

A Macro for Efficient and Flexible Data Smoothing

John Cucka, Elrick & Lavidge, Westchester, IL

ABSTRACT

This paper provides the user with a simple and flexible method for adding data smoothing to their analytical tool box. In the included macro, 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. In addition, several methods of interpolation of missing smoothed data values are proposed and contrasted.

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 moving averages 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, one DATA step, requiring only a single pass at the data, contains all of the smoothing computations. 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, and some suggestions for its usage and further development.

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 length of the tails on either side of the center point of the period of smoothing. 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.

A 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, from the point itself out to the one less than the period of smoothing. In other words, if the period of smoothing is 5, then 5 variables are created, each corresponding to the LAG0-LAG4 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 variables are 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.

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 selecting an even-period smoothing, the two end-points are 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.

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

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 for SUMWGT 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.

Recall that 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, on the previous page, 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

A 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 data set from earlier examples as it would be output from the smoothing macro. Notice the addition of the variable SMOOTH.

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	.

PARAMETER SELECTION

With the availability of a new tool comes the inevitable questions about what are the best settings for the parameters. The obvious answer is: it depends on the data and the strategic goal of the analysis!

For seasonal data, if you are trying to identify a trend that is independent of the seasonality, it is best to choose a period of smoothing that matches the seasonal period. So, for weekly data, that might be 52 or 53, for monthly data that might be 12.

For non-seasonal data, consider the goal of the analysis and the amount of smoothing you would like to accomplish (relative to the noise in the data). Keep in mind that longer periods of smoothing tend to remove more noise and smooth the line more, while shorter periods of smoothing result in smoothed series that are closer to the original data series.

Selection of the use of center-weighting requires equally careful consideration. Keep in mind that center-weighting gives higher weights to the points closest to the smoothed point, while non-center-weighted smoothing gives equal weights to all points within the period of smoothing. The result is that use of center-weighting tends to maintain more of the original data series structure than non-center-weighting, but potentially also more noise.

INTERPOLATION OF MISSING VALUES

A number of methods for interpolation of the missing values at the ends of the smoothed series are provided below. It is not in the scope of this paper to provide code for any of the methods described. Hopefully, the application of these ideas will stimulate more development of the data smoothing technique.

The simplest interpolation is to use the last (or first) valid value from the series and substitute it for the nearby missing values. For example, in the case of the data in Table 3, the value of SMOOTH for observations 1 and 2 would be 6230.4. Notice that this is equivalent to doing a PROC REG using the nearest valid data point and the missing points, where the valid data point is the predicted value for the missing data points.

Staying with the regression theme, another possible solution is to use several of the last valid smoothed values and perform a regression (linear or quadratic) for the missing values. Using a 5 point linear regression for observation 2 from the data in Table 3; the value of SMOOTH would be 6083.1.

Another interpolation method is to estimate all smoothed values by simply excluding any missing data. So the estimate for the missing values would be based strictly on the available data. Conceptually, think of this as a "disappearing" weight scheme, where, at the ends of the series, the weights from one tail of the smoothing algorithm gradually disappear. In the case of the data in Table 3, the

smoothed value for observation 2 would be the average of observations 1 through 4, or 5713.8.

Another suggestion is to select a period of smoothing for the whole series, and then to fill in the missing values with smoothed values generated from successively smaller smoothing periods. So, if the period of smoothing for the whole series is 5, then fill in the two immediate missing data points with values from a smoothing period of 3 and the last values from a smoothing period of 1 (the point itself). In the case of the data in Table 3, the smoothed value for observation 2 would be the average of observations 1 through 3, or 5656.3.

Or, you may try any combination of the above! For example, compute the first few missing data points from successively smaller periods of smoothing, and then extrapolate the rest from a regression.

Interpolation Cautions

It is important to remember that none of the interpolations described above can completely adequately replace the properly computed smoothed value (where all more data is available). In many cases, these methods give increasingly more weight to fewer data points, resulting in more noise being in the smoothed series. In addition to resulting in biased smoothed value estimates, this may result in correlation with the effects you were trying to remove. Even in our simple example using the data from Table 3, each of the interpolation methods results in considerable disparate answers.

So, if you choose to apply any of these methods, be sure to check the results of the new smoothed series against the original to make sure that the interpolations are reasonable compared with the original values. (PROC GPLOT can be especially useful for this.) Furthermore, use caution if incorporating these interpolated values in another multivariate technique because the noise introduced by interpolation may be intercorrelated with other variables in the analysis.

CONCLUSION

As seen here, using simple procedures, data steps, and a macro interface, a flexible and efficient mechanism for performing data smoothing can be made readily available to data analysts everywhere. There are also many ways to interpolate the resultant missing values from data smoothing. It is up to the analyst to carefully consider when data smoothing may be useful or necessary. Selection of parameter settings, and interpolation, if any, to use in smoothing should also be closely evaluated before further utilization.

SAS, SAS/GRAPH, 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.

Author Information:

John Cucka
Manager, Marketing Sciences
Elrick & Lavidge
3 Westbrook Corporate Center, Suite 600
Westchester, IL 60154
tel: (708) 449-4317
fax: (708) 449-4498
email: chicago_el@pop.mindspring.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 */
    sumwt=(tails+1)**2; /* sum of the weights in
                        center-weighting */

    /* write out all values to macro variables */
    call symput ('tails',left(trim(tails)));
    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 only necessary vars */
    (keep=newid smooth
    %if %length(&byvar)>0 %then %do;
    newby
    %end;

    /* rename by-vars for merging later */
    rename=(newid=&idvar
    %if %length(&byvar)>0 %then %do;
    newby=&byvar
    %end;
    ));

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

set smthdat; /* set in original data */

%if %length(&byvar)>0 %then %do;
    by &byvar;

    /* LAG_BY is defined as the value for the by-group at the
       center of the smoothing period */
    lag_by =lag&tails(&byvar);

    /* NSTART is an indicator of the first observation in each
       by-group with sufficient data for smoothing */
    if first.&byvar then do;
        nstart=_n_ + &smthpnts - 1;
    end;
    retain nstart;

%end;

/* LAG_ID is defined as the value for the id-variable at the
   center of the smoothing period */
lag_id =lag&tails(&idvar);
```

***** WEIGHTING SECTION *****

/ selects lagged-values and applies weights in one step */*

/ &l will be used as the indicator for the array of "s"
variables and also to define the lags and weights for
each variable */*

%do i=0 %to %eval(%eval(&smthpnts)-1);

/ split %DO loop into two pieces to facilitate different
weighting patterns on either "tail". */*

/ first, the ascending portion of the loop. */
%if &i <= &tails %then %do;
s&i=lag&i(&smthvar)*

/ use increasing weights in first half
of the smoothing period */
%if &ctwgt=Y %then %do; *(&i+1) %end;*

/ when using an even-period of smoothing, "halve"
the weights at the end points */
%if &odd=N and &i=0 %then %do;
/2
%end;*

%end; / end of ascending portion */*

/ next, the descending portion of the loop. */
%if &i > &tails %then %do;
s&i=lag&i(&smthvar)*

/ use decreasing weights in second half
of the smoothing period */
%if &ctwgt=Y %then %do; *(&smthpnts-&i) %end;*

/ when using an even-period of smoothing, "halve"
the weights at the end points */
%if &odd=N and &i=%eval(%eval(&smthpnts)-1)
%then %do;
/2
%end;*

%end; / end of descending portion */*

%end; / end of %DO over &l loop */*

***** AVERAGING SECTION *****

/ define value for new by-var based on sufficient data */*

/ with by-groups, sufficient data is defined by NSTART */
%if %length(&byvar)>0 %then %do;
if not first.&byvar then do;
if _n_>=nstart then do;
newby=lag_by;
%end;*

/ without by-groups, the period of smoothing defines
sufficient data */*

*%else %do;
if _n_>=&smthpnts then do;
%end;*

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

/ the smoothed value is the sum of the weighted lagged
values divided by proper weight-sum */*

/ first, compute the numerator */
smooth = sum(of
%do i= 0 %to %eval(%eval(&smthpnts)-1);
s&i
%end;*

) / (

/ next, compute the denominator */
/* when center-weighting, the proper weight-sum is as
defined in the _null_ data step at the top of the
program */
%if &ctwgt=Y %then %do;
&sumwt
%end;*

/ otherwise, the proper weight-sum is simply the number
of points in the smoothing period */
%else %do;
&smthpnts
%end;*

/ in either case, the weight-sum must be adjusted for odd-
periods smoothing (if necessary) */
%if &odd=N %then %do;
- 1
%end;*

*);
end;*

/ when by-groups are used, add an extra END */
%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);  
  by  
    %if %length(&byvar)>0 %then %do;  
      &byvar  
    %end;  
    &idvar;  
  if a;  
run;
```

%mend; */* end of smoothing macro */*

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