Chapter 7 Code for Market Basket Analysis (MBA)

/\*Please use data MBA\_Master.xlsx\*/

FILENAME REFFILE '/folders/myfolders/MBA\_Master.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT=WORK.WIDE;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=WORK.WIDE; RUN;

proc transpose data=wide out=long (rename=(col1=PRODUCTS) drop=\_label\_);

by custid;

var pr1-pr8;

run;

proc sql;

create table product\_customer\_count as

select products, count(distinct(custid)) as ANALYSIS\_UNIT\_FREQ

from long

where products ne ""

group by 1

;

quit;

proc sql noprint;

select count(distinct(products)) into: product\_count from long where products ne "";

select count(distinct(custid)) into: population from long;

select distinct products into :product\_1-:product\_%trim(&product\_count) from product\_customer\_count where products ne "" ;

select distinct analysis\_unit\_freq into :analysis\_unit\_freq\_1-:analysis\_unit\_freq\_%trim(&product\_count) from product\_customer\_count where products ne "" order by products;

quit;

%put &product\_count;

%put &product\_2;

%put &analysis\_unit\_freq\_10;

%macro product\_tot;

%do i = 1 %to &product\_count;

proc sql;

create table product\_tot\_&i as

select products, count(distinct(custid)) as FREQ\_CO\_OCCUR

from

(select custid, products

from long

where custid in (select custid from long where products eq "&&product\_&i"))

where products ne ""

group by 1

order by 1

;

quit;

%end;

%mend;

%product\_tot;

%put product is &product\_1;

%macro metrics;

%do i = 1 %to &product\_count;

proc sql;

create table metric\_&i as

select distinct "&&product\_&i" as LHS, a.PRODUCTS as RHS, a.ANALYSIS\_UNIT\_FREQ, b.freq\_co\_occur, (b.FREQ\_CO\_OCCUR/&&analysis\_unit\_freq\_&i)\*100 as CONFIDENCE format 5.2,

(b.FREQ\_CO\_OCCUR/&population)\*100 as SUPPORT format 5.2, (a.ANALYSIS\_UNIT\_FREQ/&population)\*100 as EXPECTED\_CONFIDENCE format 5.2,

calculated CONFIDENCE/calculated EXPECTED\_CONFIDENCE as LIFT format 5.2, (FREQ\_CO\_OCCUR\*(calculated LIFT-1)\*\*2)\*((calculated SUPPORT/100)\*(calculated CONFIDENCE/100))

/

((calculated CONFIDENCE/100 - calculated SUPPORT/100)\*(calculated LIFT - calculated CONFIDENCE/100)) as CHISQ format 5.2,

1 - Probchi(calculated CHISQ,1) as P format 5.4

from product\_customer\_count as a left join product\_tot\_&i as b

on a.products=b.products

order by calculated P

;

quit;

%end;

%mend;

%metrics;

proc sql noprint;

select memname into :datasets separated by ' '

from dictionary.tables

where libname = "WORK" and memname like "METRIC%";

quit;

data summary;

format LHS $50.;

set &datasets;

if LHS ne RHS;

run;

data relevant (drop=analysis\_unit\_freq);

set summary;

if freq\_co\_occur ge 50;

if P le 0.05;

if confidence ge 60;

if lift gt 1;

run;

proc sort data=relevant;

by descending lift;

run;

proc print;

run;

proc print data=product\_tot\_1;

run;

data condense;

set relevant;

if lhs not in ("Personal\_Current\_Account", "Savings\_Account", "Credit\_Card", "Locker", "Personal\_Loans");

run;

proc print;

run;

data one;

input id $ product1 $50.;

datalines;

1 BTL\_Mortgage|

2 Business\_Current\_Account|

3 Credit\_Card|

4 Currency\_Services|

5 Insurance|

6 Locker|

7 Personal\_Current\_Account|

8 Personal\_Loans|

9 Premium\_Current\_Account|

10 Residential\_Mortgage|

11 Savings\_Account|

12 Trading\_Account|

;

data two;

input id $ product2 $50.;

datalines;

1 BTL\_Mortgage|

2 Business\_Current\_Account|

3 Credit\_Card|

4 Currency\_Services|

5 Insurance|

6 Locker|

7 Personal\_Current\_Account|

8 Personal\_Loans|

9 Premium\_Current\_Account|

10 Residential\_Mortgage|

11 Savings\_Account|

12 Trading\_Account|

;

data three;

input id $ product3 $50.;

datalines;

1 BTL\_Mortgage

2 Business\_Current\_Account

3 Credit\_Card

4 Currency\_Services

5 Insurance

6 Locker

7 Personal\_Current\_Account

8 Personal\_Loans

9 Premium\_Current\_Account

10 Residential\_Mortgage

11 Savings\_Account

12 Trading\_Account

;

run;

data stage1;

set one;

do i = 1 to n;

set two point=i nobs=n;

output;

end;

run;

data stage2;

set stage1;

do i = 1 to n;

set three point=i nobs=n;

output;

end;

run;

data matrix (drop = lhand1 lhand2);

set stage2 (drop = id);

if product1 ne product2;

if product1 ne product3;

if product2 ne product3;

combo=compress(product1||product2||product3);

lhand1=scan(combo,1);

lhand2=scan(combo,2);

lhand=compress(lhand1||"|"||lhand2);

run;

proc sql noprint;

select count(distinct(combo)) into:combo\_count from matrix;

select distinct combo into:combo\_1 -:combo\_%trim(&combo\_count) from matrix;

select count(distinct(lhand)) into:lhand\_count from matrix;

select distinct lhand into:lhand\_1 -:lhand\_%trim(&lhand\_count) from matrix;

quit;

%put lhand count is &lhand\_count;

%macro combo\_find;

%do i = 1 %to &combo\_count;

proc sql;

create table combos\_main\_&i as

select a.\*, compress(pr1||"|"||pr2||"|"||pr3||"|"||pr4||"|"||pr5||"|"||pr6||"|"||pr7||"|"||pr8) as combo, "&&combo\_&i" as pattern\_found,

case when calculated combo contains "&&combo\_&i" then 1 else 0 end as combo\_count

from wide as a

;

create table combos\_sum\_&i as

select pattern\_found, sum(combo\_count) as freq\_co\_occur

from combos\_main\_&i

group by 1

having freq\_co\_occur ge 1

order by calculated freq\_co\_occur desc

;

quit;

%end;

%mend;

%combo\_find;

%macro lhand\_find;

%do i = 1 %to &lhand\_count;

proc sql;

create table lhand\_main\_&i as

select a.\*, compress(pr1||"|"||pr2||"|"||pr3||"|"||pr4||"|"||pr5||"|"||pr6||"|"||pr7||"|"||pr8) as combo,

"&&lhand\_&i" as lhand,

case when calculated combo contains "&&lhand\_&i" then 1 else 0 end as lhand\_count

from wide as a

;

create table lhand\_sum\_&i as

select lhand, sum(lhand\_count) as lhand\_co\_occur

from lhand\_main\_&i

group by 1

having lhand\_co\_occur ge 1

order by calculated lhand\_co\_occur desc

;

quit;

%end;

%mend;

%lhand\_find;

proc sql noprint;

select memname into:combosets separated by ' '

from dictionary.tables

where libname eq "WORK" and memname like "COMBOS\_SUM\_%"

;

quit;

data combos\_main (drop = lhand1 lhand2);

format pattern\_found $200.;

set &combosets;

lhand1=scan(pattern\_found,1);

lhand2=scan(pattern\_found,2);

lhand=compress(lhand1||"|"||lhand2);

rhand=scan(pattern\_found,-1);

run;

proc sort data=combos\_main;

by descending freq\_co\_occur;

run;

proc sql noprint;

select memname into:lhandsets separated by ' '

from dictionary.tables

where libname eq "WORK" and memname like "LHAND\_SUM\_%"

;

quit;

data lhand\_main;

format lhand $200.;

set &lhandsets;

run;

proc sql;

create table metric\_three\_way as

select distinct a.LHAND as LHS, a.RHAND as RHS, c.lhand\_co\_occur as ANALYSIS\_UNIT\_FREQ, a.freq\_co\_occur, (a.FREQ\_CO\_OCCUR/c.lhand\_co\_occur)\*100 as CONFIDENCE format 5.2,

(a.FREQ\_CO\_OCCUR/&population)\*100 as SUPPORT format 5.2, (d.analysis\_unit\_freq/&population)\*100 as EXPECTED\_CONFIDENCE format 5.2,

calculated CONFIDENCE/calculated EXPECTED\_CONFIDENCE as LIFT format 5.2, (a.FREQ\_CO\_OCCUR\*(calculated LIFT-1)\*\*2)\*((calculated SUPPORT/100)\*(calculated CONFIDENCE/100))

/

((calculated CONFIDENCE/100 - calculated SUPPORT/100)\*(calculated LIFT - calculated CONFIDENCE/100)) as CHISQ format 5.2,

1 - Probchi(calculated CHISQ,1) as P format 5.4

from combos\_main as a left join product\_customer\_count as b

on a.lhand=b.products

left join lhand\_main as c

on a.lhand=c.lhand

left join product\_customer\_count as d

on a.rhand=d.products

order by calculated P, calculated confidence desc, calculated expected\_confidence desc

;

quit;

data relevant\_three\_way;

set metric\_three\_way;

if P le 0.05;

if confidence ge 40;

if lift gt 1;

if freq\_co\_occur ge 15;

run;

proc sort data=relevant\_three\_way;

by descending lift;

run;

proc print data=metric\_three\_way;

run;

/\*after removal of the restrictions on the rule\*/

data relevant\_three\_way;

set metric\_three\_way;

if confidence ge 40;

if lift gt 1;

if freq\_co\_occur ge 25;

run;

proc print data=relevant\_three\_way (drop=chisq p);

run;

FILENAME REFFILE '/folders/myfolders/market basket analysis.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT=WORK.CLUST REPLACE;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=WORK.CLUST; RUN;

proc sql;

create table a0 as

select custid, count(\*) as row

from clust

group by 1

order by calculated row desc

;

quit;

proc varclus data=clust;

var pr1-pr8;

run;

proc cluster data=clust method=ward std ccc pseudo out=tree PLOTS(MAXPOINTS=300);

id custid;

var pr1-pr8;

run;