```
%macro MR_ADD_BY_REC(
   in_data       =
  ,sortvars      =
  ,byvars        =
  ,byvarstext    =
  ,byvarsfmt     =
  ,ordervar      =
  ,indentspace   = 2
  ,addblankrow   =
  ,out_data      =
  ,colname       =
  ,debug =
  ,help =
          ) / store source des='V1.0.0.5' ;

%*************************************************************************
**********************
* FILENAME:    MR_ADD_BY_REC.SAS
* DEVELOPER:   (b) (6)
* PLATFORM:    SAS 9.1.3, 9.2 on PC
* MACROS USED: mu_check_req_parameters, mu_check_data_and_var_exist, mu_wordscan,
ml_order (optional)
*              md_workinfo              md_clean_and_reset
* ASSUMPTIONS:
*
*
* DESCRIPTION:
*  This macro will
*  1. create a row for each variable entered in BYVARS containing a description of
the BYVARS variable and an ordering number
*  2. If no ordering variable is passed, one will be created based upon SORTVARS
ordering and it will be named _ORDER1_
*  3. Upon demand will created _indent_ variable needed for MR_PACK functionality
*
*
* USAGE NOTES:
*
* PARAMETERS:
* IN_DATA         = the input data set (REQUIRED)
* OUT_DATA        = the output data set.  (DEFAULT = &SYSMACRONAME) (CONDITIONAL)

* SORTVARS        = variable(s) on IN_DATA that will be used to sort data for
creation of BYVAR lines
*                   (should mimic order for display) (REQUIRED)
* BYVARS          = variable(s) on IN_DATA for which an extra row is to be added
for each variable listed. (REQUIRED)
*                   Varibles entered in BYVARS MUST also be listed in SORTVARS
*                   Order the BYVARS as they are to appear on Display.
* BYVARSTEXT      = variable(s) on IN_DATA containing descriptive text to use as
BYVAR values for added rows to be used
```

```
*                   when BYVARS variables are only numeric or contain an
abbreviated text value
* BYVARSFMT       = Enter existing formats to be used for BYVARS variables when no
descriptive text variables exist in data set
*                   There MUST be a one to one correspondence with each variable in
byvars
*                   Enter placeholder BLANK for variables that do not need to be
formatted when using this parameter.
*                   eg:  BLANK $xyz BLANK
* COLNAME         = Variable on IN_DATA to which BYVARS text is to be
added.(REQUIRED)  Usually the COL1 field.
* ORDERVAR        = Variable on IN_DATA containing a numeric ordering value for
printing of rows on display.
*                   If data set does not contain an ordering variable
*                   Leave blank (DEFAULT) to have ordering variable (_ORDER1_)
created
* ADDBLANK        = Add a blank line [B]efore or [A]fter a BYVAR row addition.
*                   If used, enter value for each BYVARS variable.  Use BLANK as a
placeholder if multiple BYVARS variable are
*                   entered, but not all are to have a blank line added.
*                   eg: BLANK BLANK B
*                   Null (DEFAULT) does not create any extra blank lines
* INDENTSPACE     = Indent value to be set to _INDENT_ variable created inside this
macro and passed to MR_PACK by independant
*                   process to occur after execution of this macro.
*                   2 (DEFAULT).  Set to null if no indenting is desired and
_INDENT_ variable will not be created.
*                   If multiple BYVARS are entered, sub-BYVARS will also be
indented by value entered and section rows will be
*                    indented incrementally using the INDENTSPACE as a factor.
*                    eg: there are 3 BYVARS, the first BYVAR has _indent_ = 0
*                     all section rows under this BYVAR have _indent_ =2
*                     The 2nd BYVAR will also have a _indent_ value of 2 and all
rows under the BYVAR will have _indent_ =4
*                     The 3rd BYVAR will also have a _indent_ value of 4 and all
rows under the BYVAR will have _indent_ =6
*********************************************************************
* DATE      INITIALS    MODIFICATION DESCRIPTION
*********************************************************************
*
*********************************************************************
*  © 2011 Pharmaceutical Product Development, Inc.
*  All Rights Reserved.
*********************************************************************************
*************************************;


%local  i blcnt reorder usetext usefmt indentit indent0 indentr _col1width_;
%global &SYSMACRONAME._RC;
```

```
%*-------------------------------------------------------------------------------
--*;
  %PUT ------------------------------------------------ ;
  %PUT INFO: (&SYSMACRONAME) ;
  %PUT INFO: Version 1.0 ;
  %PUT -START------------------------------------------ ;
  %let &SYSMACRONAME._RC =0;
%*-------------------------------------------------------------------------------
--*;


%mu_help_debug;

%* Get lists of work data sets and views, and mprint and fmterr settings;
%md_workinfo(
      debug        = &debug
    ,_workdata     = WORK_DATASETS_DATA
    ,_workview     = WORK_DATASETS_VIEW
    ,_mprinttoggle = mprint_setting
   );

%*
-------------------------------------------------------------------------------
----;
%* Step 1: Check required parameters were passed, Check variables exist in data
set;
%*         and other checking processing ;
%*
-------------------------------------------------------------------------------
----;

%mu_check_req_parameters(
   parameters_to_check = IN_DATA SORTVARS BYVARS COLNAME
  ,help                = no
    );

%* ----------------IN_DATA AND REQUIRED VARIABLES EXIST ------------------;
%mu_check_data_and_var_exist(
   data_to_check                    = &IN_DATA
  ,vars_to_check_in_all_data        = &SORTVARS &BYVARS &COLNAME &ORDERVAR
  ,vars_to_check_in_respective_data =
  ,abort_if_does_not_exist          = yes
  ,help = no
    );

%if %bquote(&OUT_DATA) = %str() OR %bquote(&OUT_DATA) = %bquote(&IN_DATA) %then
%do;
    %if %bquote(&OUT_DATA) = %str() %then %put ALERT_I: Missing parameter OUT_DATA.
 Default to &SYSMACRONAME;
    %else                               %put ALERT_I: OUT_DATA is the same as
```

```
IN_DATA, will be changed to &SYSMACRONAME;

    %let OUT_DATA = &SYSMACRONAME;
    %let &sysmacroname._RC = 1;
%end;


%else %if %sysfunc(exist(&OUT_DATA)) %then %do;
    %put ALERT_I: OUT_DATA = &OUT_DATA.  Data set already exists and will be
deleted/overwritten;
    %let &sysmacroname._RC = 2;
    proc datasets mt=data lib=work;
         delete &OUT_DATA;
    quit;
%end;


%*
-------------------------------------------------------------------------------
-----;
%* Step 2: If order variable (ORDERVAR) was passed, sort data into intermediate
data set.                          ;
%*         If no order variable was provided, run ML_ORDER to create order variable
and sort data.
%*
-------------------------------------------------------------------------------
-----;
%if %bquote(&ordervar) = %str( ) %then %do;
    %let ordervar = _order1_;
    %ml_order(
         in_data               = &IN_DATA
        ,IN_WHERE              =
        ,sort_order_vars       = &sortvars
        ,sort_order_asc_or_desc = A
        ,section               =
        ,skipvars              = &byvars
        ,convert               = YES
        ,out_data              = skipit
              );

    %* - ml_order sometimes adds a variable name DUMMYVAR to the BYVARS macro
variable - remove it -;
    %let byvars = %sysfunc(tranwrd(%bquote(&byvars), %str(DUMMYVAR), %str( ) ));

%end;


%else %do;
   proc sort data=&IN_DATA out=skipit;
     by &sortvars;
```

```
%end;

%if &debug = Y %then %do;
    %if &ordervar = _order1_ %then %str( title 'Dump of select data after running
ML_ORDER_';);
    %else %str( title 'Dump of select data after initial sort';);
    proc print data=skipit;
      var &sortvars &ordervar &colname;
    run;
%end;

%*
------------------------------------------------------------------------------
----;
%* Step 3: Determine number of BYVARS variables specified
    ;
%*         Review values of BYVARSTEXT and BYVARSFMT if entered ;
%*
------------------------------------------------------------------------------
----;
    %MU_WORDSCAN(string =%str(&byvars)
                ,root   =byvar
                ,numw   =nsecs
                   );

    %let byvarstext = %upcase(%bquote(&byvarstext));
    %if %sysfunc(tranwrd(%bquote(&byvarstext), %str(BLANK), %str( ) )) ne %str( )
%then %do;
        %let usetext = 1;
        %MU_WORDSCAN(string =%str(&byvarstext)
                    ,root   =text
                    ,numw   =ntext
                   );
    %end;


    %let byvarsfmt = %upcase(%bquote(&byvarsfmt));
    %if %sysfunc(tranwrd(%bquote(&byvarsfmt), %str(BLANK), %str( ))) ne %str( )
%then %do;
        %let usefmt = 1;
        %MU_WORDSCAN(string =%str(&byvarsfmt)
                    ,root   =fmt
                    ,numw   =nfmt
                   );
    %end;

    %* -- Determine col1 width and enlarge if necessary to accomodate textfield or
format to be applied;
    %* Get current col1 width ( dsname_variablename_len);
    %mu_var_attributes
```

```
          ( datasets   = &in_data
           ,variables  = &colname
           ,debug      = &debug
           );

    %let indata = %sysfunc(translate(%bquote(&in_data), %str(_), %str(.) ));

    %let _col1width_ = &&&indata._&colname._len ;


    %* -- Loop thru the methods of providing a value for the byvar row -----;
    %* -- the macro variable(s) TEXTn will hold Value to be applied later  -;

    %do i = 1 %to &nsecs;     %* Loop thru BYVARS variables;

        %if %str(&usetext) = 1 %then %do;
            %if &ntext >= &i %then %do;
                %if %bquote(&&text&i) = %str(BLANK) %then %let text&i = ;
            %end;
            %else %if &nsecs > &ntext %then %let text&i =;
        %end;
        %else %let text&i =;


        %if %str(&usefmt) = 1 %then %do;
            %if &nfmt >= &i %then %do;
                %if %bquote(&&fmt&i) = %str(BLANK) %then %let fmt&i =;
            %end;
            %else  %let fmt&i =;

            %if %bquote(&&fmt&i) ne %str( ) %then %do;

                %if %bquote(&&text&i) ne %str( ) %then %do;
                    %* -- Formats have been provided (BYVARSFMT) for byvars, apply
them to byvar(s) ----;
                    %put ALERT_I:
----------------------------------------------------------------------;
                    %put ALERT_I: BOTH a BYVARSTEXT (&&text&i) and a BYVARSFMT
(&&fmt&i) value have been added for this variable (&&byvar&i) ;
                    %put ALERT_I: The BYVARSFMT value will be used.;
                    %put ALERT_I:
----------------------------------------------------------------------;
                %end;

                %let fmtstring = %sysfunc(translate(&&fmt&i, %str( ), %str(.)));
%*strip period if entered;
                %let text&i    = %bquote(put(&&byvar&i, &fmtstring..) );
            %end;
        %end;
```

```
        %* -- No text variables or formats were provided, use byvars variables;
        %if %bquote(&&text&i) = %str( ) %then %let text&i = &&byvar&i;

        %* -- Determine &colname variable width if provided text and/or format ;
        %let textlength = %length(&&text&i);
        %if &textlength > &_col1width_ %then %let _col1width_ =  &textlength;

    %end;

%*
-------------------------------------------------------------------------------
-----;
%* -Step 4: Review AddBlankRow to see if requested and set up processing
-----------------;
%*-- Ensure there is a value for each byvar parameter -;
%*
-------------------------------------------------------------------------------
-----;
 %if %bquote(&addblankrow) ne %str( ) %then %do;
     %MU_WORDSCAN(string =%str(&addblankrow)
                 ,root   =blrow
                 ,numw   =nbl
                   );

     %if &nsecs > &nbl %then %do;  %* More byvars have been entered than breaks;
         %do i = 1 %to &nbl;
             %let blrow&i = %bquote(&&blrow&i);
         %end;
         %do i = %eval(&nbl + 1) %to &nsecs;
             %let blrow&i = BLANK;
         %end;
     %end;

     %do i = 1 %to &nbl;
         %let blrow&i = %upcase(%bquote(&&blrow&i));
         %if %bquote(&&blrow&i) = B or %bquote(&&blrow&i) = A %then %let blcnt =
%eval(&blcnt + 1);
     %end;
 %end;


 %* -- No value entered, create dummy macro variables for addblankrow---;
 %else %do i = 1 %to &nsecs;
     %local blrow&i;
     %let blrow&i = BLANK;
 %end;


%*
-------------------------------------------------------------------------------
```

```
-----;
%* -Step 5:Create variable containing indenting value to be passed to MR_PACK
macro;
%*
-------------------------------------------------------------------------------
-----;

 %if %bquote(&indentspace) ne %str( ) %then %do;
     %if %sysfunc(verify(&indentspace, %str(0123456789))) > 0 %then %let
indentspace = 2;
     %let indentit = 1;
     %let indentr  = %str( _indent_ = &indentspace * &nsecs );
     %let indent0  = %str( _indent_ = 0 );
 %end;


%*
-------------------------------------------------------------------------------
-----;
%* Step 6: Create rows for byvar variables and populate COLNAME and ORDERVAR values
   ;
%*
-------------------------------------------------------------------------------
-----;

  %let reorder = 1;

  data skipit;
   length &colname $206;
   set skipit;
    by &sortvars;

    &indentr;
    output;

    %* -- Loop thru each break variable to be printed on its own line;
    %do i = &nsecs %to 1 %by -1;

        %* -- Create indenting value for byvar rows, based upon # of byvars ---;
        %if &indentit = 1 %then %do;
            %if &i > 1 %then %str( _indent_ = (&i -1 ) * &indentspace; );
            %else            %str( _indent_ = 0;);
        %end;

        %* -- Create ordering value for byvars -------;
        %let reorder = %eval(&reorder - 1);

        if first.&&byvar&i then do;
           &ordervar = &reorder;
```

```
          %* -- Add blank rows [A]fter each byvar--;
          %if %bquote(&&blrow&i) = A %then %do;
              &indent0;
              &colname = ' ';
              output;
              &ordervar = &ordervar - 1;
              %let reorder = %eval(&reorder - 1);
          %end;

          &colname = &&text&i;
          output;

           %* -- Add blank rows [B]efore each byvar--;
          %if %bquote(&&blrow&i) = B %then %do;
              &indent0;
              &ordervar = &ordervar - 1;
              &colname = ' ';
              output;
              %let reorder = %eval(&reorder - 1);
          %end;

       end;

    %end;
  run;

  %* --Remove data from added records -------------------------;
  data skipit newrows(keep = &sortvars &byvars  &ordervar &colname _indent_);
   set skipit;

       if &ordervar < 1 then output newrows;
       else output skipit;
  run;

  data skipit;
   set skipit newrows;
  run;


 %*
-------------------------------------------------------------------------------
-----;
 %* - Step 6: Re Sort so newly added rows are in correct order ----;
 %*
-------------------------------------------------------------------------------
-----;
 proc sort data=skipit out = &OUT_DATA;
  by &sortvars  &ordervar ;
 run;
```

```
 %if &debug = Y %then %do;
     title 'data with BYVAR header row added';
     proc print;
          var &sortvars &byvars  &ordervar &colname
          %if %bquote(&indentspace) > 0 %then %str( _indent_ );
            ;
     run;
 %end;

%*
-------------------------------------------------------------------------------
-----;
%* Step 7 - clean up and reset options (pulled by md_workinfo) ;
%*
-------------------------------------------------------------------------------
-----;

 %md_clean_and_reset(
      debug      = &debug
    ,_workdata   = %str(&WORK_DATASETS_DATA &out_data)
    ,_workview   = %str(&WORK_DATASETS_VIEW)
    ,resetmprint = &mprint_setting
      );


 %PUT ------------------------------------------------ ;
 %PUT INFO: (&SYSMACRONAME) ;
 %PUT INFO: Version 1.0 ;
 %put &SYSMACRONAME._RC = &&&SYSMACRONAME._RC;
 %PUT -END----------------------------------------- ;

%mend MR_ADD_BY_REC;
```