Paper DM07

A Simple Macro to Flag New/Changed Records

Zhuo Chen, PPD, Austin, TX
David Gray, PPD, Austin, TX

ABSTRACT
In a long clinical study or a study with many subjects, listing reviews may be needed many times during the study. To gain efficiency, a reviewer would like to know what new or changed records have emerged since the last run of the listings.

This paper describes a simple macro developed in SAS® that flags new or changed records in the listing output. This flag will identify the new or changed records since the last run. This macro provides an option to output only new and changed records rather than a cumulative listing.

INTRODUCTION
This paper presents one simple macro that can flag new or changed records since last time the listing was run. It looks at each record in your listing output, if at least one variable in that record has been changed, then flags this as a new/changed record.

SAS MACRO %LISTING_FLAG
This macro has 3 input parameters,
- dsn: is the final data set used by proc report, a new variable named YYYYFLAG will be added into this data set after this macro call;
- varlist: is the list of all variables in the output (those variables after “column” in proc report). If at least one of them changed, then it is a new/changed record;
- onlyflag=: if it is Y (by default), then the listing only output new/changed records; otherwise, the listing is cumulative with all new/changed records flagged as "<<<<<<".

This macro first saves the current final data set under CURRENT folder, then it searches the PREVIOUS folder to see whether a previous SAS data set exists (with the same name as the program which calls this macro). If a previous data set does exist then this macro will read the previous data set and merge it with current data set by all the variables defined in the varlist. Then the flag variable YYYYFLAG will be set to '"<<<<<<" for records in current data set but not in previous data set. If the data set does not exist in the PREVIOUS folder, then the flag variable YYYYFLAG will be set to '"<<<<<<" for all records in the current data set. This macro requires that the program be run in batch mode.

%macro listing_flag(dsn,varlist,onlyflag=y);
  * Step 1: define folder to put current output data sets under CURRENT, put last run data sets under PREVIOUS ;
  libname CURRENT  ".\CURRENT" ;
  libname PREVIOUS ".\PREVIOUS" ;
  * Step 2: get SAS program name from automatic macro variable sysprocessname ;
  data_null ;
  dummycount= count(%unquote(%qscan(&sysprocessname,2,' ')),'%');
  call symput('dummycount',dummycount);
  run;
  %let dummy    = %qscan(&sysprocessname,%eval(&dummycount)+1,'\');
  %let locate   = %index(&dummy,.);
  %let filename = %substr(&dummy,1,%eval(&locate-1));
%endoftenlisting_flag;
* Step 3: save current data set under CURRENT;
data CURRENT.&filename ;
    set &dsn ;
run;

proc sort data = &dsn out = current ;
    by &varlist ;
run ;

* Step 4: if previous data set exist under PREVIOUS folder, then compare with previous data set, set variable yyyyflag=<<<< if this is a new or changed record. If onlyflag=y as default, then only output new or changed record;
%if %sysfunc(exist(PREVIOUS.&filename))  %then
    %do ;
    proc sort data = PREVIOUS.&filename out = previous nodupkey;
        by &varlist;
    run;
    data &dsn;
        merge previous(in=a) current(in=b) ;
            by &varlist;
        length yyyyflag $15. ;
        if b * ^a  then yyyyflag = "<<<<";
        else yyyyflag = " ";
        if b ;
    run;
%if &onlyflag EQ y %then %do;
    data &dsn;
        set &dsn (where=(yyyyflag NE ''));
    run;
%end;
%end; /* Step 5: if previous data set does not exist under PREVIOUS folder, then set variable yyyyflag=<<<< for all records; */
%else %do;
    data &dsn;
        set &dsn ;
        length yyyyflag $15. ;
        yyyyflag = "<<<<";
    run;
%end;
%mend listing_flag ;

OUTPUT EXAMPLES

CUMULATIVE OUTPUT WITH NEW/CHANGED RECORD FLAGEED
The first output example shows the FLAG="<<<<" for new and changed records since last run. This is a cumulative listing.
The second output example shows that after setting the option `onlyflag=y` in macro `%listing_flag`, the report only shows the new and changed records.

**CONCLUSION**

The method presented here for flagging new and changed records can dramatically reduce the time needed for iterative listing reviews. Such a method is highly recommended for large clinical trials.

**ACKNOWLEDGMENTS**

We would like to thank our colleagues John Gorden and Michael Galvan in the Biostatistics Department at PPD who provided comments on drafts of this paper.

**CONTACT INFORMATION**

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

David Gray, Zhuo Chen
PPD
7551 Metro Center Drive, Suite 300
Austin, TX 78744
E-mail: david.gray@austin.ppd.com, zhuo.chen@austin.ppd.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.