A Macro to carry values through observations forwards or backwards over missing or 'null' values within a BY group or a SAS® data set.

Timothy J. Harrington, Dataceutics, inc;

ABSTRACT

This paper explains the structure of a macro and its usefulness for imputing missing or defined 'null' values with a prior or following quantity of a numeric or character column in a SAS® data set or BY group. The macro uses DICTIONARY.COLUMNS to determine the data type, carries values over multiple successive observations with missing or 'null' amounts. When the specified data set column is numeric, parameters can be passed to return the differences between each quantity and the first or last values of the by group or the successive differences between non-missing values. The 'null' value defaults to missing but can be user specified. Included in this paper are examples of practical applications, such as imputing missing dates and times of events since the first or most recent dose in PK data.

INTRODUCTION

A common situation when handling SAS data sets is 'gaps' in the data, that is missing or incomplete values in a given column. A solution is to impute the missing or incomplete values as the most recent prior value carried forward. An example of this is in SAS PK data when the dates and times of events such as a patient receiving a dose, or a blood draw taking place. When such PK data is consolidated for analysis there arises a need to express the time of an event with respect to a specified earlier event such as the date and time of the first dose or the time since the most recent dose. If the observations from a dosing data set are combined with observations from a sampling data set, such as blood draws, using a SET statement and are then sorted by visit sequence, the dose times will have missing values for the sampling observations. In order to calculate a sample time relative to a dose time the relevant dose time must be carried forward (or backward) into the sample observation. Performing a carry forwards or carry backwards is relatively complex: the data must be appropriately sorted, a column created to hold the carried forward value, that column must be retained, the value read from each non-missing observation, and be copied into the following non-missing observations. SAS LEAD and LAG functions are helpful, but may not available in earlier versions of SAS, and oftentimes there are multiple successive missing values to be copied to. The LOCFRB macro performs all of these functions and more with one macro call.

MACRO PARAMETERS

INDS: Name of the input data set.

OUTDS: Name of an output data set created by this macro containing the resulting variables. Other variables, and the data set attributes are unchanged. If OUTDS is specified the Original data set (&INDS) is unchanged. The default is no output data set is created.

BYVARS: List of BY variables. The input data set &INDS is sorted by these variables. The default is no BY variables, when the entire data set is taken unsorted as is. If BYVARS are specified the output Data set is returned sorted by the BYVARS.

GROUPVAR: BY variable whose range is to be covered. Must be one of the &BYVARS variables. If GROUPVAR is left blank the default, the last name in &BYVARS, is used. If there are no BYVARS variables specified the whole data set is taken in its given observation ordering.

VAR: Name of the variable to be carried forwards (or backwards). This is required and may be numeric or character.

NEWVAR: Name of the variable the resulting carried values are copied to. This is created by this macro with the same data type and length as &VAR. If NEWVAR is specified the original &VAR is unchanged .The default is no NEWVAR and &VAR is overwritten. If NEWVAR, or any of the variable name parameters below, are specified a variable of that name must not already exist on the data set and must not be T_VAR01 through T_VAR09.

DIFFIRST: Name of the variable containing the arithmetic differences of &VAR between the initial value at FIRST.&GROUPVAR (or _n_=1 if &BYVARS is blank)) and each carried value of &VAR. If the initial value of &VAR is missing the first non-missing value is used. This variable is not created if DIFFIRST is not specified or &VAR is of character type.

DIFFFWD: Name of the variable containing the non-missing arithmetic difference of &VAR between the current and prior non-missing observations. This variable is not created if DIFFFWD is blank or &VAR is of character type.

DIFFLAST: Name of the variable containing the arithmetic differences of &VAR between the final value at LAST.&GROUPVAR (or the last observation in the data set if &BYVARS is blank) and each carried value of &VAR. If the final value of &VAR is missing the last non-missing value is used. This variable is not created if DIFFLAST is blank or &VAR is of character type.

DIFFREV: Name of the variable containing the non-missing arithmetic differences of &VAR between the current and following non-missing observations. This variable is not created if DIFFREV is blank or &VAR is of character type.

DIR: Carrying direction. 'F'=forwards, 'R'=reverse, 'B'=forwards and then in reverse. Default is forwards only, 'F'.

NULLVAL: Value of &VAR considered as missing, default is SAS missing value (numeric . or character ' ').

SAS CODE LISTING

```
   %macro locfrb(inds=,            /* Input data set (required) */
                 outds=&inds,      /* Output data set, defaults to input */
                 byvars=,          /* Ordering columns */
                 groupvar=t_var02, /* By group to carry forward within */
                 var=,             /* Column to be processed (required) */
                 newvar=,          /* Column for created values (optional) */
                 diffirst=t_var06, /* Diff. from 1st &GROUPVAR (optional) */
                 difflast=t_var07, /* Diff. from last &GROUPVAR (optional) */
                 difffwd=t_var08,  /* Diff. from prior value (optional) */
                 diffrev=t_var09,  /* Diff. from following value (optional) */
                 dir=F,            /* Carrying direction (def. forwards) */
                 nullval= );       /* User defined null value (optional) */


%local sortkey dds dtype dlength;


/*** Determine the data type and length of VAR */

   proc sql noprint nofeedback;
     select upcase(type), length into :dtype, :dlength from
       dictionary.columns
         where upcase(libname)='WORK' and upcase(memname)="%upcase(&inds)"
           and upcase(name)="%upcase(&var)";
   quit;


/*** Set up the default GROUPVAR if none specified */

  %if &groupvar= and &byvars ne  %then %do;
    %let groupvar=%scan(&bayvars,-1);
  %end;
  %else %if &groupvar= and &byvars= %then %do;
     %let groupvar=t_var02;
  %end;


/*** Set the default null value, missing or blank. Allocate the sort key */

  %if &nullval=  %then %do;
    %if &dtype=NUM %then %do;
      %let nullval=.;
    %end;
    %else %do;
      %let nullval=%str(' ');
    %end;
  %end;


  %let sortkey=&byvars t_var02 t_var03;

/*** Keep the original values of &VAR. t_var02 is a dummy key, t_var03 is */
/*** a key of the original observation sequencing. This is used as the    */
/*** sortkey if no BYVARS are specified. Sort by the BYVAR variables      */
```

```
  data t01;
     set &inds;
%if &dtype=CHAR %then %do;
     length o_&var $&dlength;
%end;
     o_&var=&var;
     t_var02=1;
     t_var03=_n_;
  run;

  proc sort data=t01;
     by &sortkey;
  run;


/*** Perform the forward pass if direction is forward or both. If &VAR is */
/*** is character create a temporary holding variable of the same length  */

  data t01;
     set t01;
     by &sortkey;
%if &dtype=CHAR %then %do;
     length t_var01 $&dlength;  %end;
     retain t_var01;
     retain t_var05 t_var06 t_var07;
     t_var04=(first.&groupvar)+2*(last.&groupvar);
     t_var03=_n_;
     if "%upcase(%substr(&dir,1,1))"='F' or "%upcase(%substr(&dir,1,1))"='B'
       then do;
       if t_var04 in (1,3) then do;  /* First VAR value in the group  */
         t_var01=&var;
         t_var05=.;  /* Initialize holding variables to missing */
         t_var06=.;
         t_var07=.;
       end;
%if &dtype=NUM %then %do;
       if t_var05=. then do;                    /* First value of VAR */
         t_var05=&var;
       end;
       if &var ne . and t_var05 ne . then do;  /* Difference from 1st VAR */
         t_var06=&var-t_var05;                  /* in the group */
       end;
       if &var ne . and t_var01 ne . then do;  /* Difference from latest */
         t_var07=&var-t_var01;                  /* missing VAR (difffwd) */
       end;
%end;
       if &var=&nullval then do;
        &var=t_var01;   /* Carry forward when VAR is null */
       end;
       else do;
        t_var01=&var;   /* Keep the latest non-null VAR to carry forward */
      end;
     end;
  run;


/*** Now sort in reverse order for carrying backwards */
```

```
   proc sort data=t01(drop=t_var01 t_var05);
     by descending t_var03;
   run;


/*** Perform the backward pass if direction is both or reverse. Repeat the */
/*** above steps but with the data set observations in the reverse order   */
  data t01;
    set t01;
    by descending t_var03;
 %if &dtype=CHAR %then %do;
    length o_&var t_var01 $&dlength;
 %end;
    retain t_var01;
    retain t_var05 t_var08 t_var09;
    if "%upcase(%substr(&dir,1,1))"='R' or "%upcase(%substr(&dir,1,1))"='B'
      then do;
      if t_var04 in (2,3) then do;
        t_var01=&var;
        t_var05=.;
      t_var08=.;
      t_var09=.;
      end;
 %if &dtype=NUM %then %do;
      if t_var05=. then do;                    /* Last value of VAR */
        t_var05=&var;
      end;
      if o_&var ne . and t_var05 ne . then do; /* Difference from last VAR */
        t_var08=o_&var-t_var05;                /* in the group (difflast) */
      end;
      if o_&var ne . and t_var01 ne . then do; /* Difference from next non-*/
        t_var09=o_&var-t_var01;                /* missing VAR (diffrev) */
      end;
 %end;
      if &var=&nullval then do;
        &var=t_var01;    /* Carry backward when VAR is null */
      end;
      else do;
        t_var01=&var;    /* Keep the latest non-null VAR to carry backward */
      end;
    end;
 %if &newvar ne  %then %do; /* Now assign the temporary values to the */
   &newvar=&var;            /* corresponding output parameters */
   &var=o_&var;
 %end;
 %if &diffirst ne t_var06 %then %do;
   &diffirst=t_var06;
 %end;
 %if &difffwd ne t_var07 %then %do;
   &difffwd=t_var07;
 %end;
 %if &difflast ne t_var08 %then %do;
   &difflast=t_var08;
 %end;
 %if &diffrev ne t_var09 %then %do;
   &diffrev=t_var09;
 %end;
  run;
```

```
/*** Recreate the input data set with the new columns or create the */
/*** specified output data set. Remove temporary variables         */

  proc sort data=t01 out=&outds(drop=t_var0: o_&var);
    by &sortkey;
  run;

  proc datasets nolist;                    /* Remove the temporary data set */
    delete t01;
  run;


%mend locfrb;
```

EXAMPLES

The following is a set of example macro calls used with sample PK data. The PK data consists of dosing observations, where an event ID indicator EVID is 1, and PK concentration sampling observations, where EVID=0. All dates and times are assumed to be valid SAS numeric dates and times and have been recorded in the correct chronological sequence. Below are the observations pertaining to one patient from a sample PK data set. The results of performing these examples are shown in bold figures in the table after Example 5.

Sample PK data set (PK01), for one selected patient (PATNUM)

PATNUM	EVNTDT	EVNTSTTM	EVNTDAY	EVID	DOSNO	DOSEQ	EXDOSE	EXDOSFRQ	EXDOSU	PCTPT	PCSTRESC	PCSTRESN	PCSTRESU	ETMU
203	2014-02-11	06:00:00	1	0	.	1	.		30 MIN PRE		LTR	.	ng/mL	hr
203	2014-02-11	10:00:00	1	1	1	1	20.0	QD	MG			.		hr
203	2014-02-11	12:00:00	1	0	.	1	.		2 HRS POST DOSE		68	68.0	ng/mL	hr
203	2014-02-11	14:00:00	1	0	.	1	.		4 HRS POST DOSE		80.6	80.6	ng/mL	hr
203	2014-02-11	16:00:00	1	0	.	1	.		6 HRS POST DOSE		80.4	80.4	ng/mL	hr
203	2014-02-11	18:00:00	1	0	.	1	.		8 HRS POST DOSE		57.3	57.3	ng/mL	hr
203	2014-02-11	20:00:00	1	0	.	1	.		10 HRS POST DOSE		50.5	50.5	ng/mL	hr
203	2014-02-12	05:10:00	2	0	.	1	.		30 MIN PRE		15.6	15.6	ng/mL	hr
203	2014-02-12	10:30:00	2	1	2	2	50.0	QD				.		hr
203	2014-02-12	18:30:00	2	0	.	2	.		8 HRS POST DOSE		245	245.0	ng/mL	hr
203	2014-02-13	10:30:00	3	1	3	3	50.0	QD				.		hr
203	2014-02-18	10:30:00	8	0	.	3	.		30 MIN PRE		116	116.0	ng/mL	hr
203	2014-02-18	13:15:00	8	1	4	4	100.0	QD				.		hr
203	2014-02-18	21:28:00	8	0	.	4	.		8 HRS POST DOSE		778	778.0	ng/mL	hr
203	2014-02-25	11:30:00	15	0	.	4	.		30 MIN PRE		87.9	87.9	ng/mL	hr
203	2014-02-25	12:20:00	15	1	5	5	570.0	QD				.		hr
203	2014-02-26	10:30:00	16	1	6	6	106.4	QD				.		hr

1. The columns EXDOSFRQ and EXDOSU are patient dependent but are required to be populated on every observation. The dosing frequency EXDOSFRQ is currently only populated on the EVID=1, dosing, observations and the dose unit EXDOSU is only present on the first dosing observation. To populate every observation these macro calls carry the values forward to successive observations and backward to missing preceding observations, hence DIR is 'B'.

6

```
%locfrb(  inds=pk01,   byvars=patnum,   groupvar=patnum,   var=exdosfrq,   dir=B   );
%locfrb(  inds=pk01,   byvars=patnum,   groupvar=patnum,   var=exdosu,   dir=B   );
```

Since these two columns are patient dependent and every observation is being populated, only PATNUM need be specified in BYVARS and GROUPVAR. Since no output data set is specified the changes overwrite the existing values in the input data set.

2. All the EVID=0 observations pertaining to a dose are on the same calendar day as the dose, there is a requirement to populate DOSNO on the EVID=0 observations for that day's dose. This is achieved with this macro call

```
%locfrb(  inds=pk01,   byvars=patnum evntday,   groupvar=evntday,   var=dosno,   dir=B   );
```

This time EVNTDAY (or EVNTDT) must be specified as the GROUPVAR, and be included after PATNUM in BYVARS.

3. Evaluate the change from baseline and change from the prior sample of each concentration sampling result (PCSTRESN), where baseline is the earliest non-missing post dose sample. To do this each PCSTRESN value has the baseline value subtracted from it, and each PCSTRESN value has the prior non-missing PCSTRESN value subtracted from it.

```
%locfrb(  inds=pk01,   byvars=patnum evntdt evntsttm,   groupvar=patnum,   var=pcstresn,
          newvar=resultcf,   diffirst=chbl,   difffwd=chg,   dir=F   );
```

This call sorts the data by the BYVARS PATNUM EVNTDT and EVNTSTTM to place the observations in chronological order and, since the sample results pertain to PATNUM, this column is the GROUPVAR. NEWVAR=RESULTCF creates a new numeric column called RESULTCF, which is the values of PCSTRESN carried forward over missing values. Specifying DIFFIRST=CHBL stores all the differences from the first non-missing value of PCSTRESN in the created column CHBL. DIFFFWD=CHG creates a column called CHG, which contains all the successive changes in PCSTRESN. Note that the first values for the patient will be missing because DIR='F', since CHBL and CHG values prior to the first non-missing PCSTRESN would be meaningless.

4. Evaluate the time the PREDOSE sample takes place before the corresponding dose (result will be negative) in hours. The time, as a SAS internal datetime, is the date EVNTDT and the time EVNTSTTM consolidated in to a SAS datetime EVNTSDTM (This would also allow for samples pertaining to a given dose being on different days). Now the successive differences between the current and next observation will be the 'time before' difference. This time difference is stored in the new variable NEXTTIME specified as DIFFREV. (DIFFFWD would store the time after instead of before). DIR is set to as 'R' since the PREDOSE sample is before and not after the dose sample.

```
%locfrb(  inds=pk01,   byvars=patnum dosno evntdt evntsttm,   groupvar=dosno,   var=evntsttm,
          diffrev=nexttime,   dir=R   );
```

Since NEXTTIME is the time difference in seconds, it is converted into hours and stored as NEXTTMHR in the following DATA step (and can be set to missing at the non-predose samples if need be).

```
data pk01;
  set pk01;
  nxttmhr=nexttime/3600;
run;
```

5. Obtain the actual time since the most recent dose (ACTTMMRD) and the actual time from first dose (ACTTMFDS) for each patient. DOSEQ (not DOSNO) is the preceding dose number (The PREDOSE sample has the same DOSEQ as the prior dose and not the dose about to occur.). When each succession of observations with the same DOSEQ is clustered together and sorted by DESCENDING EVID the EVID=1 dosing observation becomes the first of the cluster, hence the difference between this and each associated EVID=0 sampling observation (ACTTMMRD) is now easily obtainable using DIFFIRST carrying forward (DIR='F'). EVNTSDTM is the date EVNTDT and the time EVNTSTTM consolidated into a SAS datetime.

> %locfrb(inds=test04, byvars=patnum doseq descending evid, groupvar=doseq, var=evntsdtm, diffirst=acttmmrd, dir=F);

Obtaining ACTTMFDS uses the same method as for ACTTMMRD except the time differences are measured from the first dose for the patient instead of each DOSEQ. Hence GROUPVAR is PATNUM instead of DOSEQ

> %locfrb(inds=test04, byvars=patnum doseq descending evid, groupvar=patnum, var=evntsdtm, diffirst=acttmfds, dir=F);

The only additional code needed is no convert these results from seconds to hours.

PATNUM	EVNTDT	EVNTSTTM	EVNTDAY	EVID	DOSNO	DOSEQ	EXDOSE	EXDOSFRQ	EXDOSU	PCTPT	PCSTRESC	PCSTRESN	PCSTRESU	ETMU
203	2014-02-11	06:00:00	1	0	**1**	1	.	**QD**	**MG**	30 MIN PRE	LTR	.	ng/mL	hr
203	2014-02-11	10:00:00	1	1	1	1	20.0	QD	MG			.		hr
203	2014-02-11	12:00:00	1	0	**1**	1	.	**QD**	**MG**	2 HRS POST DOSE	68	68.0	ng/mL	hr
203	2014-02-11	14:00:00	1	0	**1**	1	.	**QD**	**MG**	4 HRS POST DOSE	80.6	80.6	ng/mL	hr
203	2014-02-11	16:00:00	1	0	**1**	1	.	**QD**	**MG**	6 HRS POST DOSE	80.4	80.4	ng/mL	hr
203	2014-02-11	18:00:00	1	0	**1**	1	.	**QD**	**MG**	8 HRS POST DOSE	57.3	57.3	ng/mL	hr
203	2014-02-11	20:00:00	1	0	**1**	1	.	**QD**	**MG**	10 HRS POST DOSE	50.5	50.5	ng/mL	hr
203	2014-02-12	05:10:00	2	0	**2**	1	.	**QD**	**MG**	30 MIN PRE	15.6	15.6	ng/mL	hr
203	2014-02-12	10:30:00	2	1	2	2	50.0	QD	**MG**			.		hr
203	2014-02-12	18:30:00	2	0	**2**	2	.	**QD**	**MG**	8 HRS POST DOSE	245	245.0	ng/mL	hr
203	2014-02-13	10:30:00	3	1	3	3	50.0	QD	**MG**			.		hr
203	2014-02-18	10:30:00	3	0	**4**	3	.	**QD**	**MG**	30 MIN PRE	116	116.0	ng/mL	hr
203	2014-02-18	13:15:00	8	1	4	4	100.0	QD	**MG**			.		hr
203	2014-02-18	21:28:00	8	0	**4**	4	.	**QD**	**MG**	8 HRS POST DOSE	778	778.0	ng/mL	hr
203	2014-02-25	11:30:00	15	0	**5**	4	.	**QD**	**MG**	30 MIN PRE	87.9	87.9	ng/mL	hr
203	2014-02-25	12:20:00	15	1	5	5	570.0	QD	**MG**			.		hr
203	2014-02-26	10:30:00	16	1	6	6	106.4	QD	**MG**			.		hr

EVNTSDTM	RESULTCF	**CHBL**	**CHG**	NEXTTIME	**NXTTMHR**	**ACTTMFDS**	**ACTTMMRD**
2014-02-11T06:00:00	.	.	.	-14400	**-4.00000**	**-4.000**	**-4.000**
2014-02-11T10:00:00	.	**.**	**.**	-7200	**-2.00000**	**0.000**	**0.000**
2014-02-11T12:00:00	68.0	**0.0**	**.**	-7200	**-2.00000**	**2.000**	**2.000**
2014-02-11T14:00:00	80.6	**12.6**	**12.6**	-7200	**-2.00000**	**4.000**	**4.000**
2014-02-11T16:00:00	80.4	**12.4**	**-0.2**	-7200	**-2.00000**	**6.000**	**6.000**
2014-02-11T18:00:00	57.3	**-10.7**	**-23.1**	-7200	**-2.00000**	**8.000**	**8.000**
2014-02-11T20:00:00	50.5	**-17.5**	**-6.8**	0	**0.00000**	**10.000**	**10.000**
2014-02-12T05:10:00	15.6	**-52.4**	**-34.9**	-19200	**-5.33333**	**19.167**	**19.167**
2014-02-12T10:30:00	15.6	**-52.4**	**-34.9**	-28800	**-8.00000**	**24.500**	**0.000**
2014-02-12T18:30:00	245.0	**177.0**	**229.4**	0	**0.00000**	**32.500**	**8.000**
2014-02-13T10:30:00	245.0	**177.0**	**229.4**	0	**0.00000**	**48.500**	**0.000**
2014-02-18T10:30:00	116.0	**48.0**	**-129.0**	-9900	**-2.75000**	**168.500**	**120.000**
2014-02-18T13:15:00	116.0	**48.0**	**-129.0**	-29580	**-8.21667**	**171.250**	**0.000**
2014-02-18T21:28:00	778.0	**710.0**	**662.0**	0	**0.00000**	**179.467**	**8.217**
2014-02-25T11:30:00	87.9	**19.9**	**-690.1**	-3000	**-0.83333**	**337.500**	**166.250**
2014-02-25T12:20:00	87.9	**19.9**	**-690.1**	0	**0.00000**	**338.333**	**0.000**
2014-02-26T10:30:00	87.9	**19.9**	**-690.1**	0	**0.00000**	**360.500**	**0.000**

CONCLUSION

The LOCFRB macro has the potential to be a very useful utility department level macro, which reduces the amount of SAS code and programming effort to achieve multiple common programming challenges involving the relationship between the values within a data set column.

ACKNOWLEDGMENTS

The author acknowledges the support and resources of Dataceutics, inc. in the compilation and verification of this paper and for sponsoring its presentation at PharmaSUG 2017. The support and dedication of PharmaSUG 2017 "Beyond the Basics" Section Chair is also greatly appreciated.

CONTACT INFORMATION

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

Timothy J. Harrington
Dataceutics, inc. 1610 Medical Drive, Pottstown, PA 19464
(610) 970 2333
harringt@dataceutics.com
www.dataceutics.com

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.