Paper AA07-2013

Introduction to Market Basket Analysis

Bill Qualls, First Analytics, Raleigh, NC

ABSTRACT

Market Basket Analysis (MBA) is a data mining technique which is widely used in the consumer package goods (CPG) industry to identify which items are purchased together and, more importantly, how the purchase of one item affects the likelihood of another item being purchased. This paper will first discuss this traditional use of MBA, as well as introduce the concepts of support, confidence, and lift. It will then show how one company used MBA to analyze safety data in an attempt to identify factors contributing to injuries. Finally, a Base SAS macro which performs MBA will be provided and its usage demonstrated. Intended audience is anyone interested in data mining techniques in general, and in market basket analysis in particular, and while a Base SAS macro will be provided, no programming knowledge is required, and non-programmers will benefit from this paper.

INTRODUCTION

Market Basket Analysis (MBA) is a data mining technique which is widely used in the consumer package goods (CPG) industry to identify which items are purchased together. The classic example of MBA is diapers and beer:

> "An apocryphal early illustrative example for this was when one super market chain discovered in its analysis that customers that bought diapers often bought beer as well, have put the diapers close to beer coolers, and their sales increased dramatically. Although this urban legend is only an example that professors use to illustrate the concept to students, the explanation of this imaginary phenomenon might be that fathers that are sent out to buy diapers often buy a beer as well, as a reward." (Retrieved May 5, 2013 from http://en.wikipedia.org/wiki/Market_basket)

The example may or may not be true, but it illustrates the point of MBA. This paper will introduce the concepts of support, confidence, and lift as used in MBA. It will then show how one company used MBA to analyze safety data in an attempt to identify factors contributing to injuries. Finally, a Base SAS macro which performs MBA will be provided and its usage demonstrated.

SALES TRANSACTIONS

Our imaginary store sales the following items: bananas, bologna, bread, buns, butter, cereal, cheese, chips, eggs, hotdogs, mayo, milk, mustard, oranges, pickles, and soda. We have recorded 20 sales transactions as follows:

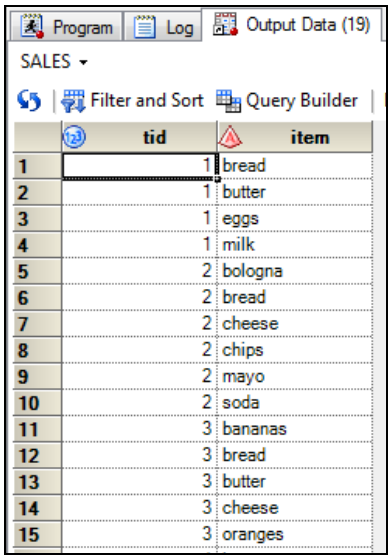
#1	#4	#7	#9	#12	#15	#18
bread	buns	bananas	bananas	bread	bologna	buns
butter	chips	cereal	bologna	butter	bread	cheese
eggs	hotdogs	eggs	bread	eggs	cheese	chips
milk	mustard	milk	cheese	milk	chips	hotdogs
	soda	oranges	milk	oranges	mayo	mustard
#2			oranges		mustard	soda
bologna	#5	#8	soda	#13	soda	
bread	buns	bologna		bananas		#19
cheese	chips	bread	#10	bologna	#16	chips
chips	hotdogs	buns	bread	bread	bread	pickles
mayo	mustard	cheese	butter	cheese	butter	soda
soda	pickles	chips	cereal	mayo	eggs	
	soda	hotdogs	eggs	mustard	milk	#20
#3		mayo	milk		oranges	bologna
bananas	#6	mustard		#14		bread
bread	bread	soda	#11	bread	#17	cheese
butter	butter		bananas	cereal	buns	chips
cheese	cereal		chips	eggs	chips	mayo
oranges	eggs		soda	milk	hotdogs	mustard
	milk				soda	soda

Figure 1. Sales transactions as recorded.

The MBA macro requires that each row of the input dataset have a transaction ID and an item. We can create that dataset with a simple DATA STEP:

```
data work.sales;
input tid item $;
datalines;
1 bread
1 butter
1 eggs
1 milk
2 bologna
2 bread
…
20 mustard
20 soda
;
run;
```

The resulting SAS dataset appears as follows:



	tid	item
1	1	bread
2	1	butter
3	1	eggs
4	1	milk
5	2	bologna
6	2	bread
7	2	cheese
8	2	chips
9	2	mayo
10	2	soda
11	3	bananas
12	3	bread
13	3	butter
14	3	cheese
15	3	oranges

Figure 2. Sales transactions as a SAS dataset

SUPPORT

There is often little benefit in examining extremely rare events. Support is one way to filter out such events. The **support** of an item is the number of transactions containing that item. Items not meeting the minimum support criteria are excluded from further analysis. For our purposes, we will assume a minimum support requirement of four. (Within the MBA macro, support can be expressed as a count or as a percentage of all transactions.) In Figure 3 we can see that pickles do not meet our minimum support requirements.

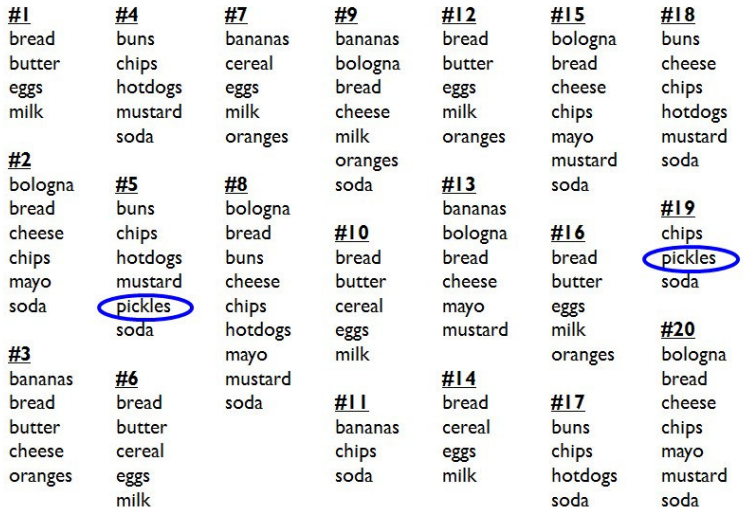


Figure 3. Pickles fail to meet our minimum support requirement (4).

PAIRS

We then create all possible pairings of the surviving items. Each pair is then checked to see that it, too, meets the minimum support requirement. The support of each pair of items is the number of transactions containing that pair. Pairs of items not meeting the minimum support criteria are excluded from further processing. Limiting ourselves to the surviving items is the key point of the **apriori algorithm**.

In Figure 4 we see that bananas and oranges each have a support of 5, but the pair (bananas, oranges) only has a support of 3, which is less than our requirement of 4, so that pairing will be excluded.

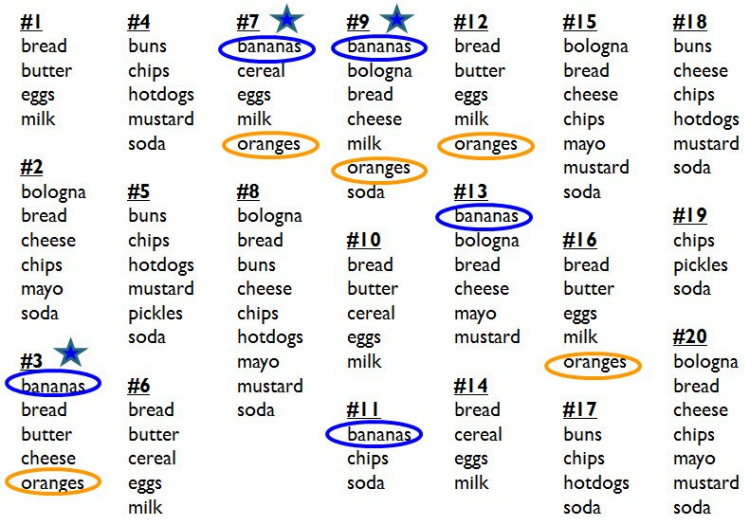


Figure 4. Bananas and oranges meet our support requirement individually but not as a pair.

In Figure 5 we see that bologna has a support of 6, and chips has a support of 10. Furthermore, the pair (bologna, chips) has a support of 4, which does meet our support requirement, so that pairing will be included.

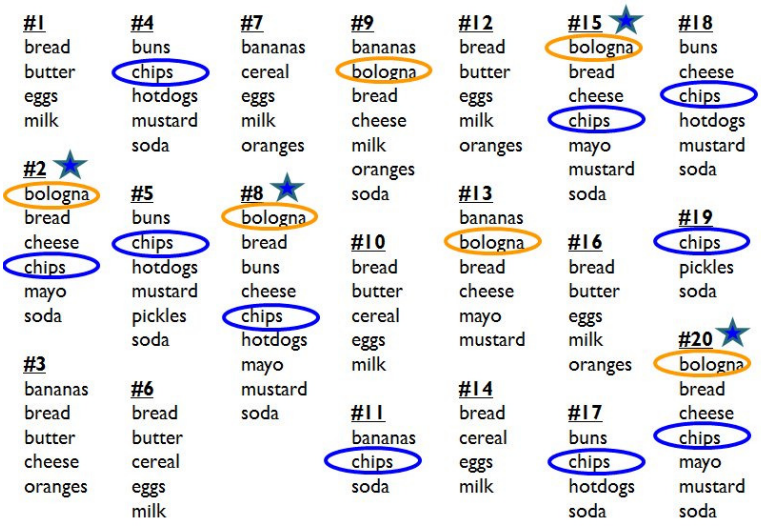


Figure 5. Bologna and chips meet our support requirement individually and as a pair.

ITERATE

We then repeat the process, iterating with itemsets of size three, size four, etc. until

- we are unable to find any itemsets with sufficient support, or
- we reach the maximum number of iterations as specified in the macro.

While it may initially be appealing to try a large number of iterations, it can be difficult to come up with a "story" to explain itemsets resulting from more than three iterations.

ASSOCIATION RULES

Our final results are often expressed as **association rules** and take the following form (where **LHS** stands for left hand side and **RHS** stands for right hand side):

{LHS} → {RHS} [support, confidence]

For example:

{chips} → {bologna} [0.20, 0.40]

where 0.20 is the support (calculated as 4/20) and 0.40 is the confidence (discussed next). The support value of 0.20 means that the pair (chips, bologna) appear in 20% of the transactions.

CONFIDENCE

Confidence is defined as the conditional probability that a transaction containing the LHS will also contain the RHS.

$$\text{Confidence}(LHS \rightarrow RHS) = P(RHS \mid LHS) = \frac{P(RHS \cap LHS)}{P(LHS)} = \frac{\text{support}(LHS \cap RHS)}{\text{support}(LHS)}$$

So the confidence for {chips}→{bologna} is calculated as

$$P(\text{bologna} \mid \text{chips}) = \frac{P(\text{bologna} \cap \text{chips})}{P(\text{chips})} = \frac{\text{support}(\text{chips} \cap \text{bologna})}{\text{support}(\text{chips})} = \frac{4/20}{10/20} = 0.40$$

LIFT

Lift is a measure of the improvement in the occurrence of the RHS given the LHS: it is the **ratio** of the conditional probability of the RHS given the LHS, divided by the unconditional probability of the RHS.

$$\text{Lift}(LHS \rightarrow RHS) = \frac{P(RHS \mid LHS)}{P(RHS)} = \frac{\text{confidence}(LHS \rightarrow RHS)}{\text{support}(RHS)}$$

So the confidence for {chips}→{bologna} is calculated as

$$\text{Lift}(\text{chips} \rightarrow \text{bologna}) = \frac{\text{confidence}(\text{chips} \rightarrow \text{bologna})}{\text{support}(\text{bologna})} = \frac{0.40}{6/20} = 1.33$$

As lift is a ratio, we are usually interested in a value greater than one.

In addition to showing support, confidence, and lift, the MBA macro will also highlight **possible interaction effects**, defined as those situations where Lift(AB→C) > max(Lift(A→C), Lift(B→C)).

THE MBA MACRO

The SAS code for the MBA macro is included at the end of this paper. The following code executes the macro. The keyword parameters are self-explanatory:

```
%mba(TRANS_FILE=work.sales
, ITEM_ID_IS_STRING="Y"
, MAXIMUM_ITERATIONS=3
, MINIMUM_SUPPORT=0.2
, RHS=('bologna' 'bread')
, RESULTS_FILE=Perm.Results_Groc
, WEBPAGE="C:\Users\Owner\Desktop\MWSUG\mba_groc.html"
);
run;
```

The macro produces a SAS dataset of results which facilitates post-processing. It also produces a report in HTML format. The RHS parameter lets you list those RHS items to be included in the webpage. If the RHS parameter is omitted, all itemsets meeting the minimum support requirement will be included. An image of the webpage is included at the end of this paper.

IT'S NOT JUST FOR GROCERIES!

While retailing may have been the impetus for market basket analysis, its use is certainly not limited to groceries! Our company used MBA to analyze injury data. Consider the following "available items": prior disciplinary event, prior positive alcohol test, prior injury, shift location, shift time, and injury.

One "shopper" may have picked up a prior positive alcohol test, a prior injury, a night shift at location A, and an injury. Another "shopper" may have picked up a prior disciplinary event, a day shift at location B, but no injury. We would be interested in the LHS factors when RHS = injury.

CONCLUSION

This paper has introduced market basket analysis, as well as its key metrics: support, confidence, and lift. We have also seen how market basket analysis can be used to identify potential interaction effects, and how it can be used in other areas besides retail, such as in examining injury data. Indeed, the use of market basket analysis is limited only by your imagination!

ACKNOWLEDGMENTS

The author would like to thank Dr. Michael Thompson of First Analytics, and Drs. Daniela Raicu, Raffaella Settimi, and Jonathan Gemmel, all of DePaul University.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Bill Qualls
Enterprise: First Analytics
Address: 1009 Capability Drive, Suite 314
City, State ZIP: Raleigh, NC 27606
Work Phone: 630-542-5185
E-mail: bqualls@firstanalytics.com
Web: http://www.firstanalytics.com

SAS SOURCE CODE

```
* ------------------------------------------------------------------------ *
*                  S E L E C T _ D I S T I N C T _ I T E M S               *
* ------------------------------------------------------------------------ *
*  This is one of several helper macros for my Market Basket Analysis macro.  *
* ------------------------------------------------------------------------ * ;

%macro select_distinct_items(CANDIDATE_LIST=, ITEMS_EACH_ROW=);

%global ITEM_LIST ITEM_COUNT; * Will be used in subsequent macros ;
%local i;

proc sql noprint;
%if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
    select distinct """ || trim(item) || """, count(distinct item)
%end;
%else %do;
    select distinct item, count(distinct item)
%end;
into :ITEM_LIST separated by " ", :ITEM_COUNT
from
(
%do i = 1 %to &ITEMS_EACH_ROW;
    %if (&i > 1) %then %do;
        union
    %end;
    select item&i as item from &CANDIDATE_LIST
%end;
)
;
run;

%put ITEM_LIST = &ITEM_LIST;
%put ITEM_COUNT = &ITEM_COUNT;

%mend select_distinct_items;


* ------------------------------------------------------------------------ *
*                  C R E A T E _ C A N D I D A T E _ S E T                 *
* ------------------------------------------------------------------------ *
*  This is one of several helper macros for my Market Basket Analysis macro.  *
* ------------------------------------------------------------------------ * ;

%macro create_candidate_set(ITEM_LIST=, ITEM_COUNT=, ITEMSET_SIZE=,
OUTPUT_CANDIDATE_FILE=);

%local i;

data &OUTPUT_CANDIDATE_FILE (keep =
    %do i = 1 %to &ITEMSET_SIZE;
        item&I
    %end;
    );

array item[&ITEM_COUNT]
%if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
    $32
%end;
(&ITEM_LIST);

i = 0;
```

```
do while ( 1 = 1 );
   i = i + 1;
   rc = lexcomb(i, &ITEMSET_SIZE, of item[*]);
   if (rc < 0) then leave;
   output;
end;

run;

*title "&OUTPUT_CANDIDATE_FILE";
*proc print data=&OUTPUT_CANDIDATE_FILE;
*run;

%mend create_candidate_set;


* ------------------------------------------------------------------------- *
*                  F I R S T _ L I S T _ I S _ T R I V I A L                *
* ------------------------------------------------------------------------- *
*  This is one of several helper macros for my Market Basket Analysis macro.  *
* ------------------------------------------------------------------------- * ;

%macro first_list_is_trivial(OUTPUT_LIST_FILE=);

proc sql noprint;
create table &OUTPUT_LIST_FILE as
select item as item1
, count(*) as Support_Count
, count(*) / &GLBL_MBA_TRANSACTION_COUNT as support
from &GLBL_MBA_TRANS_FILE
group by item
having support >= &GLBL_MBA_MIN_SUPPORT_PCT
;
quit;
run;

*title "&OUTPUT_LIST_FILE";
*proc print data=&OUTPUT_LIST_FILE;
*run;
*title;

data &GLBL_MBA_RESULTS_FILE;
set &GLBL_MBA_RESULTS_FILE &OUTPUT_LIST_FILE (in=new);
Iteration = 1;
run;

%mend first_list_is_trivial;


* ------------------------------------------------------------------------- *
*                    C A L C U L A T E _ S U P P O R T                      *
* ------------------------------------------------------------------------- *
*  This is one of several helper macros for my Market Basket Analysis macro.  *
* ------------------------------------------------------------------------- * ;

%macro calculate_support(ITEMSET_SIZE=, INPUT_CANDIDATE_FILE=, OUTPUT_LIST_FILE=);

%local i;

%do I = 1 %to &ITEMSET_SIZE;
    proc sql noprint;
    create table work.x&i as
    select *
```

```
    from &GLBL_MBA_TRANS_FILE
    where item in
      (select distinct item&i from &INPUT_CANDIDATE_FILE);
    quit;
    run;
%end;


proc sql noprint;
create table &OUTPUT_LIST_FILE as
select
%do I = 1 %to &ITEMSET_SIZE;
    %if (&I > 1) %then %do;
        ,
    %end;
    x&I..item as item&I
%end;
, count(*) as Support_Count
, count(*) / &GLBL_MBA_TRANSACTION_COUNT as support
, &ITEMSET_SIZE as Iteration
from
%do I = 1 %to &ITEMSET_SIZE;
    %if (&I > 1) %then %do;
        ,
    %end;
    work.x&I
%end;
where
%do I = 1 %to &ITEMSET_SIZE - 1;
    %if (&I > 1) %then %do;
        and
    %end;
    x&I..tid = x%sysevalf(&I+1).tid and x&I..item < x%sysevalf(&I+1).item
%end;
group by
%do I = 1 %to &ITEMSET_SIZE;
    %if (&I > 1) %then %do;
        ,
    %end;
    x&I..item
%end;
, Iteration
having support >= &GLBL_MBA_MIN_SUPPORT_PCT;
quit;
run;

*title "&OUTPUT_LIST_FILE";
*proc print data=&OUTPUT_LIST_FILE;
*run;
*title;

data &GLBL_MBA_RESULTS_FILE;
set &GLBL_MBA_RESULTS_FILE &OUTPUT_LIST_FILE;
run;

proc datasets lib=work nolist;
%do I = 1 %to &ITEMSET_SIZE;
    delete x&i;
%end;
run;

%mend calculate_support;
```

```
* ------------------------------------------------------------------------ *
*                 I N I T I A L I Z E _ R E S U L T _ S E T                *
* ------------------------------------------------------------------------ *
*  This is one of several helper macros for my Market Basket Analysis macro.   *
* ------------------------------------------------------------------------ * ;

%macro initialize_result_set();

%local I;

data &GLBL_MBA_RESULTS_FILE;
attrib Iteration   length=8;
%do I = 1 %to &GLBL_MBA_MAX_ITERATIONS;
    %if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
        attrib Item&I   length=$32;
    %end;
    %else %do;
        attrib Item&I   length=8;
    %end;
%end;

attrib Support_Count   length=8;
attrib Support     length=8  format=6.2;

* Remaining columns filled after apriori. ;

%do I = 1 %to &GLBL_MBA_MAX_ITERATIONS-1;
    %if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
        attrib LHS&I    length=$32;
    %end;
    %else %do;
        attrib LHS&I    length=8;
    %end;
%end;

%if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
    attrib RHS     length=$32;
%end;
%else %do;
    attrib RHS     length=8;
%end;

attrib Confidence  length=8  format=6.2;
attrib Lift        length=8  format=6.2;
delete;  * Without this you get an empty row ;
run;

%mend initialize_result_set;



* ------------------------------------------------------------------------ *
*              F I N D _ C O N F I D E N C E _ A N D _ L I F T             *
* ------------------------------------------------------------------------ *
*  This is one of several helper macros for my Market Basket Analysis macro.   *
* ------------------------------------------------------------------------ * ;

%macro find_confidence_and_lift();

%local I;

data Work.With_LHS_RHS (drop = i j k);
```

```
set &GLBL_MBA_RESULTS_FILE;

array items[&GLBL_MBA_MAX_ITERATIONS]
%if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
    $32
%end;
item1-item&GLBL_MBA_MAX_ITERATIONS;

* LHS has one less item than itemset size (iteration) ;
array lhs[%sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1)]
%if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
    $32
%end;
LHS1-LHS%sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1);

if (Iteration = 1) then do;
    LHS1 = item1;
    RHS = item1;
    output;
end;
else do;
    do i = 1 to Iteration;
        k = 0;
        do j = 1 to Iteration;
            * put Iteration= i= j= k=;
            if (i = j) then do;
                RHS = items[j];
            end;
            else do;
                k = k + 1;
                LHS[k] = items[j];
            end;
        end;
        output;  * write one row for each LHS->RHS ;
    end;
end;

run;

* Having populated LHS and RHS, now do confidence and lift ;

proc sql noprint;
create table Work.Almost_Confidence_And_Lift as
select a.Iteration
%do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
, a.LHS&i
%end;
, a.RHS
, a.Support
, a.support / b.support                    as Confidence  format=6.2
, (a.support / b.support) / c.support      as Lift        format=6.2
from Work.with_lhs_rhs as a
  left join &GLBL_MBA_RESULTS_FILE as b
    on (a.iteration - 1) = b.iteration
    %do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
        and a.LHS&i = b.item&i
    %end;
  left join &GLBL_MBA_RESULTS_FILE as c
    on c.Iteration = 1
    and a.RHS = c.item1
order by a.RHS
, a.ITERATION descending
%do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
```

```
    , a.LHS&i
%end;
;
quit;
run;

* Now merge individual lifts back in ;

proc sql noprint;
create table Work.Lifts_Only as
select rhs
, lhs1
, lift
from Work.Almost_Confidence_And_Lift
where iteration = 2;
quit;
run;


proc sql noprint;
create table Work.With_Confidence_And_Lift as
select a.*
%do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
  , t&i..Lift as LHS&i._lift
%end;
from Work.Almost_Confidence_And_Lift as a
  %do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
    left join Work.Lifts_Only as t&i
      on a.rhs = t&i..rhs
      and a.LHS&i = t&i..LHS1
  %end;
order by a.RHS
, a.ITERATION descending
%do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS-1);
    , a.LHS&i
%end;
;
quit;
run;

data &GLBL_MBA_RESULTS_FILE;
set Work.With_Confidence_And_Lift;
run;

proc datasets lib=Work nolist;
delete With_LHS_RHS;
delete Lifts_Only;
delete Almost_Confidence_and_Lift;
delete With_Confidence_and_Lift;
run;

%mend find_confidence_and_lift;


* ------------------------------------------------------------------------ *
*                          W R I T E _ W E B P A G E                       *
* ------------------------------------------------------------------------ *
*  This is one of several helper macros for my Market Basket Analysis macro.   *
* ------------------------------------------------------------------------ * ;

%macro write_webpage(WEBPAGE=, RHS=);

%local I;
```

```
proc sql noprint;
create table work.I as
select distinct rhs as item
from &GLBL_MBA_RESULTS_FILE
%if (&RHS ne ) %then %do;
  where rhs in &RHS
%end;
;
quit;
run;


data _NULL_;
set &GLBL_MBA_RESULTS_FILE end=eof;
%if (&RHS ne ) %then %do;
   where rhs in &RHS;
%end;

file &WEBPAGE lrecl=500;
if (_N_ = 1) then do;
   put "<html>";
   put "<head>";
   put "<title>Market Basket Analysis</title>";
   put "<body>";
   put "<h3 align='center'>Market Basket Analysis<br/>";
   put "Source File: &GLBL_MBA_TRANS_FILE (Obs = &GLBL_MBA_TRANSACTION_COUNT)<br/>";
   percent = put(&GLBL_MBA_MIN_SUPPORT_PCT, percent6.1);
   put "Minimum support: " percent " (n = &GLBL_MBA_MIN_SUPPORT_COUNT)</h3>";
   put "</br>";
   put "<table align='center' width='40%'>";
   put "<tr><td><p><b>Quicklink to Right Hand Side (RHS) variables</b><ul>";
   do while (i_eof = 0);
      set Work.I end=i_eof;
      put "<li><a href='#" item +(-1) "'>" item +(-1) "</a>";
   end;
   put "</ul></p></td></tr>";
   put "<tr bgcolor='yellow'><td align='center'><p>";
   put 'Possible interaction effects shown in yellow.</br>';
   put "Example: Lift(AB%str(&)rarr%str(;)C) %str(&)gt%str(;)
max(Lift(A%str(&)rarr%str(;)C), Lift(B%str(&)rarr%str(;)C))</td></tr>";
   put '</p></td></tr>';
   put "</table>";
end;

lag_rhs = lag(rhs);
if (rhs ne lag_rhs) then do;

   if (_n_ > 1) then do;
      put "</td></tr>";
      put "</table>";
   end;

   put "<a id='" rhs +(-1) "'</a>";
   put "<br/><br/>";
   put "<table align='center' width='90%' border='1' cellpadding='2'
cellspacing='2'>";

   put "<caption><font color='blue'><b>RHS: " rhs "<b></font></caption>";

   put "<tr>";

   %do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1);
```

```
      put "<th>LHS &i</th>";
   %end;

   put "<th width='100'>Support</th>";
   put "<th width='100'>Confidence</th>";
   put "<th width='100'>Lift</th>";
   put "</tr>";
end;

if (Lift > max(0
   %do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1);
      , LHS&i._Lift
   %end;
)) then do;
   bigLift = 1;
end;
else do;
   bigLift = 0;
end;

if (bigLift = 1) then do;
   put "<tr bgcolor='yellow'>";
end;
else do;
   put "<tr>";
end;

%do i = 1 %to %sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1);

   put "<td>";

   %if (&GLBL_ITEM_ID_IS_STRING = "Y") %then %do;
       if (LHS&i = "") then do;
          put ' ';
       end;
       else do;
          put LHS&i;
          if (bigLift = 1) then do;
             put " (" LHS&i._Lift +(-1) ")";
          end;
       end;
   %end;
   %else %do;
       if (LHS&i = .)  then o;
          put ' ';
       end;
       else do;
          put LHS&i;
          if (bigLift = 1) then do;
             put " (" LHS&i._Lift +(-1) ")";
          end;
       end;
   %end;
   * put "</td>";  * </td> optional so omitting it to get a smaller file ;
%end;

* </td> optional so omitting it to get a smaller file ;
put "<td align='center'>" Support;
put "<td align='center'>" Confidence;
put "<td align='center'>" Lift;
put "</tr>";

if (eof) then do;
```

```
   put "</td></tr>";
   put "</table>";
   dt = put(today(), weekdate29.);
   tm = put(time(), tod5.);
   put "<p align='center'>" dt " at " tm "</p>";
   put "</body>";
   put "</html>";
end;

run;

%mend write_webpage;


* ------------------------------------------------------------------------- *
*                                   M B A                                   *
*              Copyright (c) 2013 by Bill Qualls, First Analytics           *
* ------------------------------------------------------------------------- *
*  This is the main macro for my Market Basket Analysis.                    *
* ------------------------------------------------------------------------- *
*  Inputs:                                                                  *
*    TRANS_FILE - Name of a file containing a distinct transaction ID       *
*       and item ID in each row.                                            *
*    ITEM_ID_IS_STRING - Is item ID a string field? For example, "Y".       *
*    MAXIMUM_ITERATIONS - How many iterations of the macro will be run.     *
*       For example, to get {2 3 5 7} in RESULTS_FILE, use 4.               *
*    MINIMUM_SUPPORT - Minimum number of times an item, or set of items,    *
*       must appear together in the same transaction in the TRANS_FILE.     *
*       If input as an integer, it is support as a count, else if entered   *
*       as a decimal, it is a percent (example: 0.20 = 20%). Used to create *
*       a global macro variable MINIMUM_SUPPORT_COUNT used elsewhere.       *
*    RHS - List of right-hand side variables to be included in the          *
*       analysis. If omitted, include all variables with sufficient support.*
*       Example: RHS=('bologna' 'chips')                                    *
* ------------------------------------------------------------------------- *
*  Outputs:                                                                 *
*    WEBPAGE - Name of HTML file.                                           *
*    RESULTS_FILE - Name of the *final* SAS results file.                   *
* ------------------------------------------------------------------------- * ;

%macro mba(TRANS_FILE=, ITEM_ID_IS_STRING=, MAXIMUM_ITERATIONS=,
    MINIMUM_SUPPORT=, RHS=, RESULTS_FILE=, WEBPAGE=);

%local I;

%global GLBL_MBA_MAX_ITERATIONS;
%let GLBL_MBA_MAX_ITERATIONS = &MAXIMUM_ITERATIONS;

%global GLBL_ITEM_ID_IS_STRING;
%let GLBL_ITEM_ID_IS_STRING = &ITEM_ID_IS_STRING;

%global GLBL_MBA_TRANS_FILE;
%let GLBL_MBA_TRANS_FILE = &TRANS_FILE;

%global GLBL_MBA_RESULTS_FILE;
%let GLBL_MBA_RESULTS_FILE = &RESULTS_FILE;

%global GLBL_MBA_TRANSACTION_COUNT;
proc sql noprint;
select count(distinct tid)
into :GLBL_MBA_TRANSACTION_COUNT
from &GLBL_MBA_TRANS_FILE;
quit;
```

```
run;

%put GLBL_MBA_TRANSACTION_COUNT = &GLBL_MBA_TRANSACTION_COUNT;

%global GLBL_MBA_MIN_SUPPORT_PCT;
%global GLBL_MBA_MIN_SUPPORT_COUNT;

%if (&MINIMUM_SUPPORT >= 1) %then %do;
    %let GLBL_MBA_MIN_SUPPORT_COUNT = %sysevalf(&MINIMUM_SUPPORT);
    %let GLBL_MBA_MIN_SUPPORT_PCT = %sysevalf(&MINIMUM_SUPPORT /
&GLBL_MBA_TRANSACTION_COUNT);
%end;
%else %do;
    %let GLBL_MBA_MIN_SUPPORT_COUNT =
%sysfunc(round(%sysevalf(&GLBL_MBA_TRANSACTION_COUNT * &MINIMUM_SUPPORT)));
    %let GLBL_MBA_MIN_SUPPORT_PCT = %sysevalf(&MINIMUM_SUPPORT);
%end;

%put GLBL_MBA_MIN_SUPPORT_COUNT = &GLBL_MBA_MIN_SUPPORT_COUNT;
%put GLBL_MBA_MAX_SUPPORT_PCT = &GLBL_MBA_MIN_SUPPORT_PCT;

%initialize_result_set();

%first_list_is_trivial(OUTPUT_LIST_FILE=Work.L1);

%let DONE = N;
%let MAX_ITERS = %sysevalf(&GLBL_MBA_MAX_ITERATIONS - 1);
%let i = 0;
%do %while (&i < &MAX_ITERS and &DONE = N);
    %let i = %sysevalf(&i + 1);
    %select_distinct_items(CANDIDATE_LIST=work.L&I, ITEMS_EACH_ROW=&I);
    %if (&ITEM_COUNT <= &i) %then %do;
        proc datasets lib=Work nolist;
            delete L&I;
        run;
        %let DONE = Y;
    %end;
    %else %do;
        %create_candidate_set(ITEM_LIST=&ITEM_LIST
            , ITEM_COUNT=&ITEM_COUNT
            , ITEMSET_SIZE=%sysevalf(&I+1)
            , OUTPUT_CANDIDATE_FILE=Work.C%sysevalf(&I+1));

        %calculate_support(ITEMSET_SIZE=%sysevalf(&I+1)
            , INPUT_CANDIDATE_FILE=Work.C%sysevalf(&I+1)
            , OUTPUT_LIST_FILE=Work.L%sysevalf(&I+1));

        proc datasets lib=Work nolist;
            delete L&I;
            delete C%sysevalf(&I+1);
        run;
    %end;
%end;

%find_confidence_and_lift();

%write_webpage(WEBPAGE=&WEBPAGE, RHS=&RHS);

%mend mba;


%let MYLIB = C:\Users\Owner\Desktop\MWSUG;
libname Perm "&MYLIB";
```

```
data work.sales;
input tid item $;
datalines;
1 bread
1 butter
1 eggs
1 milk
2 bologna
2 bread
2 cheese
2 chips
2 mayo
2 soda
3 bananas
3 bread
3 butter
3 cheese
3 oranges
4 buns
4 chips
4 hotdogs
4 mustard
4 soda
5 buns
5 chips
5 hotdogs
5 mustard
5 pickles
5 soda
6 bread
6 butter
6 cereal
6 eggs
6 milk
7 bananas
7 cereal
7 eggs
7 milk
7 oranges
8 bologna
8 bread
8 buns
8 cheese
8 chips
8 hotdogs
8 mayo
8 mustard
8 soda
9 bananas
9 bologna
9 bread
9 cheese
9 milk
9 oranges
9 soda
10 bread
10 butter
10 cereal
10 eggs
10 milk
11 bananas
```

```
11 chips
11 soda
12 bread
12 butter
12 eggs
12 milk
12 oranges
13 bananas
13 bologna
13 bread
13 cheese
13 mayo
13 mustard
14 bread
14 cereal
14 eggs
14 milk
15 bologna
15 bread
15 cheese
15 chips
15 mayo
15 mustard
15 soda
16 bread
16 butter
16 eggs
16 milk
16 oranges
17 buns
17 chips
17 hotdogs
17 soda
18 buns
18 cheese
18 chips
18 hotdogs
18 mustard
18 soda
19 chips
19 pickles
19 soda
20 bologna
20 bread
20 cheese
20 chips
20 mayo
20 mustard
20 soda
;
run;


%mba(TRANS_FILE=work.sales
, ITEM_ID_IS_STRING="Y"
, MAXIMUM_ITERATIONS=3
, MINIMUM_SUPPORT=0.2
, RHS=('bologna' 'bread')
, RESULTS_FILE=Perm.Results_Groc
, WEBPAGE="C:\Users\Owner\Desktop\MWSUG\mba_groc.html"
);
run;
```

SAMPLE HTML OUTPUT

Market Basket Analysis
Source File: work.sales (Obs = 20)
Minimum support: 20% (n = 4)

Quicklink to Right Hand Side (RHS) variables

- bologna
- bread

Possible interaction effects shown in yellow.
Example: Lift(AB→C) > max(Lift(A→C), Lift(B→C))

RHS: bologna

LHS 1	LHS 2	Support	Confidence	Lift
bread (1.54)	cheese (2.50)	0.30	0.86	2.86
bread (1.54)	chips (1.33)	0.20	1.00	3.33
bread	mayo	0.25	1.00	3.33
bread (1.54)	mustard (1.90)	0.20	1.00	3.33
bread (1.54)	soda (1.52)	0.25	1.00	3.33
cheese (2.50)	chips (1.33)	0.20	0.80	2.67
cheese	mayo	0.25	1.00	3.33
cheese (2.50)	mustard (1.90)	0.20	0.80	2.67
cheese (2.50)	soda (1.52)	0.25	0.83	2.78
chips	mayo	0.20	1.00	3.33
chips	soda	0.20	0.40	1.33
mayo	mustard	0.20	1.00	3.33
mayo	soda	0.20	1.00	3.33
bread		0.30	0.46	1.54
cheese		0.30	0.75	2.50
chips		0.20	0.40	1.33
mayo		0.25	1.00	3.33
mustard		0.20	0.57	1.90
soda		0.25	0.45	1.52

bologna		0.30	.	.

RHS: bread				
LHS 1	**LHS 2**	**Support**	**Confidence**	**Lift**
bologna	cheese	0.30	1.00	1.54
bologna	chips	0.20	1.00	1.54
bologna	mayo	0.25	1.00	1.54
bologna	mustard	0.20	1.00	1.54
bologna	soda	0.25	1.00	1.54
butter	eggs	0.25	1.00	1.54
butter	milk	0.25	1.00	1.54
cheese	chips	0.20	0.80	1.23
cheese	mayo	0.25	1.00	1.54
cheese	mustard	0.20	0.80	1.23
cheese	soda	0.25	0.83	1.28
chips	mayo	0.20	1.00	1.54
chips	soda	0.20	0.40	0.62
eggs	milk	0.30	0.86	1.32
mayo	mustard	0.20	1.00	1.54
mayo	soda	0.20	1.00	1.54
bologna		0.30	1.00	1.54
butter		0.30	1.00	1.54
cheese		0.35	0.88	1.35
chips		0.20	0.40	0.62
eggs		0.30	0.86	1.32
mayo		0.25	1.00	1.54
milk		0.35	0.88	1.35
mustard		0.20	0.57	0.88
oranges		0.20	0.80	1.23
soda		0.25	0.45	0.70
bread		0.65	.	.

Saturday, July 20, 2013 at 17:40