

/* Decompose variables which contains additional codes
DECEMBER 2007

Programmer: Adrien FRANCAIS
Statistical Enginner

Albert Bonniot Institute
Team 11-Outcome of cancer and critical illnesses
UJF-INSERM U823
38706 LA TRONCHE CEDEX
FRANCE
04 76 54 94 00

MACRO VARIABLES

TABLE : table which contains the variable to treat
TABLEEND : table with several variables created
VARIABLE : variable in additional codes
NEWVAR : prefix of different variables obtained

Before using this macro, you have to :

- Prepare your database with correct variables
- Compile the macro.
- Execute the macro :

Example to compute on your computer to directly and easily apply the macro:

```
data try;  
input chronadd ;  
cards;  
1  
2  
4  
8  
16  
24  
9  
3  
2  
1  
8  
8  
16  
30  
31  
4  
5  
6  
7
```

```
9
11
;
```

```
run;

%additionalcode(table=try,tableend=sort,variable=chronadd,newvar=chron);
```

You can see LOG screen for information of algorithm.

If you use this macro for work to be published please use the following citation:

Francais Adrien (2007).

Decompose variables which contains additional codes

```
*/
```

```
%macro additionalcode(table=,tableend=,variable=,newvar=);
%let error=0;
/*some errors messages*/
%if &table = %then %do;
    %put %upcase("ERROR: You didn't give a valid name of table : Please enter the name
of the table");
    %let error=1;
%end;
%if &tableend = %then %do;
    %put %upcase("ERROR: You didn't give a valid name of table for results: Please
enter the name of the table");
    %let error=1;
%end;
%if &variable = %then %do;
    %put %upcase("ERROR: You didn't give variable which contains additional code :
Please enter the correct quantitative variable");
    %let error=1;
%end;
%if &newvar = %then %do;
    %put %upcase("ERROR: You didn't give the prefix for variables obtained after
decomposition: Please enter a name near from the &variable");
    %let error=1;
%end;

%if &error = 1 %then %do;
    %put "PLEASE CORRECT ERROR(S) DETAILED ABOVE";
%end;

/*if there is no error then we start the macro*/
%if &error ne 1 %then %do;

    /*we recover the maximal value in the variable*/
```

```

ods output extremeobs=ext(keep=high);
proc univariate data=&table;var &variable;run;
data sffds;set ext;h=1;run; data fdsfs;set sffds;by h;if last.h;call
symput('taille_maxi',high);run;
%put %upcase("the maximal value in the variable was &taille_maxi");

/*part to create list of additional codes*/
data &tableend;
set &table;
%put %upcase ();
%put %upcase("we create the list of additional codes");
%let liste=;%let precedent=1;%let inter=;%let i=0;%let maxim=100000;
%do %while (&inter<&maxim);
    %let i = %eval(&i+1);
    %let inter=%eval(2*&precedent);
    %let liste = &liste &inter;
    %let precedent=%eval(&inter);
%end;
%let listefinal=1 &liste;
%put &listefinal;

/*we recover the maximal number of columns to create global variable SIZE*/
%put %upcase ();
%put %upcase("we recover the maximal number of columns to create, variable
SIZE");
%local varlist i var; %global size;
%let varlist = &listefinal;
%let i=0; %let end=0;
%do %while(&end=0);
    %let i = %eval(&i+1);
    %let var = %scan(&varlist,&i,%str( ));
    %if &var > &taille_maxi %then %do;
        %let end=1; %let size=%eval(&i-1);
    %end;
%end;
%put %upcase ();
%put %upcase("The number of columns created is &size with the prefix &newvar");

/*we declare the newvar with an array statement*/
array &newvar(&size) $50.;
/*we replace missing values by zero else there is a bug */
if &variable=. then &variable=0;
/*some declarations of variables*/
%put %upcase ();
%put %upcase("We declare some variables");
&variable.bis=&variable;st=&variable.bis;
f=1;sum=0;t=0;

```

```

/*if the variable is null*/
if &variable=0 then &newvar(1)=0;

/*do while the variable is not completely decomposed*/
do while (sum ne &variable);
    /*if the code is in the list of additional code, then we add this value to
newvar(j)
and we recover the value which remains to test*/
    if &variable.bis in (&listefinal) then do;
        &newvar(f)=&variable.bis;
        &variable.bis=st-&variable.bis;
        st=&variable.bis;
        sum=sum+&newvar(f);
        f=f+1;
    end;
    /*if the code is not in the list of additional code, we continue to look for the
right value*/
    if &variable.bis not in (&listefinal) then do;
        &variable.bis=&variable.bis-1;
    end;
end;
/*we remove local variables*/
drop &variable.bis &variable f st sum t;
%put %upcase ();
%put %upcase ("We decomposed the variable &variable in &size variables named
with the prefix '&newvar'");
%put %upcase ("You can see table of results in the Explorator section, variables
&newvar are at the end of this table");
run;
%end;
%mend;

```