Entrees
Fruit
Hardware
Hot Beverages
Hygiene
Jams and Jellies
Kitchen Products
Magazines
Meat
Miscellaneous
Packaged Soup
Packaged Vegetables
Pain Relievers
Paper Products
Pizza
Plastic Products
Pure Juice Beverages
Seafood
Side Dishes
Snack Foods
Specialty

Starchy Foods

Vegetables

=== Associator model (full training set) ===

FPGrowth found 17 rules (displaying top 10)

1. [Meat=1, Fruit=1, Candy=1, Kitchen Products=1]: 14 ==> [Vegetables=1]: 14 <conf:(1)> lift:(2.29) lev:(0) conv:(7.89)