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)

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

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 );
%end;

/***Function LAG is used for the look-back read**/
```
Display 2. The Macro LookBack SAS® Code listing

Example of calling the Macro %lookback:

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

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_);
    %end;
/**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);
SAS® Macros of Performing Look-Ahead and Look-Back Reads, continued

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

Display 5. The results of calling %lookAhead

THE LOOKBOTH MACRO

SAS® code listing (combine LookBack and LookAhead):

```sas
%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=.
  %end;

  /**Look Ahead**/
```

5
SAS® Macros of Performing Look-Ahead and Look-Back Reads, continued

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);

<table>
<thead>
<tr>
<th>Country</th>
<th>Product</th>
<th>Quarter</th>
<th>Sales</th>
<th>sales_Prev1</th>
<th>sales_next1</th>
<th>sales_Prev2</th>
<th>sales_next2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CANADA</td>
<td>BED</td>
<td>1</td>
<td>220</td>
<td>.</td>
<td>756</td>
<td>530</td>
</tr>
<tr>
<td>2</td>
<td>CANADA</td>
<td>BED</td>
<td>2</td>
<td>756</td>
<td>220</td>
<td>530</td>
<td>730</td>
</tr>
<tr>
<td>3</td>
<td>CANADA</td>
<td>BED</td>
<td>3</td>
<td>530</td>
<td>756</td>
<td>730</td>
<td>220</td>
</tr>
<tr>
<td>4</td>
<td>CANADA</td>
<td>BED</td>
<td>4</td>
<td>730</td>
<td>530</td>
<td>.</td>
<td>756</td>
</tr>
<tr>
<td>5</td>
<td>CANADA</td>
<td>SOFA</td>
<td>1</td>
<td>525</td>
<td>.</td>
<td>542</td>
<td>612</td>
</tr>
<tr>
<td>6</td>
<td>CANADA</td>
<td>SOFA</td>
<td>2</td>
<td>642</td>
<td>925</td>
<td>612</td>
<td>657</td>
</tr>
<tr>
<td>7</td>
<td>CANADA</td>
<td>SOFA</td>
<td>3</td>
<td>612</td>
<td>642</td>
<td>657</td>
<td>925</td>
</tr>
<tr>
<td>8</td>
<td>CANADA</td>
<td>SOFA</td>
<td>4</td>
<td>657</td>
<td>612</td>
<td>.</td>
<td>642</td>
</tr>
<tr>
<td>9</td>
<td>U.S.A</td>
<td>BED</td>
<td>1</td>
<td>129</td>
<td>.</td>
<td>678</td>
<td>257</td>
</tr>
<tr>
<td>10</td>
<td>U.S.A</td>
<td>BED</td>
<td>2</td>
<td>678</td>
<td>129</td>
<td>257</td>
<td>.</td>
</tr>
<tr>
<td>11</td>
<td>U.S.A</td>
<td>BED</td>
<td>3</td>
<td>257</td>
<td>678</td>
<td>57</td>
<td>129</td>
</tr>
<tr>
<td>12</td>
<td>U.S.A</td>
<td>BED</td>
<td>4</td>
<td>97</td>
<td>257</td>
<td>.</td>
<td>678</td>
</tr>
<tr>
<td>13</td>
<td>U.S.A</td>
<td>SOFA</td>
<td>1</td>
<td>4</td>
<td>.</td>
<td>205</td>
<td>747</td>
</tr>
<tr>
<td>14</td>
<td>U.S.A</td>
<td>SOFA</td>
<td>2</td>
<td>205</td>
<td>4</td>
<td>747</td>
<td>.</td>
</tr>
<tr>
<td>15</td>
<td>U.S.A</td>
<td>SOFA</td>
<td>3</td>
<td>747</td>
<td>205</td>
<td>307</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>U.S.A</td>
<td>SOFA</td>
<td>4</td>
<td>307</td>
<td>747</td>
<td>.</td>
<td>205</td>
</tr>
</tbody>
</table>

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 including 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);
```

6
BY Panel_id_variable;
RUN;

A simple example of using LAG or LEAD transformation:

```sas
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;
```

<table>
<thead>
<tr>
<th>Country</th>
<th>Product</th>
<th>Sales</th>
<th>sales_Prev1</th>
<th>sales_Next1</th>
<th>Quarter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANADA</td>
<td>BED</td>
<td>220</td>
<td>756</td>
<td>530</td>
<td>1</td>
</tr>
<tr>
<td>CANADA</td>
<td>BED</td>
<td>756</td>
<td>220</td>
<td>530</td>
<td>2</td>
</tr>
<tr>
<td>CANADA</td>
<td>BED</td>
<td>530</td>
<td>756</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CANADA</td>
<td>BED</td>
<td>730</td>
<td>530</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANADA</td>
<td>SOFA</td>
<td>925</td>
<td></td>
<td>642</td>
<td>1</td>
</tr>
<tr>
<td>CANADA</td>
<td>SOFA</td>
<td>642</td>
<td>925</td>
<td>612</td>
<td>2</td>
</tr>
<tr>
<td>CANADA</td>
<td>SOFA</td>
<td>612</td>
<td>642</td>
<td>657</td>
<td>3</td>
</tr>
<tr>
<td>CANADA</td>
<td>SOFA</td>
<td>657</td>
<td>612</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U.S.A.</td>
<td>BED</td>
<td>129</td>
<td></td>
<td>678</td>
<td>1</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>BED</td>
<td>678</td>
<td>129</td>
<td>257</td>
<td>2</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>BED</td>
<td>257</td>
<td>678</td>
<td>97</td>
<td>3</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>BED</td>
<td>97</td>
<td>257</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U.S.A.</td>
<td>SOFA</td>
<td>4</td>
<td></td>
<td>205</td>
<td>1</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>SOFA</td>
<td>205</td>
<td>4</td>
<td>747</td>
<td>2</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>SOFA</td>
<td>747</td>
<td>205</td>
<td>307</td>
<td>3</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>SOFA</td>
<td>307</td>
<td>747</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

THE LOOKBACK AND LOOKAHEAD MACROS BY USING PROC EXPAND

```sas
%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;
```

7
SAS Macros of Performing Look-Ahead and Look-Back Reads, continued

```
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;
```
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:

```sas
options fulltimer;
/**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      162.73k
   DS Memory    10408.00k

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      330.34k
   DS 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®

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

SAS® and all other SAS® Institute Inc. product or service names are registered trademarks or trademarks of SAS® Institute Inc. in the USA and other countries. ® indicates USA registration.
Other brand and product names are trademarks of their respective companies.