PharmaSUG 2018 - Paper QT-05

SAS® Macros of Performing Look-Ahead and Look-Back Reads

Yanhong Liu, Cincinnati Children’s Hospital Medical Center, Cincinnati, OH

ABSTRACT

When working with the time series data, it is generally required use of previous observation(s) (Look-Back) or next observation(s) (Look-Ahead) for certain variable, in order to do some comparison or calculation. The SAS® Lag function can be used to get the Look-Back reads, and we can use the Double SET statement to perform the Look-Ahead reads. If you have SAS/ETS® installed, you can use PROC EXPAND as an alternative. In this paper, a few of macros are introduced to obtain one or more Look-back or Look-ahead observations with by-groups by implementing the above techniques. Additionally, this paper briefly compares the cost of the DATA STEP Macros to the PROC EXPAND Macros.

INTRODUCTION

There is often interest in creating variables based on how data for a given time period compares to the periods before and after when working with data across consistent units of time (years, quarters, months, days). If you have time series or longitudinal data, you wish to look across units of time within a single subject. The SAS® LAG function can be used in the data step to perform computations across observations. It will return the values of a variable from previous observations. This is known as a look-back read. Even though there is no corresponding function to look ahead the next or subsequent observations, we can use the technique called Double SET statement to perform the Look-Ahead reads. When using those techniques to process across observations with data in groups, especially, if want to read more than one back or ahead values, we must carefully program how to detect and handle the group boundaries in order to avoid producing unexpected results. In this paper, a few of macros are introduced to retrieve one or more Look-back or Look-ahead observations with or without by-groups by using the Lag function or Double SET statement. However, PROC EXPAND provides an easy-to-use alternative to the data step if you have SAS/ETS® installed. This paper also includes the macros by using the PROC EXPAND to generate the LAG and LEAD reads, but the result of the comparison for the cost of DATA STEP to PROC EXPAND may surprise you.

MACRO PARAMETERS

These macros all have 4 parameters.

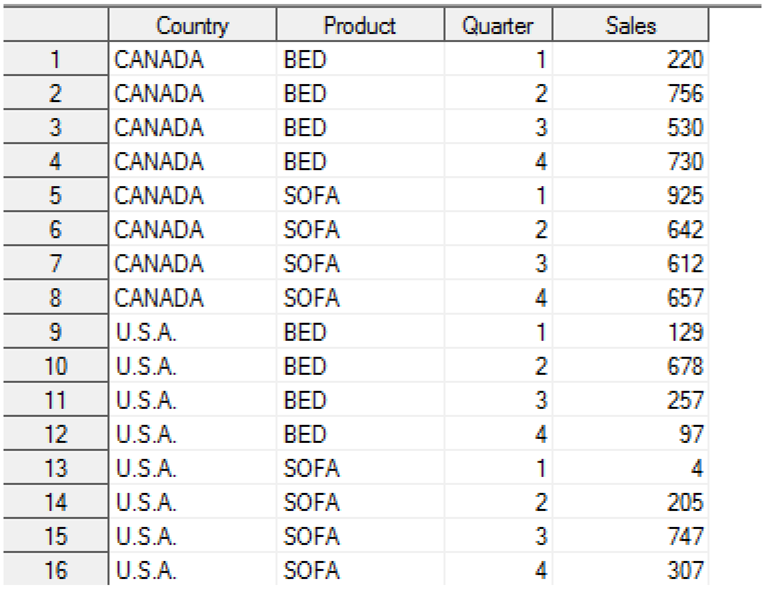
DATASET: Name of the input data set.

GROUPBY: List of BY variables. The input data set is sorted by these variables. If there are no GROUPBY variables specified then input data set is in its given observation ordering. (At the beginning of each macro, a new data set will be created by adding 2 temporary variables _temp1 and _temp2 to the input data set, _temp1 is a dummy key, and _temp2 is a key of the original observation sequencing.)

VAR: Name of the variable to be read ahead or read back

LOOKN: The number of look- ahead or look-back reads

A SAMPLE DATA SET (SAMPLE. SAS7BDAT)



	Country	Product	Quarter	Sales
1	CANADA	BED	1	220
2	CANADA	BED	2	756
3	CANADA	BED	3	530
4	CANADA	BED	4	730
5	CANADA	SOFA	1	925
6	CANADA	SOFA	2	642
7	CANADA	SOFA	3	612
8	CANADA	SOFA	4	657
9	U.S.A.	BED	1	129
10	U.S.A.	BED	2	678
11	U.S.A.	BED	3	257
12	U.S.A.	BED	4	97
13	U.S.A.	SOFA	1	4
14	U.S.A.	SOFA	2	205
15	U.S.A.	SOFA	3	747
16	U.S.A.	SOFA	4	307

Display 1. A Sample data set to be used for testing the Macros

DATA STEP MACROS

THE LOOKBACK MACRO

SAS® code listing:

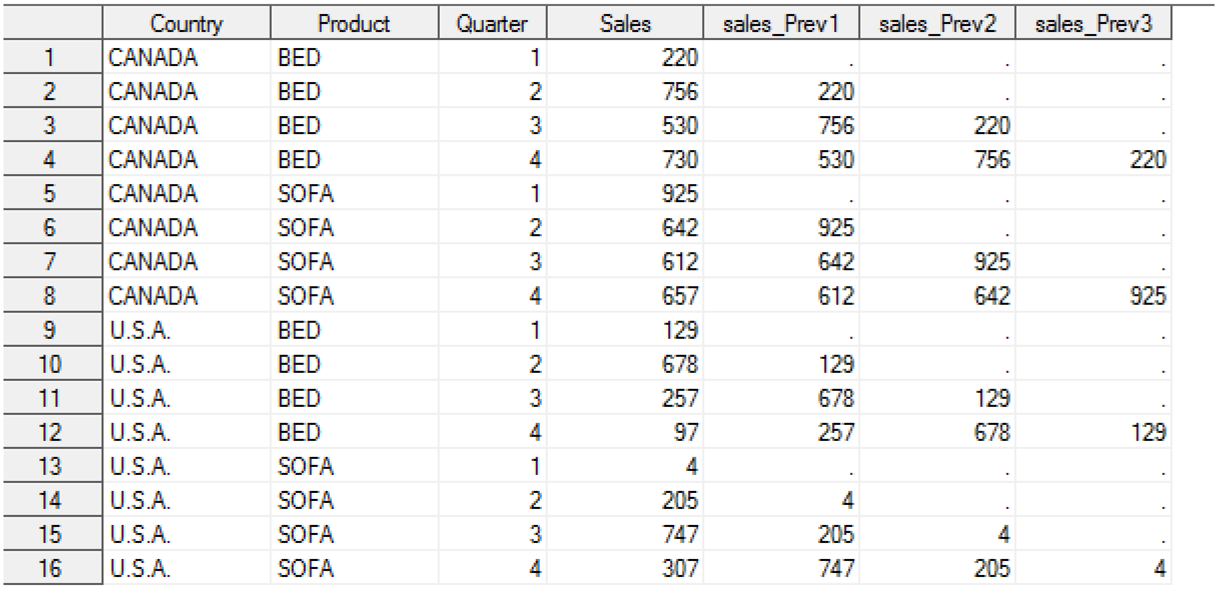
```
%macro lookback(dataset=, groupby=, LookN=, var= );

data lookback;
set &dataset;
 _temp1=1;_temp2=_n_;
run;

proc sort data=lookback; by &groupby _temp1 _temp2; run;

/*** Set up BYVAR: The by variable to determine the group boundary ***/
%if &groupby ne %then %do;
 %let byvar=%scan(&groupby,-1);%end;
%else %if  &groupby= %then %do;
 %let byvar=_temp1;%end;

data &dataset._lookback(drop=i count _temp1 _temp2);

      do count = 1 by 1 until (last.&byvar);
         set lookback(keep=&groupby _temp1 _temp2 );
         by &groupby _temp1 _temp2 ;
      end;
/**Count the observations in the group specified by GROUPBY**/

      do i = 1 to count;
            set lookback;
                  %do j=1 %to &lookn;
                        &var._Prev&j = lag&j( &var );
/**Function LAG is used for the look-back read**/
```

```
                        if i-&j<=0 then &var._Prev&j=.;
/**Reset the value to missing when necessary**/
                  %end;
            output;
      end;
run;

proc datasets nolist; /* Remove the temporary data set */
 delete lookback;
 run;
quit;
%mend;
```

Display 2. The Macro LookBack SAS® Code listing

Example of calling the Macro %lookback:

```
%lookback(dataset=sample, groupby=country product, lookN=3, var=sales);
```



	Country	Product	Quarter	Sales	sales_Prev1	sales_Prev2	sales_Prev3
1	CANADA	BED	1	220	.	.	.
2	CANADA	BED	2	756	220	.	.
3	CANADA	BED	3	530	756	220	.
4	CANADA	BED	4	730	530	756	220
5	CANADA	SOFA	1	925	.	.	.
6	CANADA	SOFA	2	642	925	.	.
7	CANADA	SOFA	3	612	642	925	.
8	CANADA	SOFA	4	657	612	642	925
9	U.S.A.	BED	1	129	.	.	.
10	U.S.A.	BED	2	678	129	.	.
11	U.S.A.	BED	3	257	678	129	.
12	U.S.A.	BED	4	97	257	678	129
13	U.S.A.	SOFA	1	4	.	.	.
14	U.S.A.	SOFA	2	205	4	.	.
15	U.S.A.	SOFA	3	747	205	4	.
16	U.S.A.	SOFA	4	307	747	205	4

Display 3. The results of calling %lookback

THE LOOKAHEAD MACRO

SAS® code listing:

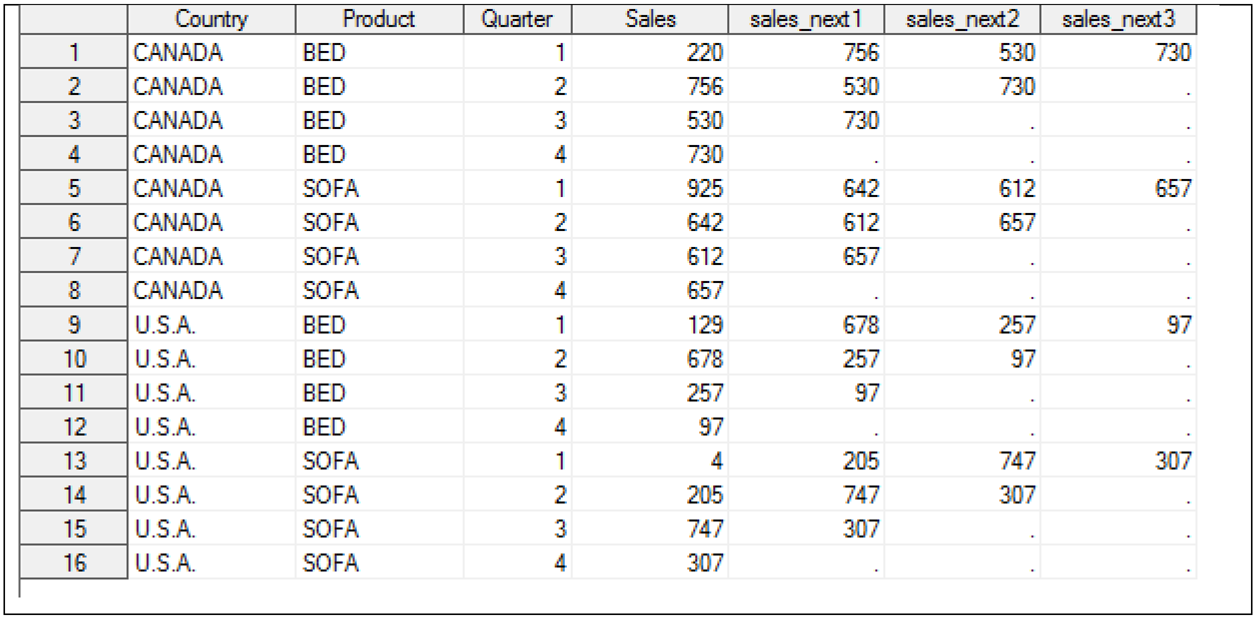
```
%macro lookahead(dataset=, groupby=, lookN=, var= );

data lookahead;
set &dataset;
_temp1=1;_temp2=_n_;
run;

proc sort data=lookahead; by &groupby _temp1 _temp2; run;
```

```
%if &groupby ne %then %do;
 %let byvar=%scan(&groupby,-1);
%end;
%else %if  &groupby= %then %do;
 %let byvar=_temp1;
%end;

data &dataset._lookahead(drop=i count _temp1 _temp2);
      do count = 1 by 1 until (last.&byvar);
      set lookahead(keep=&groupby _temp1 _temp2);
      by &groupby _temp1 _temp2;
      end;
/**Count the observations in the group specified by GROUPBY**/

      do i = 1 to count;
         set lookahead;
            %do j=1 %to &lookn;
              set lookahead ( firstobs = %eval(%eval(&j)+1) keep = &var
rename = (&var = &var._next&j )
                  lookahead (      obs = %eval(%eval(&j)) drop = _all_);
/**Double SET statement is used for the look-ahead read**/
             &var._next&j = ifn((count - i)<&j, (.), &var._next&j);
/**Reset the value to missing when necessary**/
             %end;
      output;
      end;
run;

proc datasets nolist; /* Remove the temporary data set */
 delete lookahead;
run;
quit;
%mend;
```

Display 4. The Macro LookAhead SAS® Code listing

Example of calling the Macro %lookAhead:

```
%lookahead(dataset=sample, groupby=country product, lookn=3, var=sales);
```



	Country	Product	Quarter	Sales	sales_next1	sales_next2	sales_next3
1	CANADA	BED	1	220	756	530	730
2	CANADA	BED	2	756	530	730	.
3	CANADA	BED	3	530	730	.	.
4	CANADA	BED	4	730	.	.	.
5	CANADA	SOFA	1	925	642	612	657
6	CANADA	SOFA	2	642	612	657	.
7	CANADA	SOFA	3	612	657	.	.
8	CANADA	SOFA	4	657	.	.	.
9	U.S.A.	BED	1	129	678	257	97
10	U.S.A.	BED	2	678	257	97	.
11	U.S.A.	BED	3	257	97	.	.
12	U.S.A.	BED	4	97	.	.	.
13	U.S.A.	SOFA	1	4	205	747	307
14	U.S.A.	SOFA	2	205	747	307	.
15	U.S.A.	SOFA	3	747	307	.	.
16	U.S.A.	SOFA	4	307	.	.	.

Display 5. The results of calling %lookAhead

THE LOOKBOTH MACRO

SAS® code listing (combine LookBack and LookAhead):

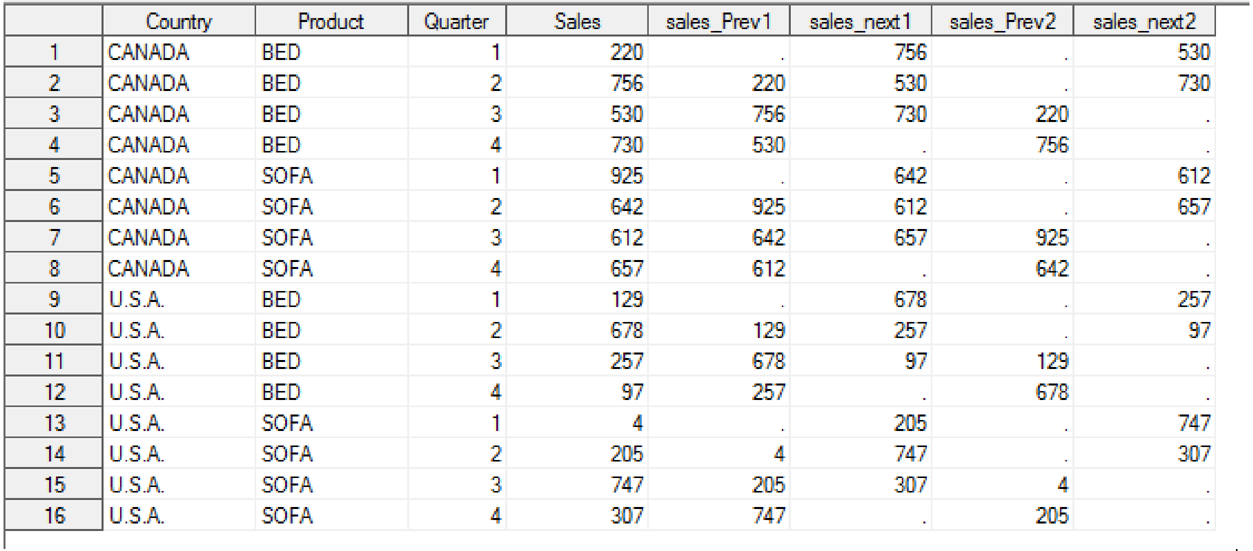
```
%macro lookboth(dataset=, groupby=, lookN=, var= );
data lookboth;
set &dataset;
_temp1=1;_temp2=_n_;
run;

proc sort data=lookboth; by &groupby _temp1 _temp2; run;

%if &groupby ne %then %do;
 %let byvar=%scan(&groupby,-1);
%end;
%else %if  &groupby= %then %do;
 %let byvar=_temp1;
%end;

data &dataset._lookboth(drop=i count _temp1 _temp2);
      do count = 1 by 1 until (last.&byvar);
          set lookboth(keep=&groupby _temp1 _temp2);
          by &groupby _temp1 _temp2;
      end;

      do i = 1 to count;
          set lookboth;

            %do j=1 %to &lookn;
                     /**Look Back**/
                  &var._Prev&j = lag&j( &var );
                  if i-&j<=0 then &var._Prev&j=.;


                  /**Look Ahead**/
```

```
               set lookboth ( firstobs = %eval(%eval(&j)+1) keep = &var
rename = (&var = &var._next&j) )
                      lookboth (      obs = %eval(%eval(&j)) drop = _all_);
                &var._next&j = ifn(  (count - i) < &j, (.), &var._next&j );
            %end;

output;
end;
run;

proc datasets nolist; /* Remove the temporary data set */
 delete lookboth;
 run;
quit;
%mend;
```

Display 6. The Macro LookBoth SAS® Code listing

Example of calling the Macro %lookBoth:

```
%lookboth(dataset=sample, groupby=country product, lookn=2, var=sales);
```



	Country	Product	Quarter	Sales	sales_Prev1	sales_next1	sales_Prev2	sales_next2
1	CANADA	BED	1	220	.	756	.	530
2	CANADA	BED	2	756	220	530	.	730
3	CANADA	BED	3	530	756	730	220	.
4	CANADA	BED	4	730	530	.	756	.
5	CANADA	SOFA	1	925	.	642	.	612
6	CANADA	SOFA	2	642	925	612	.	657
7	CANADA	SOFA	3	612	642	657	925	.
8	CANADA	SOFA	4	657	612	.	642	.
9	U.S.A.	BED	1	129	.	678	.	257
10	U.S.A.	BED	2	678	129	257	.	97
11	U.S.A.	BED	3	257	678	97	129	.
12	U.S.A.	BED	4	97	257	.	678	.
13	U.S.A.	SOFA	1	4	.	205	.	747
14	U.S.A.	SOFA	2	205	4	747	.	307
15	U.S.A.	SOFA	3	747	205	307	4	.
16	U.S.A.	SOFA	4	307	747	.	205	.

Display 7. The results of calling %lookBoth

THE PROC EXPAND

PROC EXPAND is one of the most useful procedures of SAS/ETS® (Econometric Time Series). It has multiple functions includes creating lag, lead. PROC EXPAND is designed to obtain LAG and LEAD values by using the CONVERT statement. The METHOD=NONE option is used to suppress interpolation (SAS® default to interpolating missing values using a cubic spline function).

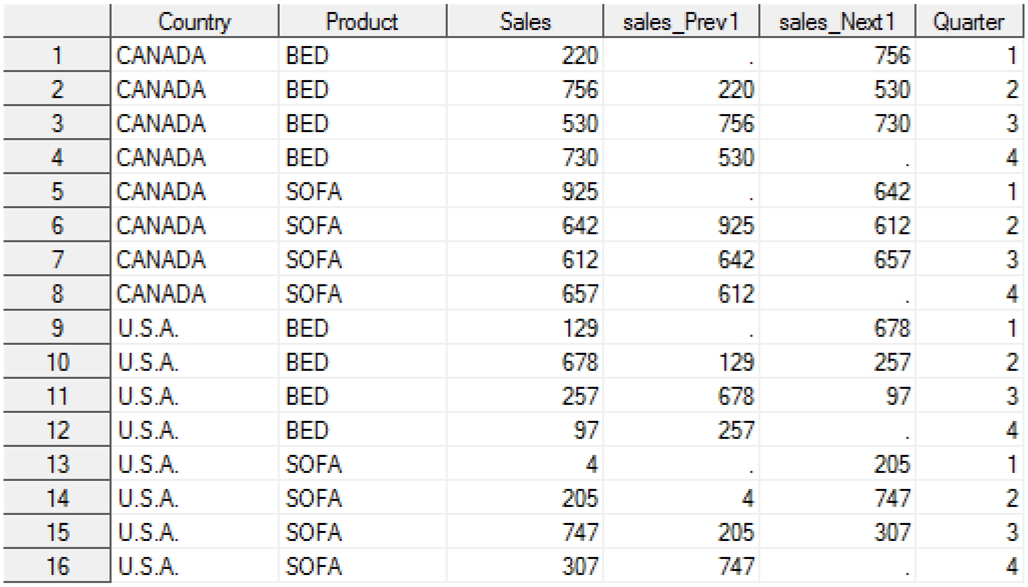
The general syntax for PROC EXPAND uses the following format:

```
PROC EXPAND DATA=dataset OUT=out_dataset METHOD=NONE;

            CONVERT existing_variable = derived_variable /
            TRANSFORMOUT=(transformation);
```

```
      BY Panel_id_variable;
      RUN;
```

A simple example of using LAG or LEAD transformation:

```
proc expand data=sample out = Sample_out(drop=time) method=none;
by country product;
convert sales;
convert sales = sales_Prev1 / transformout= (lag 1);
convert sales =sales_Next1 / transformout= (lead 1);
run;
```



	Country	Product	Sales	sales_Prev1	sales_Next1	Quarter
1	CANADA	BED	220	.	756	1
2	CANADA	BED	756	220	530	2
3	CANADA	BED	530	756	730	3
4	CANADA	BED	730	530	.	4
5	CANADA	SOFA	925	.	642	1
6	CANADA	SOFA	642	925	612	2
7	CANADA	SOFA	612	642	657	3
8	CANADA	SOFA	657	612	.	4
9	U.S.A.	BED	129	.	678	1
10	U.S.A.	BED	678	129	257	2
11	U.S.A.	BED	257	678	97	3
12	U.S.A.	BED	97	257	.	4
13	U.S.A.	SOFA	4	.	205	1
14	U.S.A.	SOFA	205	4	747	2
15	U.S.A.	SOFA	747	205	307	3
16	U.S.A.	SOFA	307	747	.	4

Display 8. An example by using Proc Expand to get the LAG and LEAD reads

THE LOOKBACK AND LOOKAHEAD MACROS BY USING PROC EXPAND

%LOOKBACK_EXPAND

```
%macro lookback_expand(dataset=, groupby=, lookn=, var=);
data lookback;
set &dataset;
_temp1=1;_temp2=_n_;
run;

proc sort data=lookback; by &groupby _temp1 _temp2; run;

proc expand data = lookback out = &dataset._lookback_expand(drop=time _temp1
_temp2) method=none;
by &groupby _temp1;
            convert &var ;
      %do j=1 %to &lookn;
            convert &var =&var._prev&j / transformout= (lag &j);
      %end;
run;
```

```
proc datasets nolist; /* Remove the temporary data set */
 delete lookback;
 run;
quit;
%mend;
```

%LOOKAHEAD_EXPAND

```
%macro lookahead_expand(dataset=, groupby=, lookn=, var=);
data lookahead;
set &dataset;
_temp1=1;_temp2=_n_;
run;

proc sort data=lookahead; by &groupby _temp1 _temp2; run;

proc expand data = lookahead out = &dataset._lookahead_expand(drop=time
_temp1 _temp2) method=none;
by &groupby _temp1;
            convert &var ;
      %do j=1 %to &lookn;
            convert &var =&var._next&j / transformout= (lead &j);
      %end;
run;

proc datasets nolist; /* Remove the temporary data set */
 delete lookahead;
 run;
quit;
%mend;
```

%LOOKBOTH_EXPAND

```
%macro lookboth_expand(dataset=, groupby=, lookn=, var=);
data lookboth;
set &dataset;
_temp1=1;_temp2=_n_;
run;

proc sort data=lookboth; by &groupby _temp1 _temp2; run;

proc expand data = lookboth out = &dataset._lookboth_expand(drop=time _temp1
_temp2) method=none;
by &groupby _temp1;
                  convert &var ;
            %do j=1 %to &lookn;
                  convert &var =&var._prev&j / transformout= (lag &j);
                  convert &var =&var._next&j / transformout= (lead &j);
            %end;
    run;
proc datasets nolist; /* Remove the temporary data set */
 delete lookboth;
 run;
quit;
```

```
%mend;
```

Display 9. SAS® Code listings of Look-Back or Look_Ahead Macros by using PROC EXPAND

DATA STEP MACRO OR PROC EXPAND MACRO?

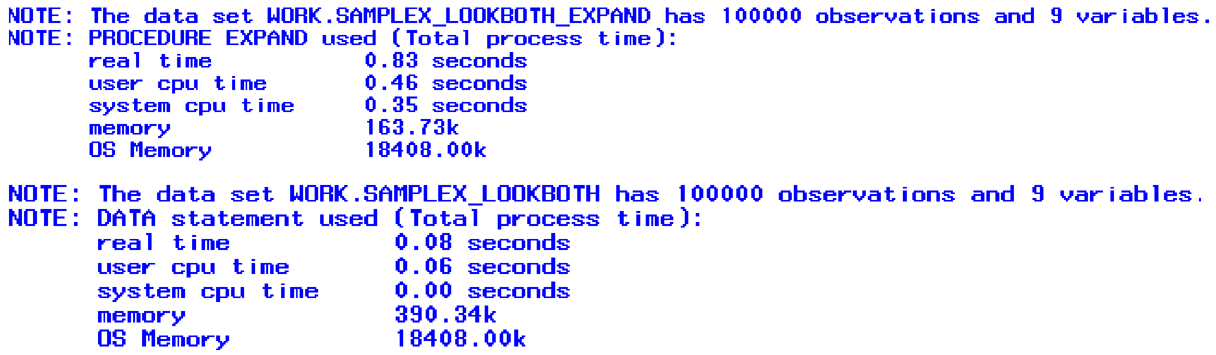
---The PROC EXPAND Macro Works only if you have a license of SAS® ETS package

---The DATA STEP Macros have more codes to type, but process speed are much faster than the PROC EXPAND.

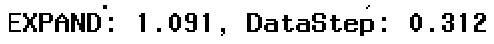
SAS Code to compare the cost of both macros:

```
options fullstimer;
/**Samplex.sas7bdat has 100,000 observations***/
%let t1 = %sysfunc(time());
%LOOKBOTH_EXPAND(dataset=samplex, groupby=id drugname, lookn=3,
var=DayofService);
%let t2 = %sysfunc(time());
%lookboth(dataset=samplex, groupby=id drugname, lookn=3, var=DayofService);
%let t3 = %sysfunc(time());

%let d2 = %sysfunc(round( %sysevalf(&t2 - &t1), 0.001));
%let d3 = %sysfunc(round( %sysevalf(&t3 - &t2), 0.001));

%put  EXPAND: &d2, DataStep: &d3;
```

Results:



```
NOTE: The data set WORK.SAMPLEX_LOOKBOTH_EXPAND has 100000 observations and 9 variables.
NOTE: PROCEDURE EXPAND used (Total process time):
      real time           0.83 seconds
      user cpu time       0.46 seconds
      system cpu time     0.35 seconds
      memory              163.73k
      OS Memory           18408.00k

NOTE: The data set WORK.SAMPLEX_LOOKBOTH has 100000 observations and 9 variables.
NOTE: DATA statement used (Total process time):
      real time           0.08 seconds
      user cpu time       0.06 seconds
      system cpu time     0.00 seconds
      memory              390.34k
      OS Memory           18408.00k
```



```
EXPAND: 1.091, DataStep: 0.312
```

Display 10. Comparing the cost of Data Step to Proc Expand

CONCLUSION

The Macros introduced in this paper are very useful for the Look-Back or Look-Ahead reads, which can reduce the programming effort to avoid the unexpected results when performing across observations with data in groups. Especially, if want to retrieve more than one LAGs or LEADs.

REFERENCES

Matlapudi, Anjan, and J. Daniel Knapp (2010) “Please Don’t Lag Behind LAG”. In the Proceedings of the

North East SAS® Users Group.SAS®

sasCommunity.org “Look-Ahead and Look-Back” http://www.sascommunity.org/wiki/Look-Ahead_and_Look-Back

Leonard Landry (2010) “Look Before You Leap – A Technique to Simulate a Look-Ahead Function”. In the proceedings of the SAS® Global Forum.

ACKNOWLEDGMENTS

The author wishes to thank the Division of Biostatistics and Epidemiology at Cincinnati Children's Hospital Medical Center for its support.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Yanhong Liu
Cincinnati Children’s Hospital Medical Center
3333 Burnet Ave.
Cincinnati, OH 45255
(513) 803-2614
E-mail: Yanhong.Liu@cchmc.org