```
%macro ma_n_pcnt(
 in_data =
,in_numerator = n
,in_denom_data = bign
,in_denom_var = denom
,in_denom_merge_vars = &_default_trtvar

,subgroup =

,display_option = 1
,pcnt_for_zero_num =No
,no_pcnt_if =
,out_n_pcnt_format = &_default_out_n_pcnt_format
,trigger_pcnt_sign = &_default_trigger_pcnt_sign
,trigger_tight_paren = &_default_trigger_tight_paren
,trigger_n_space_pcnt = &_default_trigger_n_space_pcnt
,trigger_100_with_decimal = YES

,additional_var =
,no_additional_var_if =
,additional_var_format =
,additional_var_brackets = []
,additional_var_delim =
,fraction_bar = /
,out_display_var_split = no

,blank_for_zero_denom = &_default_blank_for_zero_denom

,out_data = N_PCNT
,out_pcnt = pcnt
,out_display_var = n_pcnt

,escapechar=
,prcode_split=
,rtf_ascii =

,debug =
,help =
) / store source des='V1.0.0.13' ;

%*************************************************************************
        PROGRAM NAME:   MA_N_PCNT.SAS
             VERSION:   1.0
   DIRECTORY/ACCOUNT:
              AUTHOR:   (b) (6)
        DATE CREATED:   20080429
             PURPOSE:   This macro will produce percentages of n/N where N is
obtained from
                        the denominator dataset (default name of denominator
dataset is "BIGN")
```

```
          CHECKED BY:
  SUPPORT PROGRAMMER:
             UPDATED:
             CHECKED:
         INPUT FILES:
        OUTPUT FILES:
  CALLING PROCEDURES:   called by table program
   CALLED PROCEDURES:   mu_wordscan, mu_check_req_parameters,
mu_check_data_and_var_exist,
                                                mu_nobs
  CREATED PROCEDURES:
--------------------------------------------------------------------------------
Modification History
Date    :               Programmer:             Notes:

--------------------------------------------------------------------------------
USAGE NOTES:
IN_DATA: The name of the dataset with counts going IN to the macro.
IN_NUMERATOR: The variable containing the n statistic in counts dataset -- ie, the
numerator

IN_DENOM_DATA: Dataset that will be merged to get percents. (Default = BIGN)
(REQUIRED)
IN_DENOM_VAR: What is the denominator variable in dataset &IN_DENOM_DATA?
        (Default = DENOM) (REQUIRED)
IN_DENOM_MERGE_VARS: variable(s) to merge the numerator data with denominator data
(Default=&_TRTVAR) (REQUIRED)

SUBGROUP: The variable(s) to use in the BY statement when merging the denominator
dataset
          with the counts dataset (&IN_DATA). (default = blank) (OPTIONAL)

DISPLAY_OPTION: Value should be an integer from 0 through 3 (more options will be
added
        as deemed necessary).

        OPTION 0:  n
        OPTION 1:  n (x.xx[%])
        OPTION 2:  n/N (x.xx[%])
        OPTION 3:  n (x.xx[%]) [xx]
        OPTION 4:  n/N (x.xx[%]) [xx]

PCNT_FOR_ZERO_NUM: Whether or not to display percent when number is zero. It takes
value YES or NO. Default=NO
NO_PCNT_IF: Subsetting clause which identifies which records do not display
percentage.
        Example: _SECTION_ = 1
OUT_N_PCNT_FORMAT: The format for the percentage, eg, 7.3, 6.2, 5.1... May be a
user-defined format
TRIGGER_PCNT_SIGN: Whether or not to display the percent sign (%) in the
```

```
out_display_var
        variable
TRIGGER_TIGHT_PAREN: Whether or not to tighten the parentheses around the percent
portion of the display variable.
        If YES then display will appear as
                n (pp.p)
        If NO then display will appear as
                n( pp.p)
TRIGGER_N_SPACE_PCNT: Whether or not to add a space between the n and pcnt

ADDITIONAL_VAR: If DISPLAY_OPTION = 3, then this parameter is required. This
variable will be
        displayed within brackets (see OPTION 3 above).
NO_ADDITIONAL_VAR_IF: Subsetting clause which identifies which records do not
display
        the additional variable.
        Example: _SECTION_ = 1
ADDITIONAL_VAR_FORMAT: If ADDITIONAL_VAR is numeric, then this parameter is
required.
        For example, if the additional variable is a pvalue, then the format might
be pvalue6.4.
OUT_DISPLAY_VAR_SPLIT: (default=no) Only has effect if DISPLAY_OPTION = 3.  If set
to YES,
        then ADDITIONAL_VAR will be displayed below n (x.xx[%]), in the same cell.


OUT_DATA: The name of the dataset coming OUT of the macro. Default = N_PCNT
OUT_PCNT: The name of the numeric variable created as 100*n/N.  Default = pcnt
OUT_DISPLAY_VAR: The name of the character variable created in the format dictated
by the DISPLAY_OPTION parameter.
        Default = N_PCNT

**************************************************************************
****************************************;

%PUT ---------------------------------------------- ;
%PUT INFO: (&SYSMACRONAME) ;
%PUT INFO: Version 1.0 ;
%PUT -START---------------------------------------- ;

%*-------------------------------------------------------------------------
--*;
%*** STEP 1 - PARAMATER CHECKS;
%global &sysmacroname._rc;
%let &sysmacroname._rc = 0;

%* MA_N_PCNT_RC VALUES
        0 = ran without error
        1 = in_data contains 0 observations
        2 = in_denom_data containts 0 observations
```

```
        3 = in_denom_data has multiple records for by group resulting in
many-to-many merge causing macro to abort
        4 = value of parameter DISPLAY_OPTION is not valid.  No display variable
created
        5 = numerator is not zero, but denominator is zero and BLANK_FOR_ZERO_DENOM
is set to YES
        ;

%let abort = no;


%mu_help_debug
%md_workinfo
    (debug         = &debug
    ,_workdata     = WORK_DATASETS_DATA
    ,_workview     = WORK_DATASETS_VIEW
    ,_mprinttoggle = mprint_setting
    ) *;

%md_default_check
    (mparm_name       = prcode_split
    ,md_default_mvar  = _default_prcode_split
    ,md_default_value = $
    ) *;
%md_default_check
    (mparm_name       = escapechar
    ,md_default_mvar  = _default_escapechar
    ,md_default_value = ^
    ) *;
%md_default_check
    (mparm_name       = rtf_ascii
    ,md_default_mvar  = _default_rtf_ascii
    ,md_default_value = rtf
    ) *;
%if %upcase(%substr(&out_display_var_split,1,1)) = Y %then %do;
  %if %upcase(%substr(&rtf_ascii,1,1)) = R %then %do; %let split = &escapechar.n;
%end;
  %else %if %upcase(%substr(&rtf_ascii,1,1)) = A %then %do; %let split =
&prcode_split; %end;
  %else %do;
        %put ALERT_I: Unrecognized value (&rtf_ascii) in parameter RTF_ASCII;
        %put ALERT_I: No split character will be used in the output;
  %end;
  %put split=&split;
%end;


%if &out_data = %str( ) %then %do;
        %let out_data=%upcase(_out&in_data);
%end;
```

```
%if &in_numerator eq %str( ) %then %do;
        %let in_numerator = n;
%end;
%if &display_option eq %str( ) %then %do;
        %let display_option = 1;
%end;
%if "&out_n_pcnt_format" = "" %then %do;
        %let out_n_pcnt_format = best.;
%end;


%if &trigger_pcnt_sign ne %str() %then %do;
        %let trigger_pcnt_sign = %upcase(%substr(&trigger_pcnt_sign,1,1));
        %if &trigger_pcnt_sign ne Y and &trigger_pcnt_sign ne N %then %do;
                %put ALERT_I: TRIGGER_PCNT_SIGN is &trigger_pcnt_sign.;
                %if &debug = Y and &help = Y %then %do;
                 %put ALERT_I:   TRIGGER_PCNT_SIGN must be Y to display percent
sign;
                %end;
        %end;
%end;

%if &trigger_tight_paren ne %str() %then %do;
        %let trigger_tight_paren = %upcase(%substr(&trigger_tight_paren,1,1));
        %if &trigger_tight_paren ne Y and &trigger_tight_paren ne N %then %do;
                %put ALERT_I: trigger_tight_paren is &trigger_tight_paren.;
                %if &debug = Y and &help = Y %then %do;
                 %put ALERT_I:   TRIGGER_TIGHT_PAREN must be Y to cause leading
parenthesis to float;
                 %put ALERT_I:   TRIGGER_TIGHT_PAREN must be N to fix location of
leading parenthesis ;
                %end;
        %end;
%end;

%if &trigger_n_space_pcnt ne %str() %then %do;
        %let trigger_n_space_pcnt = %upcase(%substr(&trigger_n_space_pcnt,1,1));
        %if &trigger_n_space_pcnt ne Y and &trigger_n_space_pcnt ne N %then %do;
                %put ALERT_I: TRIGGER_N_SPACE_PCNT is &trigger_n_space_pcnt.;
                %if &debug = Y and &help = Y %then %do;
                 %put ALERT_I:   TRIGGER_N_SPACE_PCNT must be Y to allow for space;

                 %put ALERT_I:   between count and percent in display variable;
                %end;
        %end;
%end;

%if &BLANK_FOR_ZERO_DENOM ne %str() %then %do;
        %let BLANK_FOR_ZERO_DENOM = %upcase(%substr(&BLANK_FOR_ZERO_DENOM,1,1));
        %put BLANK_FOR_ZERO_DENOM = &BLANK_FOR_ZERO_DENOM;
```

```
        %if &BLANK_FOR_ZERO_DENOM ne Y and &BLANK_FOR_ZERO_DENOM ne N %then %do;
                %put ALERT_I: BLANK_FOR_ZERO_DENOM is &BLANK_FOR_ZERO_DENOM.;
                %if &debug = Y and &help = Y %then %do;
                 %put ALERT_I HELP:   Set BLANK_FOR_ZERO_DENOM to Y(es) to allow
&sysmacroname ;
                 %put ALERT_I HELP:   to blank the display variable when the the
denominator is zero.;
                %end;
        %end;
%end;
%else %if &BLANK_FOR_ZERO_DENOM eq %str() %then %do;
        %let BLANK_FOR_ZERO_DENOM = N;
        %put ALERT_I: BLANK_FOR_ZERO_DENOM is missing. &sysmacroname assumes NO.;
                %if &debug = Y and &help = Y %then %do;
                 %put ALERT_I HELP:   Set BLANK_FOR_ZERO_DENOM to Y(es) to allow
&sysmacroname ;
                 %put ALERT_I HELP:   to blank the display variable when the the
denominator is zero.;
                %end;
%end;

%if &pcnt_for_zero_num ne %str() %then %do;
  %if %upcase(%substr(&pcnt_for_zero_num , 1, 1)) ne N and
%upcase(%substr(&pcnt_for_zero_num , 1, 1)) ne Y %then %do;
        %put ALERT_I: PCNT_FOR_ZERO_NUM = %quote(&pcnt_for_zero_num) is invalid.;
        %put ALERT_I:   Setting PCNT_FOR_ZERO_NUM to NO;
          %if &debug = Y and &help = Y %then %do;
        %put ALERT_I HELP:   Set PCNT_FOR_ZERO_NUM to Y(es) to display 0 (0.0);
        %put ALERT_I HELP:   Set to N(o) to display 0 only.;
          %end;
         %let pcnt_for_zero_num = N;
  %end;
%end;
%else %do;
        %put ALERT_I: PCNT_FOR_ZERO_NUM is missing.;
        %put ALERT_I:   Setting PCNT_FOR_ZERO_NUM to NO;
     %if &debug = Y and &help = Y %then %do;
         %put ALERT_I HELP:   Set PCNT_FOR_ZERO_NUM to Y(es) to display 0 (0.0);
         %put ALERT_I HELP:   Set to N(o) to display 0 only.;
     %end;
         %let pcnt_for_zero_num = N;
%end;

        %* CHECK REQUIRED PARAMAETERS AND DATA SET EXISTS;
                %mu_check_req_parameters(
         parameters_to_check = in_data in_numerator
        ,help = no
        );

        %mu_check_data_and_var_exist(
```

```
        data_to_check = &in_data
        ,vars_to_check_in_all_data =
        ,vars_to_check_in_respective_data =
        ,abort_if_does_not_exist = yes
        ,help = no
        );

                %mu_check_data_and_var_exist(
        data_to_check = &in_denom_data
        ,vars_to_check_in_all_data =
        ,vars_to_check_in_respective_data =
        ,abort_if_does_not_exist = yes
        ,help = no
        );

%*-------------------------------------------------------------------------------
--*;
*** A B O R T macro if in_data has 0 observations ***;
        %let in_data = %upcase(&in_data);
        %mu_nobs(&in_data);


%*-------------------------------------------------------------------------------
--*;
        %if &&&in_data._nobs eq 0 %then %do;
                %put
-------------------------------------------------------------------------------
-----------;
                %put ALERT_C: &SYSMACRONAME - Dataset %upcase(&in_data) contains 0
observations. &SYSMACRONAME will abort. ;
                %let &sysmacroname._rc = 1;
                %put
-------------------------------------------------------------------------------
-----------;
                Data _null_;
                 Abort;
                Run;
        %end;

%*-------------------------------------------------------------------------------
--*;
*** A B O R T macro if in_denom_data has 0 observations ***;
        %let in_denom_data = %upcase(&in_denom_data);
        %mu_nobs(&in_denom_data);


%*-------------------------------------------------------------------------------
--*;
        %if &&&in_denom_data._nobs eq 0 %then %do;
                %put
```

```
--------------------------------------------------------------------------------
-----------;
                %put ALERT_C: &SYSMACRONAME - Dataset %upcase(&in_denom_data)
contains 0 observations. &SYSMACRONAME will abort. ;
                %let &sysmacroname._rc = 2;
                %put
--------------------------------------------------------------------------------
-----------;
                Data _null_;
                 Abort;
                Run;
        %end;


%*** STEP 2 - IF NOT CREATING PERCENTS (OPTION = 0) THEN CREATE NEW DISPLAY
VARIABLE AND END;

%if &display_option = 0 %then %do;
        %* STEP 2.1 - CHECK FOR REQUIRED PARAMETERS AND DATA SETS, AND FOR OBS IN
DATA SET;
        %mu_check_req_parameters(
         parameters_to_check = in_data in_numerator
        ,help = no
        );
        %mu_check_data_and_var_exist(
        data_to_check = &in_data
        ,vars_to_check_in_all_data = &in_numerator
        ,vars_to_check_in_respective_data =
        ,abort_if_does_not_exist = yes
        ,help = no
        );

        %let in_data = %upcase(&in_data);
        %mu_nobs(&in_data);


%*--------------------------------------------------------------------------------
--*;
        %if &&&in_data._nobs eq 0 %then %do;
                %put
--------------------------------------------------------------------------------
-----------;
                %put ALERT_C: &SYSMACRONAME - Dataset %upcase(&in_data) contains 0
observations.;
                %let &sysmacroname._rc = 1;
                %put
--------------------------------------------------------------------------------
-----------;
        %end;
```

```
        %* STEP 2.2 - CREATE NEW DISPLAY VARIABLE CONSISTING OF NUMERATOR;
        data &out_data;
          set &in_data;
          &out_display_var = right(put(&in_numerator, 6.));
        run;
%end;


%*-------------------------------------------------------------------------
--*;
%*** STEP 3 - IF CREATING PERCENTS (OPTIONS > 0) THEN DO DIVISION AND CREATE NEW
DISPLAY VARIABLE ;

%else %do;
        %* STEP 3.1 - CHECK FOR REQUIRED PARAMETERS AND DATA SETS, AND FOR OBS IN
DATA SETS;
        %if &out_pcnt eq %str( ) %then %do;
                %let out_pcnt = pcnt;
        %end;

        %* if numerator or denominator dataset is invalid then warn user and abort;

        %mu_check_req_parameters(
         parameters_to_check = in_data in_numerator in_denom_data in_denom_var
in_denom_merge_vars
        ,help = no
        );

        %mu_check_data_and_var_exist(
        data_to_check = &in_data &in_denom_data
        ,vars_to_check_in_all_data = &subgroup &in_denom_merge_vars
        ,vars_to_check_in_respective_data =
        ,abort_if_does_not_exist = yes
        ,help = no
        );

        %let subgroup_and_trtvar=%trim(&subgroup. &in_denom_merge_vars.);
        %mu_wordscan(string=&subgroup_and_trtvar, root=denom, numw=numdenom);
        %do i = 1 %to &numdenom;
                %let denom&i = %upcase(&&denom&i);
        %end;

        %* if numerator or denominator dataset is empty then alert user ;
        %let in_data = %upcase(&in_data);
        %mu_nobs(&in_data);


%*-------------------------------------------------------------------------
--*;
        %if &&&in_data._nobs eq 0 %then %do;
```

```
                %put
------------------------------------------------------------------------------
-----------;
                %put ALERT_I: &SYSMACRONAME - Dataset %upcase(&in_data) contains 0
observations.;
                %let &sysmacroname._rc = 1;
                %put
------------------------------------------------------------------------------
-----------;
        %end;

        %let in_denom_data = %upcase(&in_denom_data);
        %mu_nobs(&in_denom_data);

        %if &&&in_denom_data._nobs eq 0 %then %do;
                %put
------------------------------------------------------------------------------
-----------;
                %put ALERT_I: &SYSMACRONAME - Dataset %upcase(&in_denom_data)
contains 0 observations.;
                %let &sysmacroname._rc = 2;
                %put
------------------------------------------------------------------------------
-----------;
        %end;


%*--------------------------------------------------------------------------
--*;
        %* CHECK FOR ADDITIONAL VARIABLES [XX.X];
        %if &display_option = 3 or &display_option = 4 %then %do;
                %let dsid_dsin = %sysfunc(open(&in_data.));
                %let
additional_var_type=%sysfunc(vartype(&dsid_dsin,%sysfunc(varnum(&dsid_dsin,&additio
nal_var))));
                %let close_dsid_dsin = %sysfunc(close(&dsid_dsin));

                %if &additional_var_type = N and "&additional_var_format" eq ""
%then %do;
                        %put
------------------------------------------------------------------------------
-----------;
                        %put ALERT_I: &SYSMACRONAME - Additional variable to be
displayed is numeric,;
                        %put ALERT_I: &SYSMACRONAME - but parameter
ADDITIONAL_VAR_FORMAT is blank.;
                        %put ALERT_I: &SYSMACRONAME - Program will use BEST.;
                        %let additional_var_format= best.;
                        %put
------------------------------------------------------------------------------
```

```
-----------;
                %end;

                %if "&additional_var_brackets" ne "" %then %do;
                        %if %length(%cmpres(&additional_var_brackets)) eq 2 %then
%do;
                                %let left_bracket =
%substr(%bquote(%cmpres(&additional_var_brackets)), 1, 1);
                                %let right_bracket =
%substr(%bquote(%cmpres(&additional_var_brackets)), 2, 1);
                        %end;
                        %else %if
%length(%cmpres(%bquote(&additional_var_brackets))) eq 1 %then %do;
                                %if "%cmpres(&additional_var_brackets)" = "(" or
"%cmpres(&additional_var_brackets)" = ")" %then %do;
                                        %let left_bracket = %nrstr(();
                                        %let right_bracket = %nrstr());
                                %end;
                                %else %if "%cmpres(&additional_var_brackets)" = "["
or "%cmpres(&additional_var_brackets)" = "]" %then %do;
                                        %let left_bracket = %nrstr([);
                                        %let right_bracket = %nrstr(]);
                                %end;
                                %else %if "%cmpres(&additional_var_brackets)" = "{"
or "%cmpres(&additional_var_brackets)" = "}" %then %do;
                                        %let left_bracket = %nrstr({);
                                        %let right_bracket = %nrstr(});
                                %end;
                                %else %do;
                                        %put %str(ALERT)_I: Unrecognized character
%bquote((&additional_var_brackets)) specified in parameter ADDITIONAL_VAR_BRACKETS;
                                        %put %str(ALERT)_I: Standard square
brackets "[]" will be used around additional variable;
                                        %if &help = Y %then %do;
                                          %put %str(ALERT)_I: HELP: Either specify
a pair of characters to be used as brackets;
                                          %put %str(ALERT)_I: HELP: or choose one
of the following: '(' '[' or '{' ;
                                        %end;
                                        %let left_bracket = %nrstr([);
                                        %let right_bracket = %nrstr(]);

                                %end;
                        %end;
                        %else %if
%length(%cmpres(%bquote(&additional_var_brackets))) gt 2 %then %do;
                                %put %str(ALERT)_I: More than two characters
%bquote((&additional_var_brackets)) specified in parameter ADDITIONAL_VAR_BRACKETS;
                                %put %str(ALERT)_I: Only The first two charaters
will be used;
```

```
                                %let left_bracket =
%substr(%bquote(&additional_var_brackets), 1, 1);
                                %let right_bracket =
%substr(%bquote(&additional_var_brackets), 2, 1);
                        %end;
                %end;
                %else %if "&additional_var_brackets" eq "" %then %do;
                        %let left_bracket=;
                        %let right_bracket=;
                %end;

        %end;


%*-------------------------------------------------------------------------------
--*;
        %* STEP 3.2 -  GET THE DATA AND DO THE DIVSION;

        %* check for overwriting - if a avariable exists in both numerator and
denominator data sets
                SAS will issue a note saying the second will overwrite the first;
        proc contents data=&in_data out=cont_in(keep=name) noprint;
        run;

        proc sort data=cont_in;
          by name;
        run;

        proc contents data=&in_denom_data out=cont_denom(keep=name) noprint;
        run;

        proc sort data=cont_denom;
          by name;
        run;

        data _dropvars_(drop=__alrt_msg1__ __alrt_msg2__);
          merge cont_in(in=a) cont_denom(in=b);
          by name;
          length __alrt_msg1__ __alrt_msg2__ $ 100;
          name = compress(upcase(name));
            %do i = 1 %to &numdenom;
                if name ne "&&denom&i" and flag ne 1 then flag = 0;
                else if name eq "&&denom&i" then flag = 1;
            %end;

            if a and b and flag ne 1 then do;
                __alrt_msg1__ = "ALER" || "T_I: " || compress(name) || " exists in
both &in_data. and &in_denom_data.";
                __alrt_msg2__ = "ALER" || "T_I:   and will be dropped from
&in_denom_data." ;
```

```
                put "ALER" "T_I:";
                put __alrt_msg1__;
                put __alrt_msg2__;
                put "ALER" "T_I:";
                flag = 2;
            end;
        run;

        %* get ready for merge;
        proc sort data=&in_data out=&out_data;
                by &subgroup &in_denom_merge_vars;
        run;

        proc sort data=&in_denom_data out=denom_data;
                by &subgroup &in_denom_merge_vars;
        run;

        %* check for repeats of byvars - unacceptable many-to-many merge will force
the program to abort;
        data __temp__;
                set denom_data;
                by &subgroup &in_denom_merge_vars;
                if not (first.&&denom&numdenom and last.&&denom&numdenom);
        run;

        %let _dropvars_ =;
        proc sql noprint;
                select name into :_dropvars_ separated by " " from _dropvars_ where
flag = 2;
                select * from __temp__;
                %let numrepeat = &sqlobs;
        quit;

        %if &numrepeat > 0 %then %do;
                %put
----------------------------------------------------------------------------------
-----------;
                %put ALERT_P: &SYSMACRONAME - Dataset &in_denom_data has multiple
records for by group - cannot process further.;
                %let abort = yes;
                %let &sysmacroname._rc = 3;
                %put
----------------------------------------------------------------------------------
-----------;
                %goto exit_and_abort;
        %end;

        %*
        *Calculate Percentage;
        data &out_data;
```

```
                merge &out_data(in=a ) denom_data(%if "&_dropvars_" ne "" %then
%do; drop=&_dropvars_ %end;);
                by &subgroup &in_denom_merge_vars;
                if a;
                if &in_denom_var gt 0 then do;
                        %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=N %then
%do;
                         if &in_numerator gt 0 then do;
                        %end;
                        %else %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=Y
%then %do;
                         if &in_numerator ge 0 then do;
                        %end;
                    &out_pcnt = 100 * &in_numerator/&in_denom_var;
                  end;
                end;
        run;



%*-------------------------------------------------------------------------------
--*;
        %* STEP 3.3 -  CREATE THE DISPLAY VARIABLE (n_pcnt) USING VARIOUS OPTIONS;

        %*
        *Find the length of the maximum denominator - use to align N if displayed;
        proc means noprint data=denom_data;
                var &in_denom_var;
                output out=maxdenom max=max;
        run;

        data _null_;
                set maxdenom;
                call symput("len_max_denom", trim(left(put(length(compress(put(max,
best8.))), 8.))));
        run;

        %*
        *get the width of the format associated with the percentage;
        data _null_;
                fmt = "&out_n_pcnt_format";
                if verify(fmt, "0123456789.") eq 0 then call symput('fmt_check',
'1');
                else call symput('fmt_check', '0');
        run;

        %*
        *if the format associated with the percentage is a standard SAS numeric,
then use the integer portion of the format;
        %if &fmt_check %then %do;
```

```
                data _null_;
                          x = 1+int(&out_n_pcnt_format.);
                          call symput("pxxxp_length", trim(left(put(x, 8.))));
                run;

        %end;

        %*
        *if the format associated with the percentage is not a standard, then
`label` part of the format (right side of equal sign)
        *       to determine the width;
        %else %do;

                data _null_;
                        call symput("new_n_pcnt_format",
compress("&out_n_pcnt_format", "."));
                run;
                %put new_n_pcnt_format = &new_n_pcnt_format;

                proc format cntlout=fmt_check ;
                  select &new_n_pcnt_format;
                run;
                data _null_;
                  set fmt_check end=eof;
                  retain max_fmt 3;

                  if compress(label) ne '100' then do;
                   if verify(compress(label), "0123456789.") eq 0 then
len_fmt=int(input(label, best8.));
                   else len_fmt = input(end, best8.);
                  end;
                  else if compress(label) = '100' then len_fmt = 3;

                  max_fmt = max(max_fmt, len_fmt);
                  put label= len_fmt= max_fmt=;
                  if eof then call symput('pxxxp_length', trim(left(put(1+max_fmt,
8.))));
                run;

        %end;

        %*
        **** Display Option 1: n (xx.xx[%])
        **;
        %if &display_option = 1 %then %do;
                data &out_data(drop=pxxxp_:);
                        set &out_data;
                        length &out_display_var $ 50 pxxxp_tight pxxxp_loose $
&pxxxp_length ;
```

```
                        if &in_denom_var gt 0 then do;
                         %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=N %then
%do;
                          if &in_numerator gt 0 then do;
                         %end;
                         %else %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=Y
%then %do;
                          if &in_numerator ge 0 then do;
                         %end;
                            if .Z lt &out_pcnt then do;
                              pxxxp_tight = right("(" || compress(put(&out_pcnt,
&out_n_pcnt_format.)));
                              pxxxp_loose = "(" || right(put(&out_pcnt,
&out_n_pcnt_format.)) ;
                                %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) = N %then %do;
                                  if &out_pcnt = 100 then do;
                                        pxxxp_tight = "(100";
                                        pxxxp_loose = "(100";
                                  end;
                                %end;
                            end;

                        %if %upcase(%substr(&trigger_100_with_decimal, 1,1)) ne N
%then %do;
                            &out_display_var =
                                right(put(&in_numerator, 6.))
                                  %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_loose)
                                  %end;
                                  %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_tight)
                                  %end;
                                  %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                  %end;
                                  %else %do;
                                  || ")";
                                  %end;
                        %end;
                        %else %if %upcase(%substr(&trigger_100_with_decimal, 1,1))
eq N %then %do;
                                  if &out_pcnt ne 100 then
```

```
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_loose)
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_tight)
                                    %end;
                                    %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                    %end;
                                    %else %do;
                                  || ")";
                                    %end;

                                  else if &out_pcnt eq 100 then
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     (pxxxp_loose)
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     (pxxxp_tight)
                                    %end;
                                    %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                    %end;
                                    %else %do;
                                  || ")";
                                    %end;
                          %end;


                            end;
                            else if &in_numerator=0 then do;
                                  &out_display_var = right(put(&in_numerator, 6.));
```

```
                          end;
                        end;
                        %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = N %then
%do;
                        else if not missing(&in_numerator) and &in_denom_var=0 then
do;
                          &out_display_var = right(put(&in_numerator, 6.));
                        end;
                        %end;
                        %else %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = Y
%then %do;
                        if &in_denom_var=0 then do;
                          &out_display_var = '';
                          if &in_numerator ne 0 then do;
                                put "ALERT_C: " &IN_NUMERATOR= &IN_DENOM_VAR= ;
                                put "ALERT_C:   BLANK_FOR_ZERO_DENOM set to YES.
Display variable will be set to blank";
                                call symputx("&sysmacroname._rc" ,5);
                          end;

                        end;
                        %end;
                        %if "&no_pcnt_if" ne "" %then %do;
                                if &no_pcnt_if and &out_display_var ne '' then
&out_display_var = right(put(&in_numerator, 6.));
                        %end;
                run;
        %end;

        %*
        **** Display Option 2: n/N (xx.xx[%])
        **;
        %else %if &display_option = 2 %then %do;
                data &out_data(drop=pxxxp_:);
                        set &out_data;
                        length &out_display_var $ 50 pxxxp_tight pxxxp_loose $
&pxxxp_length ;
                        if &in_denom_var gt 0 then do;
                         %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=N %then
%do;
                          if &in_numerator gt 0 then do;
                         %end;
                         %else %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=Y
%then %do;
                          if &in_numerator ge 0 then do;
                         %end;
                            if .Z lt &out_pcnt then do;
                              pxxxp_tight = right("(" || compress(put(&out_pcnt,
&out_n_pcnt_format.)));
                                pxxxp_loose = "(" || right(put(&out_pcnt,
```

```
&out_n_pcnt_format.)) ;
                                %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) = N %then %do;
                                  if &out_pcnt = 100 then do;
                                        pxxxp_tight = "(100";
                                        pxxxp_loose = "(100";
                                  end;
                                %end;
                              end;
                            %if %upcase(%substr(&trigger_100_with_decimal, 1,1)) ne N
%then %do;
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                  || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_loose)
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_tight)
                                    %end;
                                    %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                    %end;
                                    %else %do;
                                  || ")";
                                    %end;
                            %end;
                            %else %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) eq N %then %do;
                              if &out_pcnt ne 100 then
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                  || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_loose)
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
```

```
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     right(pxxxp_tight)
                                    %end;
                                    %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                    %end;
                                    %else %do;
                                  || ")";
                                    %end;
                               else if &out_pcnt eq 100 then
                               &out_display_var =
                                   right(put(&in_numerator, 6.))
                                   || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     (pxxxp_loose)
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                     (pxxxp_tight)
                                    %end;
                                    %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                    %end;
                                    %else %do;
                                  || ")";
                                    %end;

                            %end;

                            end;
                            else if &in_numerator=0 then do;
                                  &out_display_var = right(put(&in_numerator, 6.))
                                  || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..));
                            end;
                          end;
                          %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = N %then
%do;
                          else if not missing(&in_numerator) and &in_denom_var=0 then
do;
                            &out_display_var = right(put(&in_numerator, 6.))
```

```
                          || %if "&fraction_bar" ne "" %then %do; "&fraction_bar"
|| %end; right(put(&in_denom_var,  &len_max_denom..)) ;
                        end;
                        %end;
                        %else %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = Y
%then %do;
                        if &in_denom_var=0 then do;
                          &out_display_var = '';
                        if &in_numerator ne 0 then do;
                          put "ALERT_C: " &IN_NUMERATOR= &IN_DENOM_VAR= ;
                          put "ALERT_C:   BLANK_FOR_ZERO_DENOM set to YES. Display
variable will be set to blank";
                                call symputx("&sysmacroname._rc" ,5);
                        end;

                        end;
                        %end;
                        %if "&no_pcnt_if" ne "" %then %do;
                                if &no_pcnt_if and &out_display_var ne '' then
&out_display_var = right(put(&in_numerator, 6.))
                                        || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..)) ;
                        %end;
                run;
        %end;

        %*
        **** Display Option 3: n (xx.xx[%]) [xx]
        **;
        %else %if &display_option = 3 %then %do;
                data &out_data(drop=pxxxp_:);
                        set &out_data;
                        length &out_display_var $ 50;
                        %if "&no_pcnt_if" ne "" %then %do;
                                if &no_pcnt_if then &out_display_var =
right(put(&in_numerator, 6.))
                                        %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length+1)) ||
'ab'x;
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', &pxxxp_length) || 'ab'x;
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length-1)) ||
```

```
'ab'x;
                                        %end;
                                  else do;
                           %end;

                           length pxxxp_tight pxxxp_loose $ &pxxxp_length ;
                           if &in_denom_var gt 0 then do;
                            %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=N %then
%do;
                             if &in_numerator gt 0 then do;
                            %end;
                            %else %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=Y
%then %do;
                             if &in_numerator ge 0 then do;
                            %end;
                               if .Z lt &out_pcnt  then do;
                                 pxxxp_tight = right("(" || compress(put(&out_pcnt,
&out_n_pcnt_format.)));
                                 pxxxp_loose = "(" || right(put(&out_pcnt,
&out_n_pcnt_format.)) ;
                                   %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) = N %then %do;
                                     if &out_pcnt = 100 then do;
                                           pxxxp_tight = "(100";
                                           pxxxp_loose = "(100";
                                     end;
                                   %end;
                               end;

                             %if %upcase(%substr(&trigger_100_with_decimal, 1,1)) ne N
%then %do;
                               &out_display_var =
                                   right(put(&in_numerator, 6.))
                                     %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                   || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                      right(pxxxp_loose)
                                     %end;
                                     %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                   || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                      right(pxxxp_tight)
                                     %end;
                                     %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                   || "%)";
                                     %end;
                                     %else %do;
```

```
                                || ")";
                                  %end;
                            %end;
                            %else %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) eq N %then %do;
                              if &out_pcnt ne 100 then
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                   right(pxxxp_loose)
                                  %end;
                                  %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                   right(pxxxp_tight)
                                  %end;
                                  %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                || "%)";
                                  %end;
                                  %else %do;
                                || ")";
                                  %end;
                              else if &out_pcnt eq 100 then
                              &out_display_var =
                                  right(put(&in_numerator, 6.))
                                    %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                   (pxxxp_loose)
                                  %end;
                                  %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                   (pxxxp_tight)
                                  %end;
                                  %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                || "%)";
                                  %end;
                                  %else %do;
                                || ")";
                                  %end;
                            %end;
```

```
                        end;
                        else if &in_numerator=0 then do;
                              &out_display_var = right(put(&in_numerator, 6.))
                                    %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                    || repeat(' ', %eval(&pxxxp_length+1)) ||
'ab'x;
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                    || repeat(' ', &pxxxp_length) || 'ab'x;
                                    %end;
                                    %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                    || repeat(' ', %eval(&pxxxp_length-1)) ||
'ab'x;
                                    %end;
                        end;
                      end;
                      %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = N %then
%do;
                      else if not missing(&in_numerator) and &in_denom_var=0 then
do;
                        &out_display_var = right(put(&in_numerator, 6.));
                      end;
                      %end;
                      %else %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = Y
%then %do;
                      if &in_denom_var=0 then do;
                        &out_display_var = '';
                        if &in_numerator ne 0 then do;
                              put "ALERT_C: " &IN_NUMERATOR= &IN_DENOM_VAR= ;
                              put "ALERT_C:   BLANK_FOR_ZERO_DENOM set to YES.
Display variable will be set to blank";
                              call symputx("&sysmacroname._rc" ,5);
                        end;

                      end;
                      %end;
                      %if "&no_pcnt_if" ne "" %then %do;
                              if &no_pcnt_if and &out_display_var ne '' then
&out_display_var = right(put(&in_numerator, 6.));
                      end;
                      %end;

                      %if "&no_additional_var_if" eq ""  %then %do;
                              %if &additional_var_type = N %then %do;
```

```
                                        if &out_display_var ne '' then
&out_display_var = trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(put(&additional_var,&additional_var_format)) || "&right_bracket ";
                                %end;
                                %else %do;
                                        if &out_display_var ne '' then
&out_display_var = trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(&additional_var) || "&right_bracket ";
                                %end;
                        %end;

                        %else %do;
                                %if &additional_var_type = N %then %do;
                                        if &no_additional_var_if then
&out_display_var = &out_display_var;
                                        else if &out_display_var ne '' then
                                                &out_display_var =
trim(&out_display_var)
                                                %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                        || "&split"
                                                %end;
                                                ||
"&additional_var_delim.&left_bracket" ||
strip(put(&additional_var,&additional_var_format)) || "&right_bracket ";
                                %end;
                                %else %do;
                                        if &no_additional_var_if then
&out_display_var = &out_display_var;
                                        else if &out_display_var ne '' then
                                                &out_display_var =
trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(&additional_var) || "&right_bracket ";
                                %end;
                        %end;
```

```
                        &out_display_var = translate(&out_display_var, ' ', 'ab'x
);
                run;
        %end;

        %*
        **** Display Option 4: n/N (xx.xx[%]) [xx]
        **;
        %else %if &display_option = 4 %then %do;
                data &out_data(drop=pxxxp_:);
                        set &out_data;
                        length &out_display_var $ 50 pxxxp_tight pxxxp_loose $
&pxxxp_length ;
                        if &in_denom_var gt 0 then do;
                         %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=N %then
%do;
                          if &in_numerator gt 0 then do;
                         %end;
                         %else %if %upcase(%substr(&pcnt_for_zero_num , 1, 1))=Y
%then %do;
                          if &in_numerator ge 0 then do;
                         %end;
                            if .Z lt &out_pcnt  then do;
                              pxxxp_tight = right("(" || compress(put(&out_pcnt,
&out_n_pcnt_format.)));
                              pxxxp_loose = "(" || right(put(&out_pcnt,
&out_n_pcnt_format.)) ;
                                %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) = N %then %do;
                                  if &out_pcnt = 100 then do;
                                        pxxxp_tight = "(100";
                                        pxxxp_loose = "(100";
                                  end;
                                %end;
                            end;
                            else do;
                                        %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                  pxxxp_tight = repeat(' ', %eval(&pxxxp_length+1))
|| 'ab'x;
                                  pxxxp_loose = repeat(' ', %eval(&pxxxp_length+1))
|| 'ab'x;
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                  pxxxp_tight = repeat(' ', &pxxxp_length) || 'ab'x;
                                  pxxxp_loose = repeat(' ', &pxxxp_length) || 'ab'x;
                                        %end;
```

```
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                   pxxxp_tight = repeat(' ', %eval(&pxxxp_length-1))
|| 'ab'x;
                                   pxxxp_loose = repeat(' ', %eval(&pxxxp_length-1))
|| 'ab'x;
                                        %end;

                               end;

                              %if %upcase(%substr(&trigger_100_with_decimal, 1,1)) ne N
%then %do;
                               &out_display_var =
                                   right(put(&in_numerator, 6.))
                                   || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                     %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                   || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                      right(pxxxp_loose)
                                     %end;
                                     %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                   || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                      right(pxxxp_tight)
                                     %end;
                                     %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                   || "%)";
                                     %end;
                                     %else %do;
                                   || ")";
                                     %end;
                              %end;
                              %else %if %upcase(%substr(&trigger_100_with_decimal,
1,1)) eq N %then %do;
                               if &out_pcnt ne 100 then
                               &out_display_var =
                                   right(put(&in_numerator, 6.))
                                   || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                     %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                   || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                      right(pxxxp_loose)
                                     %end;
                                     %else %if
```

```
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                    right(pxxxp_tight)
                                   %end;
                                   %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                   %end;
                                   %else %do;
                                  || ")";
                                   %end;
                             else if &out_pcnt eq 100 then
                             &out_display_var =
                                 right(put(&in_numerator, 6.))
                                 || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                   %if %upcase(%substr(&trigger_tight_paren,1,1)) ne
Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                    (pxxxp_loose)
                                   %end;
                                   %else %if
%upcase(%substr(&trigger_tight_paren,1,1)) eq Y %then %do;
                                  || %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y %then %do; " " ||%end;
                                    (pxxxp_tight)
                                   %end;
                                   %if %upcase(%substr(&trigger_pcnt_sign, 1,1)) = Y
%then %do;
                                  || "%)";
                                   %end;
                                   %else %do;
                                  || ")";
                                   %end;
                          %end;


                          end;
                          else if &in_numerator=0 then do;
                                &out_display_var = right(put(&in_numerator, 6.))
                                || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                       %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                       || repeat(' ', %eval(&pxxxp_length+1)) ||
'ab'x
                                       %end;
```

```
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', &pxxxp_length) || 'ab'x
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length-1)) ||
'ab'x
                                        %end;
                                        ;

                          end;
                        end;
                        %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = N %then
%do;
                        else if not missing(&in_numerator) and &in_denom_var=0 then
do;
                          &out_display_var = right(put(&in_numerator, 6.))
                          || %if "&fraction_bar" ne "" %then %do; "&fraction_bar"
|| %end; right(put(&in_denom_var,  &len_max_denom..))
                                        %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length+1)) ||
'ab'x
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', &pxxxp_length) || 'ab'x
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length-1)) ||
'ab'x
                                        %end;
                                        ;
                        end;
                        %end;
                        %else %if %upcase(%substr(&blank_for_zero_denom, 1, 1)) = Y
%then %do;
                        if &in_denom_var=0 then do;
                          &out_display_var = '';
                          if &in_numerator ne 0 then do;
                                put "ALERT_C: " &IN_NUMERATOR= &IN_DENOM_VAR= ;
                                put "ALERT_C:   BLANK_FOR_ZERO_DENOM set to YES.
Display variable will be set to blank";
                                call symputx("&sysmacroname._rc" ,5);
                          end;
```

```
                        end;
                        %end;
                        %if "&no_pcnt_if" ne "" %then %do;
                                if &no_pcnt_if and &out_display_var ne '' then
&out_display_var = right(put(&in_numerator, 6.))
                                        || %if "&fraction_bar" ne "" %then %do;
"&fraction_bar" || %end; right(put(&in_denom_var,  &len_max_denom..))
                                        %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y and
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length+1)) ||
'ab'x;
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1))=Y or
%upcase(%substr(&trigger_pcnt_sign,1,1)) eq Y %then %do;
                                        || repeat(' ', &pxxxp_length) || 'ab'x;
                                        %end;
                                        %else %if
%upcase(%substr(&trigger_n_space_pcnt,1,1)) ne Y %then %do;
                                        || repeat(' ', %eval(&pxxxp_length-1)) ||
'ab'x;
                                        %end;
                        %end;

                        %if "&no_additional_var_if" eq ""  %then %do;
                                %if &additional_var_type = N %then %do;
                                        if &out_display_var ne '' then
&out_display_var = trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(put(&additional_var,&additional_var_format)) || "&right_bracket ";
                                %end;
                                %else %do;
                                        if &out_display_var ne '' then
&out_display_var = trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(&additional_var) || "&right_bracket ";
                                        %end;
                        %end;

                        %else %do;
```

```
                                %if &additional_var_type = N %then %do;
                                        if &no_additional_var_if then
&out_display_var = &out_display_var;
                                        else if &out_display_var ne '' then
                                                &out_display_var =
trim(&out_display_var)
                                                %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                        || "&split"
                                                %end;
                                                ||
"&additional_var_delim.&left_bracket" ||
strip(put(&additional_var,&additional_var_format)) || "&right_bracket ";
                                %end;
                                %else %do;
                                        if &no_additional_var_if then
&out_display_var = &out_display_var;
                                        else if &out_display_var ne '' then
                                                &out_display_var =
trim(&out_display_var)
                                        %if
%upcase(%substr(&out_display_var_split,1,1))=Y %then %do;
                                                || "&split"
                                        %end;
                                        || "&additional_var_delim.&left_bracket" ||
strip(&additional_var) || "&right_bracket ";
                                %end;
                        %end;
                        &out_display_var = translate(&out_display_var, ' ', 'ab'x);
                run;
        %end;

        %else %do;
                %put
----------------------------------------------------------------------------------
-----------;
                %put ALERT_I: Value of parameter DISPLAY_OPTION is
&display_option.;
                %put ALERT_I: Value must be an integer from 0 to 4.;
                %put ALERT_I: &SYSMACRONAME will exit;
                %let &sysmacroname._rc = 4;
                %put
----------------------------------------------------------------------------------
-----------;
                %goto exit_and_abort;
        %end;

%end;

/*************************************************************************/
```

```
%*** STEP 4 - END OF PROCESS TASKS;


%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 INFO: MA_N_PCNT_RC = &MA_N_PCNT_RC;
%PUT -END------------------------------------------- ;

%exit_and_abort:

%if &abort = yes %then %do;
        data _null_;
                abort;
        run;
%end;

%mend ma_n_pcnt;
```