```
put
'/************************************
*/';
put;
put 'DATA SCORING;        /** Modify
**/';
put ' SET ScoringDataset; /** Modify
**/';
put;
put
'/************************************
*/';
put
'/************************************
*/';
end;

/* print the dataset RulesDS */
%if &IntOpt=1 %then xPoints=int(Points);
%else xPoints=Points; ;

if VarName="_BasePoints_" then do;
put
'/************************************
*/';
put "/* Base Points   */";
put
'/************************************
*/';
put "Points=" xPoints ";";
                            end;
 else do;
   if first.VarName then do;
put
'/************************************
*/';
put "/* Variable : " VarName "
*****/";
put
```

```
'/*************************************
*/';
                    end;
    value= "  THEN  Points=Points
+("||compress(xPoints)||");";
    /* The rule */
    if VarType=1 then  do;/* continuous */
if first.VarName then  cond='IF
'||compress(VarName)||
   ' LE  ('||compress(UL)  || ') ';
else if last.VarName then cond='IF
'||compress(VarName)||
  ' GT  ('|| compress(LL)||')';
    else  cond='IF '||compress(VarName)||' GT
('||
    compress(LL)||') AND
'||compress(VarName)||
    ' LE  (' ||compress(UL)  || ') ';
                       end;
    else if VarType=2 then /* nominal string */
cond='IF '||compress(VarName)||' =
'|| quote(compress(Category)) ;
else /* nominal numeric */
cond='IF '||compress(VarName)||' = ('||
compress(N_Category)||') ';
put "       " cond value;
 end;
 if _N_=Nx then do;
put 'RUN;';
put;
put '/*****END OF SCORING DATA
STEP *******/';
put
'/*************************************
*/';
end;
run;
%mend;
```

C.52 %TRollup

```
/*******************************/
/* The rollup macro */
/*******************************/
%macro TRollup( TDS, IDVar, TimeVar,
TypeVar, Nchars, Value, RDS);

/* Sort using the ID, time, and type variables
*/
proc sort data=&TDS;
by &IDVar &TimeVar &TypeVar;
run;
/* Accumulate the values over time in _TOT */
data _Temp1;
retain _TOT 0;
set &TDS;
by &IDVar &TimeVar &TypeVar;
if first.&TypeVar then _TOT=0;
_TOT = _TOT + &Value;
if last.&TypeVar  then output;
drop &Value;
   run;

proc sort data=_Temp1; by &IDVar &TypeVar;
run;

/* Extract the categories of the TypeVar and
store them in  macro
variables. We use PROC  FREQ to find all non-
missing categories */
proc freq data =_Temp1 noprint;
tables &TypeVar /out=_Types ;
run;
data _null_;
 set _Types nobs=Ncount;
 if &typeVar ne '' then call
symput('Cat_'||left(_n_), &TypeVar);
if _n_=Ncount  then call
```

```
symput('N', Ncount);
run;
%do i=1 %to &N;
proc transpose data =_Temp1 out=_R_&i
     prefix=%substr(&&Cat_&i, 1, &Nchars)_;
by &IDVar &TypeVar;
ID &TimeVar ;
var _TOT ;
where &TypeVar="&&Cat_&i";
run;
%end;
/* Finally, assemble all these files by the ID
variable */
data &RDS;
 merge
 %do i=1 %to &N;
 _R_&i
 %end ; ;
 by &IDVar;
 drop &TypeVar _Name_;
run;
/* clean workspace */
proc datasets library=work nodetails nolist;
delete _Temp1 _Types
%do i=1 %to &N;
    _R_&i    %end; ;  ;
 run;
 quit;

%mend;
```

C.53 %VarMode

```
/*******************************/
/* Macro VarMode */
/*******************************/
%macro VarMode(TransDS, IDVar, XVar,
OutDS);
```

```
/* Calculation of the mode of a variable Xvar
from a transaction
dataset using the classic implementation in
ANSI SQL */

proc sql noprint;
create table &OutDS as
SELECT &IDVar , MIN( &XVar )
AS mode
FROM (
 SELECT &IDVar,  &XVar
 FROM &TransDS p1
 GROUP BY &IDVar, &XVar
 HAVING COUNT( * ) =
       (SELECT MAX( CNT )
        FROM (SELECT COUNT( * ) AS CNT
              FROM &TransDS p2
              WHERE p2.&IDVar= p1.&IDVar
              GROUP BY p2.&XVar
              ) AS p3
        )
) AS p
GROUP BY p.&IDVar;
quit;
%mend;
```