

Code Review of A Macro from Observations

H. Ian Whitlock, Westat Inc., Rockville, MD

Introduction

In [1] Dooley and Todd consider computer efficiency when joining data sets. They compare PROC SQL to a macro, DAMERGE, which uses DATA steps and the POINT option of the SET statement to do the join.

Their DAMERGE code included an unneeded DATA step. Consequently, their conclusion that DAMERGE is preferable as the problem size increases could be made more strongly.

This paper suggests efficiency and user friendly improvements to their DAMERGE code.

Test Data

Removing the extra DATA step resulted in approximately a 20% reduction of CPU time when tested with SAS 6.10 under Windows 3.1 on a Compaq Prolinea 4/50. The reduction in time is based using on test data generated by

```
data lab ( keep = id labdate a1 - a20 )
    patient ( keep = id begtest endtest
              drug b1 - b20 ) ;
    retain a1 - a20 1 b1 - b20 2 ;
    begtest = '1jan94'd ;
    endtest = '10jan94'd ;
    format begtest endtest labdate date7. ;

do id = 1 to 2000 ;
    do drug = 'a', 'b', 'c', 'd', 'e' ;
        if ranuni ( 0 ) < .7 then output patient ;
    end ;
    do i = 1 to 5 ;
        labdate = '1jan94'd + int (ranuni(0)*20);
        output lab ;
    end ;
end ;
run ;
```

User Friendly Improvements

Ease of use improvement include:

- 1) BREAK_BY is calculated as the last variable in &COM_SORT.
- 2) SRC_SORT renamed XTR_SORT includes only variables not already listed under COM_SORT.
- 3) WRK_DS is eliminated, since the user doesn't care.
- 4) CRITERIA simplified to just the condition to be evaluated.
- 5) OUT_DS added to give user control of output data set name.
- 6) SORTMETH gives the user control of the sort process.
- 7) Double underscore used to avoid name conflicts with user variables.

Suggested Revision

```
%macro damerge
(  src_ds = , /* data to be looked up */
  trg_ds = , /* target lookup file */
  out_ds = , /* final output file */
  criteria = , /* just the condition */
  sortmeth = INPLACE ,
              /* WORK or NONE */
  com_sort = , /* common sort order */
  xtr_sort = /* extra sort vars */
) ; /* for &src_ds */

%local i word break_by ;
%let sortmeth =
    %upcase(%substr(&sortmeth,1,1)) ;
%let i = 1 ;
%let word = %scan ( &com_sort , &i ) ;
%do %while ( %quote ( &word ) ^= ) ;
    %let break_by = &word ;
    %let i = %eval ( &i + 1 ) ;
    %let word = %scan ( &word , &i ) ;
%end ;

%if &sortmeth = I %then
%do ;
    proc sort data = &src_ds ;
        by &com_sort &xtr_sort ;
    run ;
```

```

proc sort data = &trg_ds ;
  by &com_sort ;
run ;
%end ;
%else %if &sortmeth = W %then
%do ;
  proc sort data = &src_ds out = __src ;
    by &com_sort &xtr_sort ;
  run ;
  %let src_ds = __src ;
  proc sort data = &trg_ds out = __trg ;
    by &com_sort ;
  run ;
  %let trg_ds = __trg ;
%end ;

/* get pointers to &trg_ds */
data __ptrs (keep = &com_sort __beg __end);
  retain __beg 1 ;
  set &trg_ds ( keep = &com_sort ) ;
  by &com_sort ;
  if last.&break_by then
  do ;
    __end = _n ;
    output __ptrs ;
    __beg = __end + 1 ;
  end ;
run ;

data __dummy ;
  output ;
  stop ;
  set &trg_ds ( drop = &com_sort ) ;
run ;

/* do the look-up */
data &out_ds ( drop = __: ) ;
  merge   &src_ds
         __ptrs ( in = __hasptr )
  ;
  by &com_sort ;

  if __hasptr then
  do ;
    do __ptr = __beg to __end ;
      set &trg_ds point = __ptr ;
      if &criteria then

```

```

do ;
  __match = 1 ;
  output &out_ds ;
end ;
end ;
end ;
if __match = . then
do ;
  retain __ptr2 1 ;
  set __dummy point = __ptr2 ;
  output &out_ds ;
end ;
run ;

proc datasets lib = work ;
  delete
    %if &sortmeth = W %then __src __trg ;
    __ptrs __dummy
  ;
run ;
%mend damerge ;

```

Conclusion

Code review is interesting and usually leads to improvements. In correspondence Dooley and Todd noted that their macro was a cut down version of a more complex in house macro that was intended for test purposes only.

The author can be contacted by mail or e-mail

H. Ian Whitlock
Westat
1650 Research Boulevard
Rockville, MD 20850-3129

Whitloi1@westat.com

[1] Kathleen Dooley and Michael Todd, "A Performance Evaluation of Two Techniques for Many-to-Many Merges", *Observations*, Fourth Quarter 1994

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.