```
%macro ma_summ_stats
    ( in_data                  =
    , in_where                 =
    , trtvar                   =
    , DEFINE_TOTAL_GROUPS      =
    , TRTVAR_PRELOADFMT        =
    , subgroup                 =
    , bign_in_data             =
    , byvars                   =
    , byvars_preloadfmt        =
    , byvars_sort_order        =
    , byvars_sort_asc_or_desc  =
    , vars                     =
    , summ_stats_additional    =
    , usubjid                  =
    , _section_                = 1
    , out_data                 = SUMM_STATS
    , debug                    =
    , help                     =
    ) / store source des='V1.0.0.12';

%*************************************************************************
FILENAME:    MA_SUMM_STATS.SAS
DEVELOPER:   (b) (6)
PLATFORM:    SAS 9.1.3, 9.2 on PC
MACROS USED: mu_wordscan.sas
ASSUMPTIONS: Macro MA_BIGN must be called before calling this macro
DESCRIPTION:

This macro:

  1. outputs a dataset as specified in parameter &OUT_DATA (default is SUMM_STATS)
     containing summary statistics based on VARS and SUMM_STATS_ADDITIONAL
     parameters grouped by BY_VARS, SUBGROUPS, and TREATMENTS groups.
     Examples of when macro would be called:

        a) For Vital Signs table, summarize all subjects statistics at each vist
interval, per
           treatment group and subgroup, using numerical variable AVAL.

  2. Final dataset will have following variables:

        &SUBGROUP: will exist only if parameter SUBGROUP is populated.

        SUMMARY STATS VARS: Will be based off of the values passed to the input
parameters
            VARS and SUMM_STATS_ADDITIONAL.  See examples below for how stat vars
            are created in output dataset.

            EXAMPLE 1:
```

```
    When input parameters are set to:

        VARS = AVAL
        SUMM_STATS_ADDITIONAL =

    The following output variables will be created:

        VAR1_N
        VAR1_MEAN
        VAR1_SD
        VAR1_MED
        VAR1_MIN
        VAR1_MAX

EXAMPLE 2:

    When input parameters are set to:

        VARS = AVAL CHG
        SUMM_STATS_ADDITIONAL = P5 P95

    The following output variables will be created:

        VAR1_N
        N_VAR2_N
        VAR1_MEAN
        VAR2_MEAN
        VAR1_SD
        VAR2_SD
        VAR1_MED
        VAR2_MED
        VAR1_MIN
        VAR2_MIN
        VAR1_MAX
        VAR2_MAX
        VAR1_P5
        VAR2_P5
        VAR1_P95
        VAR2_P95

EXAMPLE NOTES:

    The following stats will be converted as follows for standard
    naming conventions on the output variables.

        STDDEV, STD, & SD       ==> STDDEV
        MEDIAN, P50, & MED      ==> MEDIAM
        MINIMUM & MIN           ==> MIN
        MAXIMUM & MAX           ==> MAX
```

```
                    SKEWNESS & SKEW         ==> SKEWNESS
                    KURTOSIS & KURT         ==> KURTOSIS
                    Q1 & P25                ==> P25
                    Q3 & P75                ==> P75
                    PROBT & PRT             ==> PROBT
```

BYVAR1_ORDER: order number corresponding to the first variable in BYVAR

BYVAR2_ORDER: order number corresponding to the second variable in BYVAR.

BYVAR3_ORDER: order number corresponding to the third variable in BYVAR

USAGE NOTES:

IN_DATA = input data used to calculate the summary statistics. In the form LIB.DATA, where lib may be ommitted if WORK
(default = blank) REQUIRED)
IN_WHERE = subsetting clause
(default = blank) (OPTIONAL)

TRTVAR = the treatment group variable. Used in the CLASS statement of the PROC MEANS.
(default = &_default_trtvar) (REQUIRED)

DEFINE_TOTAL_GROUPS = used to define (sub)total group(s), this parameter must be defined as follows:
list of treatment variable values = subtotal 1 ! list of treatment variable values = subtotal 2 ! ...
(default = &_default_define_total_groups) (OPTIONAL)

For example, the values of TRTVAR are 'A' 'B' and 'C' and the analysis calls for a subtotal of 'A' and
'B' and an overall total of all three treatments. Set the parameter as
define_total_groups = 'A' 'B' = 'subtotal' ! 'A' 'B' 'C' = 'total'
The (sub)total value(s) must be of the same type as the exisiting variable
The (sub)totals must be separated by an exclamation point (!)
The (sub)total values are then used in the TRTVAR_PRELOADFMT (see below)

TRTVAR_PRELOADFMT = the format which assigns the order of the treatments across the page. If no
format is listed, SAS will generate a warn-ing that PRELAODFMT will have no effect,
ie, treatment groups which are not already in the data will not be created/dummied.

```
    Also, note that any (sub)total groups should be defined here.
    (default = &_default_trtvar_preloadfmt) (OPTIONAL - *but highly recommended)

    For example, as above, the values of TRTVAR are 'A' 'B' and 'C' and the
analysis calls
    for the columns to be displayed as
        'A', 'B', 'Subtotal of A&B', 'C', 'Total'
    set the format as
      value $_trt <<<--fmtname of user choice
        'A' = '1'
        'B' = '2'
        'subtotal' = '3'
        'C' = '4'
        'total' = '5'

        NOTE - 'subtotal' and 'total' were created with DEFINE_TOTAL_GROUPS.  User
can pick
        whatever makes sense.

SUBGROUP = subgroups variable(s) to use in the BY statement of the PROC MEANS.
    (default = blank) (OPTIONAL)

USUBJID = variable to identify unique subject in IN_DATA.
    (default = USUBJID) (REQUIRED)

OUT_DATA = the output data set created.
    (default = SUMM_STATS) (REQUIRED)

HELP = set to Y(es) to output helpful hints to the log
    (Default = &_default_help)

DEBUG = set to Y(es) to save all intermediate datasets and to turn on mprint. Set
to NO to delete
    all intermediate datasets and to not turn on mprint (maintains original option
setting)
    (Default = &_default_debug)

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


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


%**** STEP 1 - PARAMETER CHECKS ;
%global &sysmacroname._RC ;
%let &sysmacroname._RC = 0 ;
```

```
%let abort = no ;

/****
R1: IN_DATA not blank(MU_CHECK_REQ_PARAMETERS_RC=1)? and
existed(MU_CHECK_DATA_AND_VAR_EXIST_RC=3)
R2: VARS not blank(MU_CHECK_REQ_PARAMETERS_RC=1)
R3: IN_WHERE can be used
R4: variables specified VARS should exist(s)
R5: BYVARS can be used
R6: TRTVAR not blank(MU_CHECK_REQ_PARAMETERS_RC=1)
R7: TRTVAR_PRELOADFMT not blank (MU_CHECK_REQ_PARAMETERS_RC=1)
R8: exsitance of format specified in TRTVAR_PRELOADFMT (MA_SUMM_STATS_RC=1)
R9: DEFINE_TOTAL_GROUPS is optional
R10: SUBGROUP optional. If specified, should be in IN_DATA
(MU_CHECK_DATA_AND_VAR_EXIST_RC=4)

R16: Sort OUT_DATA by TRTVAR, BYVARS (if provided) and SUBGROUP (if provided).
***/

%mu_help_debug ;

%* get current setting of mprint and if set to DEBUG mode turn mprint on;
%md_workinfo(debug = &debug );

*check for defaults;
%md_default_check
    (mparm_name       = trtvar
    ,md_default_mvar  = _default_trtvar
    ,md_default_value =
    ) *;
%md_default_check
    (mparm_name       = define_total_groups
    ,md_default_mvar  = _default_define_total_groups
    ,md_default_value =
    ) *;
%md_default_check
    (mparm_name       = trtvar_preloadfmt
    ,md_default_mvar  = _default_trtvar_preloadfmt
    ,md_default_value =
    ) *;
%md_default_check
    (mparm_name       = usubjid
    ,md_default_mvar  = _default_usubjid
    ,md_default_value =
    ) *;

%*R1 R2
%* check if required parameters TRTVAR, vars, and IN_DATA are blank ;
%mu_check_req_parameters
    ( parameters_to_check   = trtvar vars in_data TRTVAR_PRELOADFMT usubjid
```

```
    , help                  = no
    ) ;


%*R1 R4;
%* check for invalid IN_DATA or invalid variables in IN_DATA;
%mu_check_data_and_var_exist
    ( data_to_check                     = &in_data
    , vars_to_check_in_all_data         =
    , vars_to_check_in_respective_data  = &subgroup &byvars &trtvar &vars &usubjid
    , abort_if_does_not_exist           = yes
    , help                              = no
    ) ;


%* check for missing TRTVAR_PRELOADFMT ;
%if &trtvar_preloadfmt eq %str( ) %then %do ;
    %put
---------------------------------------------------------------------------------
---------- ;
    %put %str(ALERT_)I: &SYSMACRONAME - No format has been specified in the
TRTVAR_PRELOADFMT parameter. ;
    %put %str(ALERT_)I: &SYSMACRONAME - If %upcase(&TRTVAR) is not already
formatted PRELOADFMT will have no effect. ;
    %put
---------------------------------------------------------------------------------
---------- ;
%end ;
%else %do ;
    data _null_ ;
        call symput("trtvar_preloadfmt", compress("&trtvar_preloadfmt", '.')) ;
    run ;
%end ;

%* check PARAMETERS related to byvars ;


%md_byvar_check(in_data=&in_data
               ,byvars=&byvars
               ,byvars_preloadfmt=&byvars_preloadfmt
               ,byvars_sort_order=&byvars_sort_order
               ,byvars_sort_asc_or_desc=&byvars_sort_asc_or_desc);



%if "&out_data" eq "" %then %do ;
    %let out_data = SUMM_STATS  ;
%end ;
```

```
%if "&subgroup" ne "" %then %do ;
    %mu_wordscan
        ( string    = &subgroup
        , root      = subgroupvar
        , numw      = num_subgroup
        ) ;
%end ;

%if "&vars" ne "" %then %do ;
    %mu_wordscan
        ( string    = &vars
        , root      = summ_stats_var
        , numw      = num_summ_stats_vars
        ) ;
%end ;


%*remove base stats from summary_stats_addl and use standard terms (found in
MA_SUMM_STAT_RULES) for all additioanl stats ;
%if "&summ_stats_additional" ne "" %then %do ;

    %mu_wordscan
        ( string    = &summ_stats_additional
        , root      = summ_stats_addl
        , numw      = num_summ_stats_addl
        ) ;

    %do l = 1 %to &num_summ_stats_addl ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("n") |
            %upcase("&&summ_stats_addl&l") eq %upcase("mean") |
            %upcase("&&summ_stats_addl&l") eq %upcase("stddev") |
            %upcase("&&summ_stats_addl&l") eq %upcase("std") |
            %upcase("&&summ_stats_addl&l") eq %upcase("sd") |
            %upcase("&&summ_stats_addl&l") eq %upcase("median") |
            %upcase("&&summ_stats_addl&l") eq %upcase("med") |
            %upcase("&&summ_stats_addl&l") eq %upcase("p50") |
            %upcase("&&summ_stats_addl&l") eq %upcase("minimum") |
            %upcase("&&summ_stats_addl&l") eq %upcase("min") |
            %upcase("&&summ_stats_addl&l") eq %upcase("maximum") |
            %upcase("&&summ_stats_addl&l") eq %upcase("max") %then %do ;
            %let summ_stats_addl&l = ;
        %end ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("skew") %then %do ;
            %let summ_stats_addl&l = skewness ;
        %end ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("kurt") %then %do ;
            %let summ_stats_addl&l = kurtosis ;
        %end ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("p25") %then %do ;
            %let summ_stats_addl&l = q1 ;
```

```
        %end ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("p75") %then %do ;
            %let summ_stats_addl&l = q3 ;
        %end ;
        %if %upcase("&&summ_stats_addl&l") eq %upcase("prt") %then %do ;
            %let summ_stats_addl&l = probt ;
        %end;
    %end ;

    data _null_ ;
        call symput("summ_stats_additional",
            %do l = 1 %to &num_summ_stats_addl ;
                "&&summ_stats_addl&l" || " " ||
            %end ;
                " ") ;
        %do l = 1 %to &num_summ_stats_addl ;
            call symput("summ_stats_addl&l"," ") ;
        %end ;
    run ;


    %mu_wordscan
        ( string    = &summ_stats_additional
        , root      = summ_stats_addl
        , numw      = num_summ_stats_addl
        ) ;

    %*remove dups from summ_stats_addl (additioanl stats) ;
    %if &num_summ_stats_addl ge 1 %then %do ;

        %do l = &num_summ_stats_addl %to 1 %by -1 ;
            %do m = 1 %to &num_summ_stats_addl ;
                %if %upcase("&&summ_stats_addl&l") eq
%upcase("&&summ_stats_addl&m") & &l ne &m %then %do ;
                    %let summ_stats_addl&l = ;
                %end;
            %end ;
        %end ;

        data _null_ ;
            call symput("summ_stats_addl",
                %do l = 1 %to &num_summ_stats_addl ;
                    "&&summ_stats_addl&l" || " " ||
                %end ;
                    " ") ;
            %do l = 1 %to &num_summ_stats_addl ;
                call symput("summ_stats_addl&l"," ") ;
            %end ;
        run ;
```

```
        %if "&summ_stats_additional" ne "" %then %do ;
            %mu_wordscan
                ( string    = &summ_stats_additional
                , root      = summ_stats_addl
                , numw      = num_summ_stats_addl
                ) ;
        %end ;

    %end ;

%end ;
%else %do;
  %let summ_stats_addl =;
%end;

%*create summary_stats global macro variabls based on base summary stats and user
additional summary stats;
%let summ_stats = n mean stddev median min max &summ_stats_addl ;

%if "&summ_stats" ne "" %then %do ;
    %mu_wordscan
        ( string    = &summ_stats
        , root      = summ_stats
        , numw      = num_summ_stats
        ) ;
%end ;

%if &&&sysmacroname._RC = 0 %then %do ;
    %put ------------------------------------ ;
    %put %str(ALERT_)I: &SYSMACRONAME - Using dataset &IN_DATA for Summary
Statistics ;
    %put ------------------------------------ ;
%end ;

   %if "&trtvar_preloadfmt" ne "" %then %do ;

        %* determine one possible value of the trtvar variable for situations with
0 records. ;
        proc format lib=work
            cntlout=trtvar_format
(where=(fmtname=upcase(compress("&trtvar_preloadfmt",'$')))) ;
        run ;

        data trtvar_format ;
            set trtvar_format ;
            if _n_ = 1 ;
            call symputx('trtvar_start',start) ;
        run ;

        %mu_nobs(trtvar_format);
```

```
        %MU_CHECKRC
        (CONDITION = %str(&trtvar_format_nobs = 0)
        ,RC       = 1
        ,rcprefix = ma_summ_stats
        ,MSG1     = &SYSMACRONAME: Parameter TRTVAR_PRELOADFMT is specified as
&trtvar_preloadfmt
        ,MSG2     = &SYSMACRONAME: however the format &trtvar_preloadfmt cannot be
found/loaded
        ,MSG3     = &SYSMACRONAME - Program will ABORT
        ,ALERTID  = P
        ,SETABORT = Y
        );

    %end ;

%*** STEP 2 - GET THE DATA,CREATE THE DUMMY COUNTING VARIABLE, APPLY FORMATS, CHECK
FOR MISSING VALUES *** ;
    data dsin ;
        set &in_data ;
            %if %sysevalf(%superq(in_where)=,boolean)=0 %then %do; where &in_where;
%end;
        format
            %if "&trtvar_preloadfmt" ne "" %then %do ;
                &trtvar &trtvar_preloadfmt..
            %end ;
            %if "&byvars" ne "" and "&byvars_preloadfmt" ne "" %then %do ;
                %do i = 1 %to &num_byvars ;
                    %if "&&fmt&i" ne "BLANK" %then %do ;
                        &&byvar&i &&fmt&i...
                    %end ;
                %end ;
            %end ;
        ;
    run;




    %if "&byvars" ne "" and "&byvars_preloadfmt" ne "" %then %do ;
            %mu_var_attributes
                ( datasets  = &in_data
                , variables = &byvars
                , debug=&debug
                ) ;

            data _null_ ;
                call symput("muva_in_data", tranwrd("&in_data", ".", "_")) ;
            run ;
```

```
      %end ;

    data dsin ;
        set dsin ;
        %if "&subgroup" ne "" %then %do ;
            %do i = 1 %to &num_subgroup ;
                if missing(&&subgroupvar&i)
                    then put "ALERT_I: SUBGROUP variable %upcase(&&subgroupvar&i)
is missing at record " _n_ ;
            %end ;
        %end ;
        if missing(&trtvar)
            then put "ALERT_I: TRTVAR variable %upcase(&trtvar) is missing at
record " _n_  "Record will be removed." ;
        if missing(&trtvar)
            then delete ;
        %if "&byvars_preloadfmt" ne "" %then %do ;
            %do j = 1 %to &num_byvars ;
                %if "&&fmt&j" ne "BLANK" %then %do ;
                    length new_&&byvar&j $200;
                    new_&&byvar&j = put(&&byvar&j,&&fmt&j...) ;
                %end ;
                %else %do ;
                    length new_&&byvar&j $200;
                    %if &&&&&muva_in_data._&&byvar&j.._vartyp=C %then %do;
                       new_&&byvar&j = &&byvar&j ;    ***if numeric then ....???*;
                    %end;
                    %else %if &&&&&muva_in_data._&&byvar&j.._vartyp=N %then %do;
                       if not missing(&&byvar&j) then  new_&&byvar&j =
strip(put(&&byvar&j,best.)) ;
                    %end;
                %end ;
            %end ;
        %end;
    run ;



    %mu_nobs (dsin) ;

    %* if input dataset has no observations after subsetting then create dummy
dataset
       with count zero ;
    %if &dsin_nobs eq 0 %then %do ;

        %put %str(ALERT_)I: Dataset &in_data contains 0 observations after
subsetting. ;

        %* get format associated with subgroup variable ;
        %if "&subgroup" ne "" %then %do;
```

```
            %mu_var_attributes
                ( datasets  = &in_data
                , variables = &subgroup
                ) ;

            data _null_ ;
                call symput("muva_in_data", tranwrd("&in_data", ".", "_")) ;
            run ;

        %end ;

        *Bring in input dataset without subsetting and later replace all counts
with 0. ;
        proc sort data=&in_data (keep = &subgroup &usubjid &byvars &trtvar &vars)
out=_dsin nodupkey ;
            by &subgroup &trtvar ;
        run ;

        %mu_nobs (_dsin) ;

        %if &_dsin_nobs eq 0 %then %do ;

            *If non-subsetted input data also contains zero observations then fool
program to think that data exists. ;
            %if "&subgroup" ne "" %then %do ;
                %if &bign_in_data eq %then %do;
                    %let bign_in_data = bign;
                %end;
                                %* check for invalid IN_DATA or invalid variables
in IN_DATA;
                                %mu_check_data_and_var_exist
                                    ( data_to_check                     =
&bign_in_data
                                    , vars_to_check_in_all_data         =
                                    , vars_to_check_in_respective_data  = &subgroup
                                    , abort_if_does_not_exist           = yes
                                    , help                              = no
                                    ) ;
                proc sort data = &bign_in_data (keep=&subgroup) out=dummy nodupkey;
                    by &subgroup ;
                run;

            %end ;

            %mu_var_attributes
                ( datasets  = _dsin
                , variables = &usubjid
                                %if "&byvars" ne "" %then %do ;
                                    %do i = 1 %to &num_byvars ;
```

```
                                    &&byvar&i
                                %end ;
                            %end ;
                           &trtvar
                , debug     = &debug
                , help      = &help
                ) ;

            data dsin;
                %if "&subgroup" ne "" %then %do ;
                    set dummy ;
                    format
                        %do i = 1 %to &num_subgroup ;
                            &&subgroupvar&i.
&&&&&muva_in_data._&&subgroupvar&i.._FMT
                        %end ;
                    ;
                %end ;
                %if "&&_dsin_&usubjid._vartyp" eq "C" %then %do ;
                    length &usubjid $ &&_dsin_&usubjid._len ;
                    &usubjid='A' ;
                %end ;
                %else %do ;
                    &usubjid = 1 ;
                %end ;
                %if "&byvars" ne "" %then %do ;
                    %do i = 1 %to &num_byvars ;
                        %if "&&&&_dsin_&&byvar&i.._vartyp" eq "C" %then %do ;
                            length &&byvar&i $ &&&&_dsin_&&byvar&i.._len ;
                            %if "&&fmt&i" ne "BLANK" %then %do ;
                                &&byvar&i = "&&byvar_start&i" ;
                            %end ;
                            %else %do ;
                                &&byvar&i = 'A' ;
                            %end ;
                        %end ;
                        %else %do ;
                            %if "&&fmt&i" ne "BLANK" %then %do ;
                                &&byvar&i = &&byvar_start&i ;
                            %end ;
                            %else %do ;
                                &&byvar&i = 1 ;
                            %end ;
                        %end ;
                    %end ;
                %end ;
                %if "&&_dsin_&trtvar._vartyp " eq "C" %then %do ;
                    length &trtvar $ &&&trtvar._len ;
                    %if "&trtvar_preloadfmt" ne "" %then %do ;
                        &trtvar = "&trtvar_start" ;
```

```
            %end ;
            %else %do ;
                &trtvar = 'A' ;
            %end ;
        %end ;
        %else %do ;
            %if "&trtvar_preloadfmt" ne "" %then %do ;
                &trtvar = &trtvar_start ;
            %end ;
            %else %do ;
                &trtvar = 1 ;
            %end ;
        %end ;

        %*create varables specified in VAR and set them to missing*;
        %do i=1 %to &num_summ_stats_vars;
            &&summ_stats_var&i = .;
        %end;
    run ;

    data dsin ;
        set dsin ;
        %if "&trtvar_preloadfmt" ne "" %then %do ;
            format &trtvar &TRTVAR_PRELOADFMT.. ;
        %end ;
        %if "&byvars" ne "" %then %do ;
            %do i = 1 %to &num_byvars ;
                %if "&&fmt&i" ne "BLANK" %then %do ;
                    format &&&byvar&i &&fmt&i... ;
                    new_&&&byvar&i = put(&&&byvar&i,&&fmt&i...) ;
                %end ;
            %end ;
        %end ;
    run ;

%end ;

%else %do;

    data dsin ;
        set _dsin ;
        %do i=1 %to &num_summ_stats_vars;
          &&summ_stats_var&i =. ;
        %end;
        %if "&trtvar_preloadfmt" ne "" %then %do ;
            format &trtvar &TRTVAR_PRELOADFMT.. ;
        %end ;
        %if "&byvars" ne "" %then %do ;
            %do i = 1 %to &num_byvars ;
                %if "&&fmt&i" ne "BLANK" %then %do ;
```

```
                            format &&&byvar&i &&fmt&i... ;
                            new_&&&byvar&i = put(&&&byvar&i,&&fmt&i...) ;
                        %end ;
                    %end ;
                %end ;
            run ;
        %end ;

    %end ;



%*** STEP 3 - CREATE SUBTOTALS AND TOTAL GROUPS AS REQUESTED *** ;
    *get the type and length of the treatment variable ;
    %let dsid_in_data   = %sysfunc(open(dsin)) ;
    %let trtvar_type    = %sysfunc(vartype(&dsid_in_data,
%sysfunc(varnum(&dsid_in_data, &trtvar)))) ;
    %let trtvar_length  = %sysfunc(varlen(&dsid_in_data,
%sysfunc(varnum(&dsid_in_data, &trtvar)))) ;
    %let close_in_data  = %sysfunc(close(&dsid_in_data)) ;

    %if &define_total_groups eq %str( ) or %index(&define_total_groups, =) eq 0
%then %do ;
        %put %str(ALERT_)I:
------------------------------------------------------------------------------
-- ;
        %put %str(ALERT_)I: DEFINE_TOTAL_GROUPS paramenter is missing or does not
specify a total. ;
        %put %str(ALERT_)I: No subtotal and/or total groups will be calculated. ;
        %let define_total_groups = ;
        %if &help eq Y %then %do ;
            %put HELP: To define subtotal and/or total groups enter the definition
of the group as: ;
            %put HELP: list of treatments = new group ! list of treatments = new
group. ;
            %put HELP: For example: TRTC is in the data with values of 'A' 'B' and
'C' and the analysis ;
            %put HELP: calls for a subtotal of 'A' and 'B' along with an overall
total of all three treatments. ;
            %put HELP: set DEFINE_TOTAL_GROUPS = 'A' 'B' = 'subtotal' ! 'A' 'B' 'C'
= 'total' ;
            %put HELP: The left side of the equals sign must be in the syntax of
the actual data and will be ;
            %put HELP: used in an IN statement, ie, TRTC in ('A' 'B')  (no
commas!). ;
            %put HELP: The right side of the equals sign will be the value of the
new group and must be of the same ;
            %put HELP: type as the existing treatment group variable. ;
            %put HELP: Separate each subtotal/total definition with the exclamation
```

```
point (!) ;
            %put HELP: Make sure to include these new groups in the value statement
of the format that will be ;
            %put HELP: assigned to the TRTVAR_PRELOADFMT parameter. ;
        %end ;
        %put %str(ALERT_)I:
------------------------------------------------------------------------------
-- ;
    %end ;
    %else %do ;
        %mu_wordscan
            ( string    = &define_total_groups
            , root      = total
            , numw      = numtotals
            , delim     = !
            ) ;
        %do i = 1 %to &numtotals ;
            %mu_wordscan
                ( string    = &&total&i
                , root      = side&i._
                , numw      = numside
                , delim     = %str(=)
                ) ;
        %end ;
    %end ;

    %if &define_total_groups ne %str( ) %then %do ;
        ** START CROSSOVER SUBTOTAL / TOTAL GROUPS ;
        %if &trtvar_type eq C %then %do ;
            * may need to redefine the length of the treatment variable - consider
the new treatment group names ;
            data _null_ ;
                call symput
                    ( 'new_trtvar_length'
                    , trim(left(put(max(
                        %do i = 1 %to &numtotals ;
                            length(&&side&i._2),
                        %end ;
                            &trtvar_length), 8.)))
                    ) ;
            run ;
        %end ;

        %* cycle through each definition of subtotal/total and create new
observations for each subject in each group ;
        %do i = 1 %to &numtotals ;

            * keep one record per subject per group of treatments ;
            proc sort data=dsin out=total&i(drop=&trtvar) ;
                by &usubjid ;
```

```
            where &trtvar in (&&side&i._1) ;
        run ;

        * create the new treatment group ;
        data total&i ;
            %if &trtvar_type eq C %then %do ;
                length &trtvar $ &new_trtvar_length ;
            %end ;
            set total&i ;
            retain &trtvar &&side&i._2 ;
        run ;

    %end ;

    *set the new treatment groups together with the original data ;
    data dsin ;
        %if &trtvar_type eq C %then %do ;
            length &trtvar $ &new_trtvar_length ;
        %end ;
        set dsin
            %do i = 1 %to &numtotals ;
                total&i
            %end ;
        ;
    run ;

    proc sort data=dsin ;
        by &subgroup &usubjid &trtvar ;
    run ;

    *END OF SUBTOTAL / TOTAL GROUPS ;
%end ;


%*** STEP 4 - COUNT ;
    %let at_least_one_byvar_no_format = 0 ;

    %if "&byvars" ne "" %then %do ;
        %do i = 1 %to &num_byvars ;
            %if "&&fmt&i" = "BLANK" %then %let at_least_one_byvar_no_format = 1 ;
        %end ;
    %end ;

    %if &at_least_one_byvar_no_format = 1 %then %do ;
        proc sort data=dsin ;
            by &subgroup
                %if "&byvars" ne "" %then %do ;
                    %do i = 1 %to &num_byvars ;
                        %if "&&fmt&i" eq "BLANK" %then %do ;
                            &&byvar&i
```

```
                    %end ;
                %end ;
            %end ;
        ;
    run ;
%end ;

%*R18;
proc sort data=dsin out=unique dupout=dups nodupkey;
    by &subgroup.
        %if "&byvars" ne "" %then %do ;
            %do i = 1 %to &num_byvars ;
                &&byvar&i
            %end ;
        %end ;
       &trtvar &usubjid;
run;
%mu_nobs (dups) ;
%MU_CHECKRC
(CONDITION = %str(&dups_nobs gt 0)
,RC       = 2
,rcprefix = ma_summ_stats
,MSG1     = SUBGROUP/BY/CLASS/USUBJID variables %qupcase(&subgroup &byvars
&trtvar &usubjid)
,MSG2     = do not yield a unique record
,MSG3     = Program will ABORT
,ALERTID  = P
,SETABORT = Y
);

proc means data=dsin noprint nway completetypes missing ;
    by &subgroup
        %if "&byvars" ne "" %then %do ;
            %do i = 1 %to &num_byvars ;
                %if "&&fmt&i" eq "BLANK" %then %do ;
                    &&byvar&i
                %end ;
            %end ;
        %end ;
    ;

    %if "&byvars" ne "" %then %do ;
        %do j = 1 %to &num_byvars ;
            %if "&&fmt&j" ne "BLANK" %then %do ;
                class &&byvar&j / preloadfmt
                    %if &&byvar_mlf&j._nobs gt 0 %then %do ;
                        mlf
                    %end ;
                ;
            %end ;
```

```
            %end ;
        %end ;
        class &trtvar / preloadfmt mlf ;
        var  &vars ;
        output out=summ_stats (drop=_type_ _freq_)
            %do l = 1 %to &num_summ_stats ;
                %if %upcase("&&summ_stats&l") eq %upcase("sd") %then %do ;
                    std =
                %end ;
                %else %if %upcase("&&summ_stats&l") eq %upcase("med") %then %do ;
                    median =
                %end ;
                %else %do ;
                    &&summ_stats&l =
                %end ;
                %do m = 1 %to &num_summ_stats_vars ;
                    var&m._&&summ_stats&l     /*&&summ_stats_var&m..*/
                %end ;
            %end ;
        ;
    run ;


    %* order byvars - internal, formatted, notsorted, ascending, descending if
there is format associated with shift_from and shift_to vars ;

    %md_add_byvar_order(in_data = summ_stats
                      , byvars=&byvars
                      , out_data=summ_stats
                      , subgroup=&subgroup
                      ,byvars_preloadfmt=&byvars_preloadfmt
                      ,byvars_sort_order=&byvars_sort_order
                      ,byvars_sort_asc_or_desc=&byvars_sort_asc_or_desc
                      , rootbyvars= );


%let temp_macro_name = &sysmacroname;
%macro md_check_integer(param_name =, value=) / des='Defined in MA_SUMM_STATS';
 %let RE1=%sysfunc(prxparse(#\D#)) ;
 %put &RE1. ;
  %if %sysfunc(prxmatch(&re1.,&value))>0 %then %do ;
      %put %str(ALERT_)P: A NON-DIGIT character was found for parameter =
&param_name Value found was &value..;
      %put %str(ALERT_)P: &param_name must be a positive integer. Macro will abort;

      %let &temp_macro_name._RC = 3;
      %PUT INFO: &temp_macro_name._RC = &&&temp_macro_name._RC;
      %let abort = yes ;
  %end ;
%mend md_check_integer;
```

```
%if &_section_ eq %then %do;
    %let _section_=1;
%end;
%md_check_integer(param_name=_SECTION_, value=&_section_);

%if &abort = yes %then %do;
    %goto exit ;
%end;

data &out_data;
  retain &trtvar &subgroup _section_
            %if "&byvars" ne "" %then %do ;
                %do x = 1 %to &num_byvars ;
                    byvar&x._order &&byvar&x fmtd_&&byvar&x
                %end ;
            %end ;
            %do m = 1 %to &num_summ_stats_vars ;
                var&m
                %do l = 1 %to &num_summ_stats ;
                    var&m._&&summ_stats&l
                %end ;
            %end ;;
  set summ_stats;
%do n = 1 %to &num_summ_stats_vars ;
    length var&n $%length(&&&summ_stats_var&n);
    var&n=upcase(strip("&&&summ_stats_var&n"));
%end;
_section_=&_section_;
  keep &trtvar &subgroup _section_
            %if "&byvars" ne "" %then %do ;
                %do x = 1 %to &num_byvars ;
                    byvar&x._order &&byvar&x fmtd_&&byvar&x
                %end ;
            %end ;
            %do m = 1 %to &num_summ_stats_vars ;
                var&m
                %do l = 1 %to &num_summ_stats ;
                    var&m._&&summ_stats&l
                %end ;
            %end ;;
run;


%*R16;

proc sort data = &out_data;
   by &trtvar &subgroup _section_
      %if "&byvars" ne "" %then %do;
         %do __idx__ = 1 %to &num_byvars;
```

```
            byvar&__idx__._order &&byvar&__idx__ fmtd_&&byvar&__idx__
        %end ;
     %end ;
          ;
run;


%**remove variable label from the dataset **;
proc datasets lib=work nolist;
  modify &out_data;
  attrib _all_ label=' ';
quit;

%*** STEP 5 - END OF PROCESS TASKS ;
%* Clean up;


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


%PUT ---------------------------------------------- ;
%PUT INFO: (&SYSMACRONAME) ;
%PUT INFO: &sysmacroname._RC = &&&sysmacroname._RC;
%PUT -END------------------------------------------ ;

%exit:
%if &&&sysmacroname._RC gt 0 | %upcase("&abort") = YES %then %do ;
    data _null_ ;
        abort ;
    run ;
%end ;

%mend ma_summ_stats;
```