A Macro for Efficient and Flexible Data Smoothing

John Cucka, Millward Brown International, Naperville, IL

ABSTRACT

This macro provides the user with a simple and flexible method for adding data smoothing to their analytical tool box. The user can specify the input data set, the variable to be smoothed, the period of smoothing, and optionally to use center-weighting in the smoothing process. The macro also includes facilities for by-group type processing of stacked groups of data in a single variable.

INTRODUCTION

Data smoothing intends to draw out the underlying trend from “noisy” data. The strength of this technique lies in the flexibility allowed in achieving this goal. Data smoothing does not require linear or even quadratic modeling of the data. Instead, the results are based on averaging over a number of nearby points, where the number of points and weighting system for those points is user-definable. As such, data smoothing is flexible not only in fitting the data, but also in its applicability to a wide variety of data.

To make the smoothing process as efficient as possible, a single DATA step, requiring only a single pass at the data, contains the smoothing computations. All other steps in the macro perform input and output data handling, or define the macro itself. What follows is a description of that code, which is included at the end of this paper.

Macro Definitions

The macro has 5 required definitions:

- The name of the input data set
- The name of the variable to smooth
- The name of the variable identifying the dimension to smooth over (for example, a time-indicator variable)
- The length of the smoothing period
- An indicator for activating or de-activating center-weighting of the smoothing process

In addition, there are 3 macro variable definitions that are optional:

- A format to use for the variable indicating the dimension to smooth over (for example, DATE7. - don’t forget the period!)
- The name of the variable defining by-groups
- The maximum length of the by-group variable, if it is character.

Data Preparation

There are two preparation stages before performing the smoothing. The first of these defines macro variable values from the information provided by the user. The second step prepares the input data set.

A _NULL_ DATA step computes some useful values from the information defined in the macro call. First, it determines if the period for smoothing is odd or even, and then adjusts the macro call for either situation. If the period of smoothing is even, these adjustments are: 1) increase the period of smoothing by one, making it odd, and 2) define the macro variable ODD so that the endpoints are weighted by one-half during smoothing. Second, it determines the center point of the period of smoothing and the length of the tails on either side. Third, it determines the sum of the weights in the case of doing center-weighting. Since center-weighting uses a triangular weighting scheme, the sum of the weights is equal to the square of the center-point of the smoothing period. Finally, it writes out the necessary values to macro variables for later use.

PROC SORT prepares the input data set. In addition to being sorted properly, the data is output to a temporary data set including only those variables needed for the smoothing process. Table 1 shows a partial listing of an input data set.

Table 1 Partial Listing of Input Data Set

```
OBS      WEEK       TU

  1    27JUN93    5602
  2    04JUL93    5892
  3    11JUL93    5475
  4    18JUL93    5886
  5    25JUL93    8297
  6    01AUG93    6259
  7    08AUG93    7561
            .
            .
            .
130    17DEC95    2207
131    24DEC95    2071
132    31DEC95    2029
133    07JAN96    1595
134    14JAN96    1274
135    21JAN96    1436
136    28JAN96    1614
```

SMOOTHING ENGINE

A single DATA step runs the entire smoothing process. Here is an overview of the computations used in the DATA step, and how they relate to the smoothing process.

First, the DATA step computes the lagged values of the variable to be smoothed, out to the period of smoothing. In other words, if the period of smoothing is 5, then 5 variables are created, each corresponding to the LAG1-LAG5 values of the variable being smoothed. At the same time, the lagged variables are weighted, if necessary. Weighting is necessary when either center-weighting is on (utilizing a triangular weighting pattern) or an even-period of smoothing is selected (where either end-point is down-weighted by one-half). Next, the weighted average of all the lagged values is computed. Since the average for the lagged values is actually the smoothed value for the mid-point of the lags, new indicator variables are necessary. These new variables, representing the id-variable and if necessary the by-variable, are given values corresponding to the center-point of the lags included on that record. So, using the example of a 5-point smoothing, the new values of the id-variable and by-variable are from the LAG3 record, as that would be the center of the 5-point lag.

Next, we take a more detailed look at the code used at each stage of this process.

Initialize Values

In the DATA statement, the output data set is defined including only the necessary variables for output: the id-variable, the by-variable, if applicable, and the new variable SMOOTH. The id-variable and by-variable are renamed for merging later with the input data set.

Next, the input data is SET in. In the case where the user selects by-group processing, three statements are added: 1) The length of the new by-group variable is established according to user specifications, 2) The data set is SET in BY the by-variable so that the computations may be done separately by group, and 3) The variable NSTART is defined, which is used to indicate when there is sufficient data for variable smoothing within each by-group.

Table 2 Partial Listing of Work Data Set (Period=5, Center-Weighting=N)

```
OBS       WEEK     TU       NEWID     S0      S1      S2      S3      S4     SMOOTH

  1    27JUN93    5602          .    5602      .       .       .       .         .
  2    04JUL93    5892          .    5892    5602      .       .       .         .
  3    11JUL93    5475          .    5475    5892    5602      .       .         .
  4    18JUL93    5886          .    5886    5475    5892    5602      .         .
  5    25JUL93    8297    11JUL93    8297    5886    5475    5892    5602    6230.4
  6    01AUG93    6259    18JUL93    6259    8297    5886    5475    5892    6361.8
  7    08AUG93    7561    25JUL93    7561    6259    8297    5886    5475    6695.6
                                        .
                                        .
                                        .
130    17DEC95    2207    03DEC95    2207    1733    1835    1804    2088    1933.4
131    24DEC95    2071    10DEC95    2071    2207    1733    1835    1804    1930.0
132    31DEC95    2029    17DEC95    2029    2071    2207    1733    1835    1975.0
133    07JAN96    1595    24DEC95    1595    2029    2071    2207    1733    1927.0
134    14JAN96    1274    31DEC95    1274    1595    2029    2071    2207    1835.2
135    21JAN96    1436    07JAN96    1436    1274    1595    2029    2071    1681.0
136    28JAN96    1614    14JAN96    1614    1436    1274    1595    2029    1589.6
```

Whether or not there is by-group processing, a new id-variable is defined at the proper lagged value (that is, the center-point of the smoothing period). At the end of the DATA step, note that all variables related to the id-variable are formatted, using the format designated by the user.

LAGs and Weighting

In the weighting section, a macro loop generates the code creating the variables containing the *N* lagged values for each record, where *N* is the period of smoothing.

The macro loop over &I is split into two pieces to facilitate center-weighting. The first half of the smoothing period uses increasing weights by adding &I to the weight multiplier as the lags become closer to the center-point. The second half of the smoothing period uses decreasing weights by subtracting &I from the weight multiplier as the lags move away from the center-point. Furthermore, in the case of even-period smoothing, the two end-points are be down-weighted by one-half.

Averaging

In the final section of the DATA step, a conditional first checks for records that have sufficient data for smoothing. In the case of by-group processing, the condition is positive when there is enough data in each group, as indicated by the NSTART variable defined at the top of the DATA step. Otherwise, the condition is positive when there are enough observations in total. When this condition is positive, the values for all the lagged variables defined in the previous section are summed and divided by the proper sum of the weights. In the case of non-center-weighted smoothing, the sum of the weights is equal to the number of points in the smoothing. In the case of center-weighting smoothing, the sum of the weights is equal to the value generated in the _NULL_ DATA step at the top of the macro. Also, if the selected period of smoothing is even, one is subtracted from the sum of the weights.

Lagged values are used for the new by-variable and id-variable. Because there is not sufficient data for smoothing at the end of the series, there will be no records generated with those id-variable values. Table 2 shows the effect of insufficient data on the smoothing estimates of the data shown in Table 1. (The smoothing was generated using a 5-point period of smoothing, without center-weighting.) Notice how the last value for the variable NEWID is actually the third-to-last value for the original id-variable, WEEK.

Output Results

PROC SORT prepares the output data set. Using a WHERE statement, the data is pruned to include only those records for which there was sufficient data for smoothing, specifically removing the records from the beginning of the data series.

In a final DATA step, the output data, including only the id variables and the new variable SMOOTH, is merged onto the original data set. Table 3 shows a partial printout of the output data set from earlier examples.

Table 3 Output Data Set

```
OBS       WEEK     TU     SMOOTH

  1    27JUN93    5602        .
  2    04JUL93    5892        .
  3    11JUL93    5475    6230.4
  4    18JUL93    5886    6361.8
  5    25JUL93    8297    6695.6
  6    01AUG93    6259    6861.4
  7    08AUG93    7561    6837.8
                   .
                   .
                   .
130    17DEC95    2207    1975.0
131    24DEC95    2071    1927.0
132    31DEC95    2029    1835.2
133    07JAN96    1595    1681.0
134    14JAN96    1274    1589.6
135    21JAN96    1436        .
136    28JAN96    1614        .
```

CONCLUSION

As seen here, using simple procedures and a macro interface, a flexible and efficient mechanism for performing data smoothing can be made readily available to data analysts everywhere. It is up to the analysts themselves to carefully consider when data smoothing may be useful or necessary and what parameter settings to use in their smoothing.

SAS, The SAS System, and Base SAS software are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

John Cucka
Senior Statistician
Millward Brown International
1245 E. Diehl Rd.
Naperville, IL 60563-9349
tel: (630) 955-8637
fax: (630) 245-5620
email: johncuc@mbinp.milwrdbrn.com

SMOOTHING MACRO CODE

```
%macro smthloop(
/*** REQUIRED MACRO VARIABLE DEFINITIONS ***/
   indata=trendat, /* indata is the name of the input data set.
                            NOTE:  for output, the variable SMOOTH,
                            containing the smoothed data value, will be
                            added to &indata */
   smthvar=tu,    /* name of variable to smooth */
   idvar=week,     /* name of variable identifying
                           dimension to smooth over */
   smthpnts=53,   /* number of points in smoothing period
                            NOTE: must be an integer */
   ctrwgt=Y,       /* Y or N, Y for center-weighted smoothing */

  /*** OPTIONAL MACRO VARIABLE DEFINITIONS ***/
   idfmt=date7.,    /* a format for the id variable */

  /*** OPTIONAL - FOR BY-GROUP PROCESSING ***/
   byvar=market,   /* name of variable identifying separate
                             groups in the input data set */
   bylen=3,       /* indicate max length for byvar (if character) */
);


/*** GENERATE KEY NUMBERS IN POINT RANGE ***/
 data _null_;
  odd=‘Y’;      /* indicates odd or even period for smoothing */
  points=&smthpnts;

/* make modifications for even-numbered period of smoothing */
  if mod(points,2)=0 then do;
   points=points+1;
   odd=‘N’;
   call symput ('smthpnts',left(trim(points)));
  end;

/* calculate critical values from period of smoothing */
  tails =floor(points/2);    /* how many points on either side */
  pivot =floor(points/2)+1;  /* the center of smoothing period */
  sumwt =pivot**2;       /* sum of the weights in center-weighting */

/* write out all values to macro variables */
  call symput ('tails',left(trim(tails)));
  call symput ('pivot',left(trim(pivot)));
  call symput ('sumwt',left(trim(sumwt)));
  call symput ('odd',left(trim(odd)));
 run;


/*** PREPARE INPUT DATA ***/
/* sort input data and keep only the necessary variables */
  proc sort data=&indata out=smthdat(keep=&idvar &smthvar
            %if %length(&byvar)>0 %then %do;
             &byvar
            %end;
           );
   by %if %length(&byvar)>0 %then %do;
       &byvar
      %end;
       &idvar;
  run;


/*** SMOOTHING ENGINE ***/
data smth(keep=newid smooth       /* keep only necessary vars */
              %if %length(&byvar)>0 %then %do;
               newby
              %end;
          rename=(newid=&idvar   /* rename by-vars for merging */
                %if %length(&byvar)>0 %then %do;
                 newby=&byvar
                %end;
           ));

%if %length(&bylen)>0 %then %do;
  length newby $ &bylen;     /* define new variable for by-var */
 %end;

set smthdat;                    /* set in original data */

 %if %length(&byvar)>0 %then %do;
  by &byvar;
  /* the value for the by-group at the center of smoothing */
   lag_by =lag&tails(&byvar);
  if first.&byvar then do;
  /* indicator of first obs in each by-group with sufficient data */
   nstart= _n_ + &smthpnts - 1;
  end;
  retain nstart;
 %end;
```

```
  /* set the value for the new id-var at the center of smoothing */
  lag_id =lag&tails(&idvar);



* WEIGHTING SECTION *;
 /* do weighting and lagged-value selection in one step */
  %do i=0 %to %eval(%eval(&smthpnts)-1);
  /* &i used as indicator for array of “s” variables and to define the
     lags and weights for each variable */

/* split loop in two pieces to facilitate different weighting pattern on
   either “tail”.  */
%if &i <= &tails %then %do;
     s&i=lag&i(&smthvar)
   /* use increasing weights in first half of smoothing period */
      %if &ctrwgt=Y %then %do; *(&i+1) %end;
   /* “halve” weights at end points when smoothing even-period */
      %if &odd=N and &i=0 %then %do; /2 %end;
    ;
   %end;

   %if &i > &tails %then %do;
     s&i=lag&i(&smthvar)
    /* use decreasing weights in second half of smoothing period */
      %if &ctrwgt=Y %then %do; *(&smthpnts-&i) %end;
   /* “halve” weights at end points when smoothing even-period */
      %if &odd=N and &i=%eval(%eval(&smthpnts)-1) %then %do;
         /2
      %end;
    ;
   %end;

  %end;



* AVERAGING SECTION *;
/* nstart defines sufficient data for by-group processing */
 %if %length(&byvar)>0 %then %do;
  if not first.&byvar then do;
  if _n_>=nstart then do;
    newby=lag_by;     /* define value for new by-var */
 %end;

/* otherwise, period of smoothing defines sufficient data */
 %else %do;
  if _n_>=&smthpnts then do;
 %end;

    newid =lag_id;    /* define value for new id-var */

  /* smoothed value is sum of weighted lagged values
     divided by proper weight-sum */
    smooth = sum(of
     %do i= 0 %to %eval(%eval(&smthpnts)-1);
      s&i
     %end;
     )

   /* weight-sum with center-weighting is
      &sumwt as defined in _null_ data step above */
```

```
      %if &ctrwgt=Y %then %do;
       /(&sumwt
       %if &odd=N %then %do;
        - 1     /* adjust for odd-period smoothing if necessary */
       %end;
       )
      %end;

  /* weight-sum is number of points in period
      of smoothing when not center-weighting */
      %else %do; /&smthpnts %end;
     ;
    end;

/* add end for by-var loop */
%if %length(&byvar)>0 %then %do;
  end;
 %end;

/* format new id-vars */
 %if %length(&idfmt)>0 %then %do;
  format newid lag_id &idvar &idfmt;
 %end;
run;


/*** PREPARE OUTPUT ***/
/*  sort output data set and remove unused records */
proc sort data=smth;
 by
      %if %length(&byvar)>0 %then %do;
        &byvar
      %end;
    &idvar;


/* remove records based on missing id- and by- variable values */
 where
      %if %length(&byvar)>0 %then %do;
       &byvar^="" and
      %end;
        &idvar^=.;
run;


/*** OUTPUT DATA ***/
  /* merge new variable SMOOTH onto original data set */
data &indata;
 merge &indata(in=a) smth(in=b);   /* new variable is SMOOTH */
 by
    %if %length(&byvar)>0 %then %do;
     &byvar
    %end;
      &idvar;
 if a;
run;

%mend;   /* end of smoothing macro */

%smthloop;   /* call smoothing macro */
```