```
%macro ma_summ_stats_precision(
 IN_DATA =
,VAR =
,PRECISION_VAR =
,BYVARS =
,FMT_OVERRIDE =
,PRECISION_THRESHOLD_ALERT = 4
,OUT_DATA =
,DEBUG =
,HELP =
) /store source des='V1.0.0.10';
%*************************************************************************
FILENAME:    MA_SUMM_STATS_PRECISION.SAS
DEVELOPER:   (b) (6)
PLATFORM:    SAS 9.1.3, 9.2 on PC
MACROS USED:
ASSUMPTIONS:
DESCRIPTION:

This macro .

USAGE NOTES:

IN_DATA = input data used to create order vars.  In the form LIB.DATA, where lib
may be ommitted if WORK
  (default = blank) (REQUIRED)

VAR = Single Variable in the IN_DATA for which the precision will be calculated
  (default = blank) (REQUIRED)
PRECISION_VAR = The name of the single variable to hold the numeric precision
result
  (default = blank) (REQUIRED)
BYVARS = Variables in the IN_DATA data set for which the precision value will be
derived for each unique combiniation of the BYVARS
         If FMT_OVERRIDE is specified, BYVARS must only include a single variable
  (default = blank) (REQUIRED)

OUT_DATA = the output data set created.
  (default = blank) (OPTIONAL)

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)

FMT_OVERRIDE = Format to overrides the derived value from the VAR parameter or the
copied version from the PRECISION_VAR
```

```
               parameter for PRECISION
               (Default = blank) (OPTIONAL)
*********************************************************************************;

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


%**** STEP 1 - PARAMETER AND DATA CHECKS ****;

%global &sysmacroname._RC;
%let &sysmacroname._RC = 0;

%let currentmacro = &sysmacroname;
/*******
%*R1: &IN_DATA does not exist: MU_CHECK_DATA_AND_VAR_EXIST_RC=1 abort
      &IN_DATA is empty: MU_CHECK_DATA_AND_VAR_EXIST_RC=3 abort
%*R2: &VAR not provided: MU_CHECK_REQ_PARAMETERS_RC=1, exit
      &VAR is a char variable: SUMM_STATS_PRECISION_RC =1 , exit
      &VAR provided more than one variable: SUMM_STATS_PRECISION_RC=3 exit
%*R3  &VAR does not exist: MU_CHECK_DATA_AND_VAR_EXIST_RC=2 exit
%*R4  PRECISION will be created, unless specified in PRECISION_VAR
%*R5  &PRECISION_VAR does not exist: MU_CHECK_DATA_AND_VAR_EXIST_RC = 2 exist
      &PRECISION_VAR exist: copy integer to PRECISION
      &PRECISION_VAR provided more than one variable: SUMM_STATS_PRECISION_RC=4
exit

%*R6  &PRECISION_VAR is char, convert to num
%*R7  FMT_OVERRIDE: format does not exist, MA_SUMM_STATS_PRECISION_RC =2 exit
%*R8  &OUT_DATA = &IN_DATA._PRECISION if &OUT_DATA is blank
%*R9  &BYVARS does not exist: MU_CHECK_DATA_AND_VAR_EXIST_RC = 4 exit


%*R: FM_OVERRIDE is not blank, but BYVARS is blank
******/



%let abort = no;

%mu_help_debug;

%*  If IN_DATA parameter is not supplied then leave the macro without any errors;
%if &in_data eq %then %do;
    %put %str(ALERT_)I:
----------------------------------------------------------------------;
    %put %str(ALERT_)I: &SYSMACRONAME: No value is supplied in IN_DATA parameter.;
```

```
    %put %str(ALERT_)I: &SYSMACRONAME: PRECISION variable will not be derived.;
        %put %str(ALERT_)I:
----------------------------------------------------------------------;
    %goto exit ;
%end;

%* get a list of the datasets in the WORK library before starting to enable cleanup
at the end of this macro;
%md_workinfo(debug = &debug );


%*R1 and R3;
%* CHECK: invalid &IN_DATA or invalid &VAR in &IN_DATA;
%mu_check_data_and_var_exist(
data_to_check = &in_data
,vars_to_check_in_all_data = &var
,abort_if_does_not_exist = Yes
,help = no);

%*R2;
%*CHECK: &VAR must be numeric;
%mu_var_attributes
    ( datasets  = &in_data
     , variables = &var
     , debug=&debug.
     ) ;

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

%*put checkA;


%MU_CHECKRC(condition=&&&muva_in_data._&var._vartyp ne N,
            rc=1,
            msg1=%bquote(&VAR must be a numeric variable - abort),
            rcprefix= &currentmacro,
            alertid=P,
            setabort=Y);


%*put checkB;

%*R2;

%* CHECK: required parameters VAR are blank;
%mu_check_req_parameters(
 parameters_to_check = IN_DATA VAR
,help = no);
```

```
%if &MU_CHECK_REQ_PARAMETERS_RC gt 0 %then %goto EXIT;



%*put checkC;

%*R9;
%* CHECK: BYVARS parameter;
%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 = no);



%*put checkD;
%if %length(&BYVARS)>0 %then
  %mu_wordscan(string=&BYVARS, root=BYVAR, numw=BYVAR_NUM);
  %else %let BYVAR_NUM=0;



%*put checkE;
%*R8;
%* Populate with defaults if certain parameters are blank;
%if %length(&out_data)=0 %then %do;
  %let OUT_DATA = &IN_DATA._PRECISION;
%end;


%*put checkF;
%*precision_var;
%if %length(&PRECISION_VAR)>0 %then
  %mu_wordscan(string=&PRECISION_VAR, root=PRECISION_VAR, numw=PRECISION_VAR_NUM);
%else %let PRECISION_VAR_NUM=0;

%*R5;
%*put checkG;
%*if more than one precision_var included, abort**;
%if &precision_var_num>1 %then %do;
 %let &sysmacroname._RC=4 ;
    %put %str(ALERT_)P:
----------------------------------------------------------------------;
    %put %str(ALERT_)P: &SYSMACRONAME: There are more than one variable specified
in PRECISION_VAR: &PRECISION_VAR..;
    %put %str(ALERT_)P: &SYSMACRONAME - Program will ABORT. ;
    %let abort = yes ;
    %goto exit ;
%end;
```

```
%if &precision_var_num>0 %then %do;
%*put checkI;
%*R5;

  %*check if PRECISION_VAR exists in the dataset;
  %mu_check_data_and_var_exist(
    data_to_check = &in_data
   ,vars_to_check_in_all_data = &precision_var
   ,abort_if_does_not_exist = Y
   ,help = no);
 /*
  %MU_CHECKRC(condition=&MU_CHECK_DATA_AND_VAR_EXIST_RC eq 4,
              rcprefix = &currentmacro
              rc =4,
              msg1=%str(The variable &PRECISON_VAR specified in PRECISON_VAR does
not exist - abort),
              alertid=P,
              setabort=Y);
 */

%*put checkJ;
%*check if PRECISION_VAR is character variable or numerica variable;
%mu_var_attributes
    ( datasets  = &in_data
     , variables = &precision_var
     , debug=&debug.
     ) ;
%end;


%*check if PRESCISION already exist in the dataset;
%let DSID = %sysfunc(open(&in_data));
%let EXIST_PRECISION = NO;

%if %sysfunc(varnum(&dsid, PRECISION)) ne 0 %then %do;
    %let EXIST_PRECISION = YES;
%end;
%let DSID_RC = %sysfunc(close(&dsid));

%if &exist_precision=YES %then %do;
%*put checkK;
%*check if PRECISION_VAR is character variable or numerica variable;
 %mu_var_attributes
    ( datasets  = &in_data
     , variables = precision
     , debug=&debug.
     ) ;
%end;
```

```
%if &exist_precision=YES and &precision_var ne %then %do;

%*put checkL;
   %if %upcase(&precision_var) ne PRECISION %then %do;
    %let &SYSMACRONAME._RC=1;
    %put %str(ALERT_)P:
------------------------------------------------------------------------;
    %put %str(ALERT_)P: &SYSMACRONAME: Variable PRECISION exists in the IN_DATA,
but specified variable in;
    %put %str(ALERT_)P: &SYSMACRONAME: PRECISION_VAR is not PRECISION.;
    %put %str(ALERT_)P: &SYSMACRONAME - Program will ABORT. ;
    %let abort = yes ;
    %goto exit ;
   %end;

%end;


%*R2;
%* should have only 1 variable specified in &VAR;
%mu_wordscan(string=&VAR, root=VAR, numw=VAR_NUM);
%if &var_num.>1 %then %do ;
  %let &sysmacroname._RC=3 ;
    %put %str(ALERT_)P:
------------------------------------------------------------------------;
    %put %str(ALERT_)P: &SYSMACRONAME: There are more than one variable specified
in VAR: &VAR..;
    %put %str(ALERT_)P: &SYSMACRONAME - Program will ABORT. ;
    %let abort = yes ;
    %goto exit ;
%end ;

%*put checkC;

%if &PRECISION_THRESHOLD_ALERT = %str() %then %do;
    %put %str(ALERT_)I:
------------------------------------------------------------------------;
    %put %str(ALERT_)I: &SYSMACRONAME: Parameter PRECISION_THRESHOLD_ALERT is
missing;
    %if &HELP=Y %then %do;
    %put %str(ALERT_)I: HELP: Use this parameter as a cutoff to ALERT the user when
a precision exceeds the given threshold;
    %end;
%end;

%**** STEP 1 - END ****;


%**** STEP 2 - PROCESS DATA ****;
```

```
%macro test;
%* Get type of input variables (char/num);
%do i=1 %to &VAR_NUM;
  proc sql noprint;
    select upcase(TYPE) into :VAR_TYPE&i
    from sashelp.VCOLUMN
    where LIBNAME=ifc(index("&IN_DATA","."),upcase(scan("&IN_DATA",1,.)),"WORK")
and
          MEMNAME=coalescec(upcase(scan("&IN_DATA",2,.)),upcase("&IN_DATA")) and
          upcase(NAME) = "%upcase(&&VAR&i)";
  quit;
%end;

%* Calculate precision;
data DS1;
  set &in_data;
  %do i=1 %to &VAR_NUM;
    %if &&VAR_TYPE&i = CHAR %then %do;
      %* Process character variables;
          if &&VAR&i='' then do;
            _PREC_INT&i = 0;
                _PREC_DEC&i = 0;
          end;
          else do;
        _PREC_INT&i = length(  ifc(index(&&VAR&i,"."), scan(strip(&&VAR&i),1,'.'),
strip(&&VAR&i))  );
        _PREC_DEC&i = ifn( index(&&VAR&i,"."), length(scan(strip(&&VAR&i),2,.')),
0 ) ;
          end;
    %end;
    %else %do;
      %* Process numeric variables;
          if &&VAR&i=. then do;
            _PREC_INT&i = 0;
                _PREC_DEC&i = 0;
          end;
          else do;
        _PREC_INT&i = length(  ifc(index(put(&&VAR&i,?best32.),"."),
scan(strip(put(&&VAR&i,?best32.)),1,'.'), strip(put(&&VAR&i,?best32.)))  );
        _PREC_DEC&i = ifn( index(put(&&VAR&i,?best32.),"."),
length(scan(strip(put(&&VAR&i,?best32.)),2,.')), 0 ) ;
          end;
    %end;
  %end;
run;

%* Get precision for each BYVARS block;
proc sql;
```

```
  create table DS2 as
    select *
    %do i=1 %to &VAR_NUM;
      ,max(_PREC_INT&i) as _PREC_MAXINT&i
      ,max(_PREC_DEC&i) as _PREC_MAXDEC&i
    %end;
    from DS1
    %if &BYVAR_NUM>0 %then %do;
      group by  %do i=1 %to &BYVAR_NUM ;
          %if &i.=1 %then &&BYVAR&i ;
            %else  , &&BYVAR&i ;
        %end;
    %end;
    ;
quit;

data &OUT_DATA;
  set DS2;
  %do i=1 %to &VAR_NUM;
  /*length &&PRECISION_VAR&i $10;*/
  &&PRECISION_VAR&i = _PREC_MAXDEC&i;
  %* Uncomment to get precision values like '2.3' (both integer and decimal parts);
  %*&&PRECISION_VAR&i = catx('.',_PREC_MAXINT&i,_PREC_MAXDEC&i);
  %end;
  drop _PREC_:;
run;
%mend;

%if &PRECISION_VAR_NUM=0 and &EXIST_PRECISION = NO %then %do;
%*put checkM;
    %let PRECISION_VAR1 = PRECISION;
    %test;
%end;
%else %if &PRECISION_VAR_NUM=0 and &EXIST_PRECISION = YES %then %do;
    %if &&&muva_in_data._precision_vartyp   = N %then %do;
    %*put checkN;
          data &out_data;
            set &in_data;
            precision = int(precision);
          run;
    %end;
    %else %if &&&muva_in_data._precision_vartyp  = C %then %do;
    %*put checkO;
          data &out_data;
            set &in_data(rename=(precision=__precision_c__));
            precision = int(input(strip(__precision_c__),?best32.));
            drop __precision_c__;
          run;
    %end;
%end;
```

```
%else %if &precision_var_num>0  %then %do;
    %if &&&muva_in_data._&precision_var._vartyp   = N %then %do;
    %*put checkP;
          data &out_data;
            set &in_data(rename=(&precision_var1=__precicion_c__));
            precision = int(__precicion_c__);
            drop __precicion_c__;
          run;
    %end;
    %else %if &&&muva_in_data._&precision_var._vartyp  = C %then %do;
    %*put checkQ;
          %*R6;
          data &out_data;
            set &in_data(rename=(&precision_var1=__precicion_c__));
            precision = int(input(strip(__precicion_c__),?best32.));
            drop __precicion_c__;
          run;
    %end;
%end;

%if &FMT_OVERRIDE ne %then %do;
%*put checkR;
   %if &BYVAR_NUM>1 %then %do;
      %let &SYSMACRONAME._RC=1;
      %put %str(ALERT)_P:
----------------------------------------------------------------------------;
      %put %str(ALERT)_P: &SYSMACRONAME: FMT_OVERRIDE is specified, multiple
variables found in BYVARS;
      %put %str(ALERT)_P: &SYSMACRONAME: &BYVARS;
      %put %str(ALERT)_P: &SYSMACRONAME - Program will ABORT. ;
      %let abort = yes ;
      %goto exit ;
   %end;
   %else %if %bquote(&byvars) = %then %do;
      %let &SYSMACRONAME._RC=1;
      %put %str(ALERT)_P:
----------------------------------------------------------------------------;
      %put %str(ALERT)_P: &SYSMACRONAME: FMT_OVERRIDE is specified, no variable
found in BYVARS;
      %put %str(ALERT)_P: &SYSMACRONAME - Program will ABORT. ;
      %let abort = yes ;
      %goto exit ;
   %end;

   %mu_var_attributes
    ( datasets  = &in_data
     , variables = &byvars
     , debug=&debug.
     ) ;
```

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


       %**check if fmt&i can be loaded or not, if not then abort the program*;
       proc format lib=work
          cntlout=tmpfmt (where=(fmtname=upcase(compress("&FMT_OVERRIDE",'$.')))) ;
       run ;

       %mu_nobs(tmpfmt);

       %if &tmpfmt_nobs = 0   %then %do;
       %*put checkS;
         %let &SYSMACRONAME._RC = 2 ;
         %put %str(ALERT)_P: &SYSMACRONAME: Format specified in FMT_OVERRIDE cannot
be found/loaded;
         %put %str(ALERT)_P: &SYSMACRONAME - Program will ABORT. ;
         %let abort = yes ;
         %goto exit ;
       %end;

       data tmpfmt2;
         set tmpfmt;

         %if %upcase(&&&muva_in_data._&byvars._vartyp)=C %then %do;
            length _tmpbyvar_ $200;
            _tmpbyvar_=strip(upcase(start));
            precision_from_format = input(strip(label),best32.);
         %end;
         %else %if  %upcase(&&&muva_in_data._&byvars._vartyp)=N %then %do;
             do  _tmpbyvar_= input(strip(start),?best32.) to
input(strip(end),?best32.);
                precision_from_format = input(strip(label),best32.);
                output;
             end;
         %end;
       run;

       proc sort data = tmpfmt2 out=tmpfmt3 tagsort ;
         by _tmpbyvar_;
       run;

       data &out_data.2 ;
         set &out_data. ;
         %if %upcase(&&&muva_in_data._&byvars._vartyp)=C %then %do;
           length _tmpbyvar_ $200;
           _tmpbyvar_=upcase(&byvars);
         %end;
         %else %if  %upcase(&&&muva_in_data._&byvars._vartyp)=N %then %do;
```

```
          _tmpbyvar_ = &byvars;
        %end;
      run;

      proc sort data = &out_data.2 out=&out_data.3 tagsort ;
        by _tmpbyvar_;
      run;

      data &out_data ;
         merge &out_data.3(in=a) tmpfmt3(in=b);
         by _tmpbyvar_;
         if a;
         if b then precision = precision_from_format;
         drop _tmpbyvar_ precision_from_format;

      run;

%end;

%*put checkT;

%* Create output dataset;
proc sort data=&out_data(keep=&byvars precision) nodupkey;
       by &byvars precision;
run;

%* PRECISION THRESHOLD CHECKS;
%if &PRECISION_THRESHOLD_ALERT ne %str() %then %do;
       data _null_;
       set &out_data;
       by &byvars;
       retain flag 0;
       if flag = 0 and precision gt &PRECISION_THRESHOLD_ALERT then do;
         put "ALERT_C: Precision exceeds PRECISION_THRESHOLD_ALERT
(&PRECISION_THRESHOLD_ALERT). Consider override(s)";
         flag = 1;
       end;

       %if &byvar_num gt 1 %then %do;
         if first.&&byvar&byvar_num then do;
           if precision gt &PRECISION_THRESHOLD_ALERT then
               put "ALERT_C: PRECISON=" precision " for " %do i = 1 %to
&byvar_num; &&byvar&i %end; ;
           else
               put "PRECISON=" precision " for " %do i = 1 %to &byvar_num;
&&byvar&i %end; ;
         end;
       %end;

       %else %if &byvar_num = 1 %then %do;
```

```
          if first.&byvars then do;
            if precision gt &PRECISION_THRESHOLD_ALERT then
                put "ALERT_C: PRECISON=" precision " for &byvars=" &byvars;
            else
                put "PRECISON=" precision " for &byvars=" &byvars;
          end;
        %end;

        %else %do;
          if precision gt &PRECISION_THRESHOLD_ALERT then put "ALERT_C: PRECISON="
precision  ;
          else put "PRECISON=" precision  ;
        %end;

        run;
%end;



%**** STEP 2 - END ****;


%* Clean up;
proc catalog cat=work.sasmacr;
        delete test / et=macro;
quit;
%md_clean_and_reset(
    debug     =&debug
   ,_workdata = %str(&WORK_DATASETS_DATA &out_data)
   ,resetmprint = &mprint_setting
    );

%EXIT:

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


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

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

```
%mend ma_summ_stats_precision;
```