```
%macro ma_summ_stats_rules
(in_data =
,byvars =
,out_data = MA_SUMM_STATS_RULES
,help=
,debug=
) / store source des='V1.0.0.5';

%*************************************************************************
        PROGRAM NAME:   MA_SUMM_STATS_RULES.SAS
             VERSION:   1.0
   DIRECTORY/ACCOUNT:
              AUTHOR:   (b) (6)
        DATE CREATED:   20130213
             PURPOSE:   This macro will produce a rules data set that controls the
display of summary statistics
          CHECKED BY:
  SUPPORT PROGRAMMER:
             UPDATED:
             CHECKED:
         INPUT FILES:
        OUTPUT FILES:
  CALLING PROCEDURES:   called by table program
   CALLED PROCEDURES:   mu_help_debug
  CREATED PROCEDURES:
-------------------------------------------------------------------------------
Modification History
Date:               Programmer:             Notes:

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

%******RETURN CODES******
0 = no error
1 = missing out_data, default to &sysmacroname
2 = out_data already exists, will be replaced
3 = in_data provided, but byvars is blank - abort
;

%mu_help_debug
%let abort=no;
%global &sysmacroname._rc;

%let &sysmacroname._rc = 0;

%* get current setting of mprint and if set to DEBUG mode turn mprint on;
```

```
%md_workinfo(debug = &debug );

%if &out_data = %str() %then %do;
    %put ALERT_I: Missing OUT_DATA.  Default to &sysmacroname;
    %let out_data = &sysmacroname;
    %let &sysmacroname._rc = 1;
%end;


%* establish RULES data set;
data __rules__;
length stat $ 8 ;
%*descriptive statistics;
stat = 'CSS';       fixed=1; rule=3; output;
stat = 'CV';        fixed=1; rule=1; output;
stat = 'KURTOSIS';  fixed=1; rule=3; output;
stat = 'LCLM';      fixed=1; rule=1; output;
stat = 'MAX';       fixed=0; rule=0; output;
stat = 'MEAN';      fixed=0; rule=1; output;
stat = 'MIN';       fixed=0; rule=0; output;
stat = 'N';         fixed=1; rule=0; output;
stat = 'NMISS';     fixed=1; rule=0; output;
stat = 'RANGE';     fixed=0; rule=1; output;
stat = 'SKEWNESS';  fixed=1; rule=1; output;
stat = 'STDDEV';    fixed=0; rule=2; output;
stat = 'STDERR';    fixed=0; rule=2; output;
stat = 'SUM';       fixed=0; rule=0; output;
stat = 'SUMWGT';    fixed=0; rule=0; output;
stat = 'UCLM';      fixed=1; rule=1; output;
stat = 'USS';       fixed=1; rule=1; output;
stat = 'VAR';       fixed=0; rule=2; output;
%*quantile statistics;
stat = 'MEDIAN';    fixed=0; rule=1; output;
stat = 'P1';        fixed=0; rule=1; output;
stat = 'P5';        fixed=0; rule=1; output;
stat = 'P10';       fixed=0; rule=1; output;
stat = 'Q1';        fixed=0; rule=1; output;
stat = 'Q3';        fixed=0; rule=1; output;
stat = 'P90';       fixed=0; rule=1; output;
stat = 'P95';       fixed=0; rule=1; output;
stat = 'P99';       fixed=0; rule=1; output;
stat = 'QRANGE';    fixed=0; rule=1; output;
%*hypothesis testing;
stat = 'PROBT';     fixed=1; rule=4; output;
stat = 'T';         fixed=1; rule=4; output;
run;


    %if %sysfunc(exist(&out_data)) %then %do;
        %put ALERT_I: &sysmacroname: data set &out_data already exists;
```

```
    %put ALERT_I:     data set will be replaced;
    %let &sysmacroname._rc = 2;
%end;

%if &in_data ne %str() %then %do;
    %mu_check_data_and_var_exist
    ( data_to_check = &in_data
    , vars_to_check_in_all_data = &byvars
    , abort_if_does_not_exist = yes
    , help = &help
    )

    %if %quote(&byvars) ne %str() %then %do;
     proc sort data=&in_data(keep=&byvars) out=__byvars__ nodupkey;
      by &byvars;
     run;


     data &out_data;
            set __byvars__;
            do i = 1 to xnobs;
            set __rules__ nobs=xnobs point=i;
            output;
            end;
    run;
   %end;
   %else %do;
    %put ALERT_I: &sysmacroname.: Parameter IN_DATA (&in_data) provided.;
    %put ALERT_I:   But, parameter BYVARS is blank.;
    %put ALERT_I:   Standard Rules data set will be created as &out_data.;
    %let &sysmacroname._rc = 3;
           data &out_data;
            set __rules__;
           run;
   %end;

%end;
%else %if &in_data eq %str() %then %do;
  %if %quote(&byvars) ne %str() %then %do;
    %put ALERT_P: &sysmacroname.: Parameter IN_DATA is missing.;
    %put ALERT_P:   But, parameter BYVARS is not blank.;
    %put ALERT_P:   Macro will abort;
    %let &sysmacroname._rc = 4;
    %let abort = yes;
    %goto exit;
  %end;

   data &out_data;
    set __rules__;
   run;
```

```
    %end;




%exit:

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


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


%PUT ------------------------------------------------- ;
%PUT INFO: &sysmacroname._RC = &&&sysmacroname._RC ;
%PUT ------------------------------------------------- ;


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

%mend ma_summ_stats_rules;
```