PharmaSUG 2017 - Paper PO20

Dealing with alignment of statistics: Beyond the decimal point.

Eric Pedraza, INC Research.

ABSTRACT

When displaying summarizations for continuous variables, we need to take several factors into account, such as dynamically assessing the required formats to obtain the desired alignment, values comprised of two statistics, a text to display in case the statistic cannot be computed, what is going to be aligned and in relation to what (i.e. a comma in a value comprised of two statistics aligned to the decimal point of single-statistic values), etc.

These are time consuming tasks that can be simplified by using a macro to take care of the issues mentioned above, and taking very little time from the programmer to parametrize.

This allows customizing the look and alignment of each separate statistic independently, thus it can be used for different studies with different display requirements.

INTRODUCTION

Customizing the look of summary tables of continuous variables is a time-consuming effort. Wouldn't it be great to be able to do it in just a few lines of code? How about just a single line for each statistic to be displayed? To this end, I propose the use of a macro that is very simple in the way it is parameterized.

MACRO OUTLINE

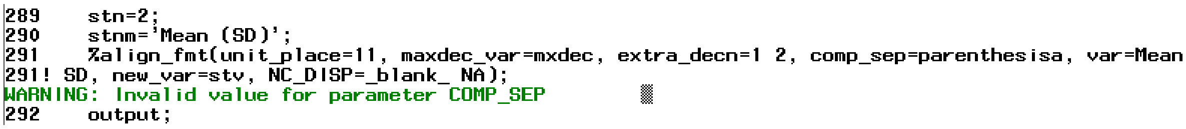
This macro was designed to be called within a DATA STEP. The DATA SET being read must contain the variables representing the statistics to be formatted (i.e. a resulting DATA SET from PROC MEANS or PROC UNIVARIATE, or containing values from statistical models like those created with PROC TTEST and ODS OUTPUT).

In order to disrupt the flow of a program as little as possible, this macro includes a couple of sections that help the user assess if anything is wrong with the way the macro is being used/parameterized.

Integrity Checks: The objective of this section is to insure the correct functioning of the macro, by making sure that parameters have permissible values and that parameters that are inter-related interact in the expected way.

This section also contains code that displays warnings/notes related to the issue found by the integrity checks. (For a more in-depth understanding of the macro and its required parameters please see the header of the macro and the macro itself in the Appendix A section of this paper).

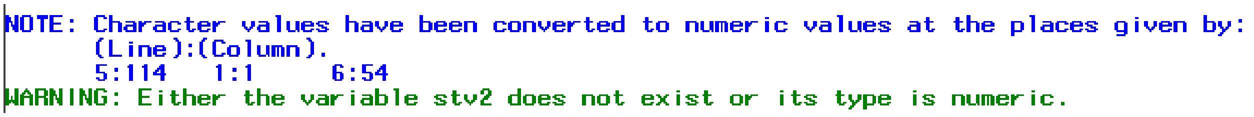
Figure 1. Custom warning message from integrity check of parameters.



```
289    stn=2;
290    stnm='Mean (SD)';
291    %align_fmt(unit_place=11, maxdec_var=mxdec, extra_decn=1 2, comp_sep=parenthesisa, var=Mean
291! SD, new_var=stv, NC_DISP=_blank_ NA);
WARNING: Invalid value for parameter COMP_SEP
292    output;
```

DATA STEP variable check: This Macro automatically creates some variables and defines their attributes within the DATA STEP it is invoked in. This is not the case for the variable(s) in which the aligned values will be stored, since depending on whether the desired DATA SET is narrow or wide, one or several char variables and their attributes must be defined beforehand (each macro call can use a different char variable, or all macro calls can use the same char variable), so a check was set in place to make sure the variable(s) exist and that it/they is/are not Numeric.

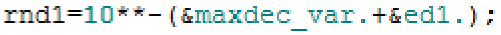
Figure 2. Custom warning message from DATA STEP variable check.



```
NOTE: Character values have been converted to numeric values at the places given by:
      (Line):(Column).
      5:114   1:1     6:54
WARNING: Either the variable stv2 does not exist or its type is numeric.
```

As for the actual formatting of the values, the process is as follows: a variable containing the maximum number of decimal places from the raw data must exist within the DATA SET being read. This variable will be used to help to dynamically generate the format to be used for the value. For single-statistic values, 2 variables are used: RND1 and FMT1. The first is a numeric variable that is assigned an operation that will yield the number representing the decimal precision the statistic result should be rounded to, using the maximum number of decimal places (remember the DATA SET must have such variable, and its name must be assigned to the parameter maxdec_var) along with the number of extra decimal places that was specified in the parameter extra_decn. This is necessary because in some rare occasions the format applied to a number does not round it properly.

Figure 3. Example of how rounding values are obtained.

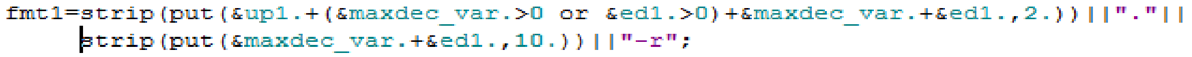


```
rnd1=10**-(&maxdec_var.+&ed1.);
```

The latter variable is used to store the format (in char string form) that will allow us to align the value properly. For single statistics, the base value used for width is the one defined in the unit_place parameter. We add the resulting value from the expression (&maxdec_var > 0 or &extra_decn > 0) so that if there are to be any decimal places, we add 1 to the width of the format to account for the decimal point. We also need to add the number of decimal places to the format width, using the values from parameter extra_decn, and the value stored in the variable specified in the parameter maxdec_var.

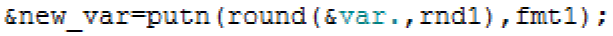
The unit of the integer part of the value will be at the place defined by the value of the parameter unit_place.

Figure 4. Example of how right-justifying formats are put together.



```
fmt1=strip(put(&up1.+(&maxdec_var.>0 or &ed1.>0)+&maxdec_var.+&ed1.,2.))||"."||
      strip(put(&maxdec_var.+&ed1.,10.))||"-r";
```

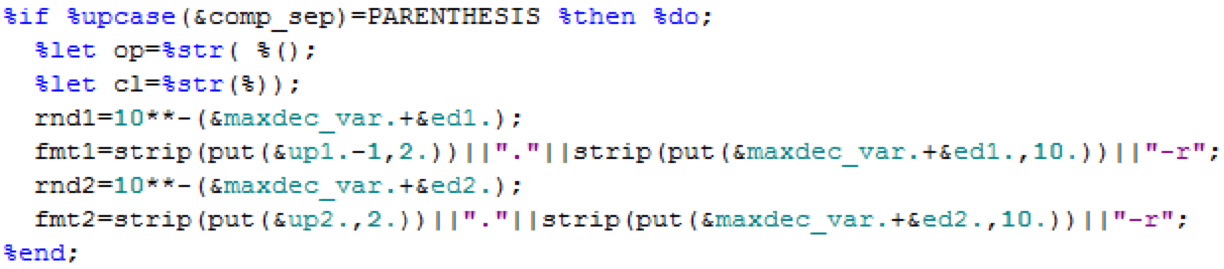
And lastly, we use the values of rnd1 and fmt1 to obtain the desired result:



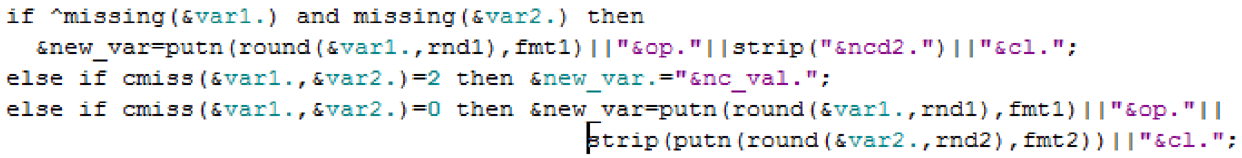
```
&new_var=putn(round(&var.,rnd1),fmt1);
```

For multi-statistic values the process is similar, but some other factors should be taken into account.

Figure 5. Sample code for the creation of formats to be used on multi-statistic values.



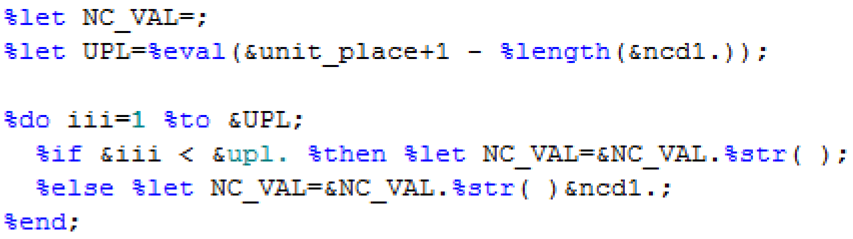
```
%if %upcase(&comp_sep)=PARENTHESIS %then %do;
  %let op=%str( %();
  %let cl=%str(%));
  rnd1=10**-(&maxdec_var.+&ed1.);
  fmt1=strip(put(&up1.-1,2.))||"."||strip(put(&maxdec_var.+&ed1.,10.))||"-r";
  rnd2=10**-(&maxdec_var.+&ed2.);
  fmt2=strip(put(&up2.,2.))||"."||strip(put(&maxdec_var.+&ed2.,10.))||"-r";
%end;
```



```
if ^missing(&var1.) and missing(&var2.) then
  &new_var=putn(round(&var1.,rnd1),fmt1)||"&op."||strip("&ncd2.")||"&cl.";
else if cmiss(&var1.,&var2.)=2 then &new_var.="&nc_val.";
else if cmiss(&var1.,&var2.)=0 then &new_var=putn(round(&var1.,rnd1),fmt1)||"&op."||
                                        strip(putn(round(&var2.,rnd2),fmt2))||"&cl.";
```

When a statistic is non-calculable, sometimes displaying a text is required, but for legibility's sake, more often than not, it should be aligned to a certain place. This macro, by default aligns the last letter of the text to the decimal point of single-statistic values. This is achieved by concatenating leading blanks the number of times resulting from subtracting the length of the text to the number specified by the parameter unit_place and adding 1, and in the last iteration concatenate the text as well.

Figure 6. Sample code for the alignment of texts when the statistic is non-calculable.

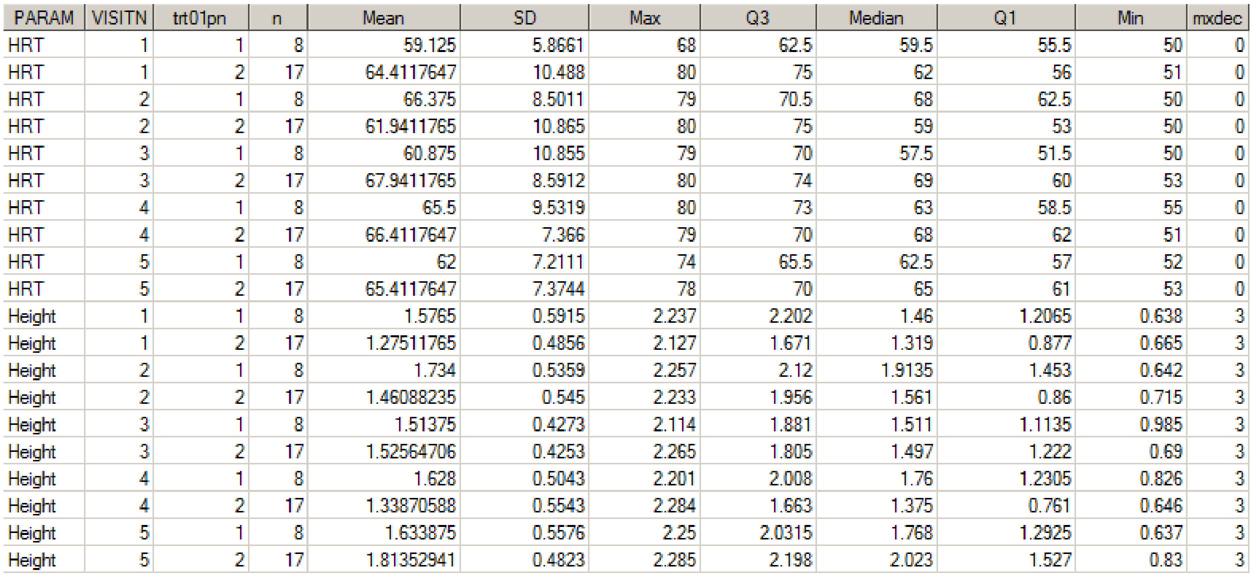


```
%let NC_VAL=;
%let UPL=%eval(&unit_place+1 - %length(&ncd1.));

%do iii=1 %to &UPL;
  %if &iii < &upl. %then %let NC_VAL=&NC_VAL.%str( );
  %else %let NC_VAL=&NC_VAL.%str( )&ncd1.;
%end;
```

MACRO USAGE

In order to use this macro, it must be called within a DATA STEP. A DATA SET with numeric variables, resulting from PROC UNIVARIATE, PROC MEANS or other SAS procedures can be used. The data set must also contain a variable containing the maximum number of decimals for the data before it was summarized. The values can differ according to the kind of data being summarized, for example, when reporting vital signs data, different parameters can have different maximum number of decimals.

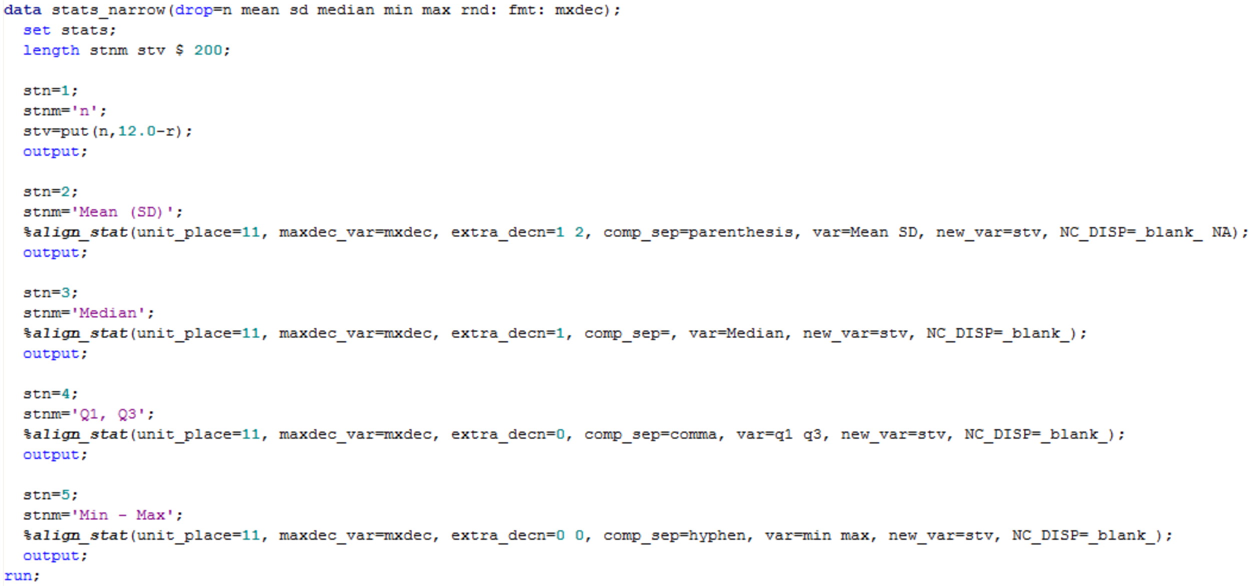
Figure 7. Sample data on which the macro can be used.



PARAM	VISITN	trt01pn	n	Mean	SD	Max	Q3	Median	Q1	Min	mxdec
HRT	1	1	8	59.125	5.8661	68	62.5	59.5	55.5	50	0
HRT	1	2	17	64.4117647	10.488	80	75	62	56	51	0
HRT	2	1	8	66.375	8.5011	79	70.5	68	62.5	50	0
HRT	2	2	17	61.9411765	10.865	80	75	59	53	50	0
HRT	3	1	8	60.875	10.855	79	70	57.5	51.5	50	0
HRT	3	2	17	67.9411765	8.5912	80	74	69	60	53	0
HRT	4	1	8	65.5	9.5319	80	73	63	58.5	55	0
HRT	4	2	17	66.4117647	7.366	79	70	68	62	51	0
HRT	5	1	8	62	7.2111	74	65.5	62.5	57	52	0
HRT	5	2	17	65.4117647	7.3744	78	70	65	61	53	0
Height	1	1	8	1.5765	0.5915	2.237	2.202	1.46	1.2065	0.638	3
Height	1	2	17	1.27511765	0.4856	2.127	1.671	1.319	0.877	0.665	3
Height	2	1	8	1.734	0.5359	2.257	2.12	1.9135	1.453	0.642	3
Height	2	2	17	1.46088235	0.545	2.233	1.956	1.561	0.86	0.715	3
Height	3	1	8	1.51375	0.4273	2.114	1.881	1.511	1.1135	0.985	3
Height	3	2	17	1.52564706	0.4253	2.265	1.805	1.497	1.222	0.69	3
Height	4	1	8	1.628	0.5043	2.201	2.008	1.76	1.2305	0.826	3
Height	4	2	17	1.33870588	0.5543	2.284	1.663	1.375	0.761	0.646	3
Height	5	1	8	1.633875	0.5576	2.25	2.0315	1.768	1.2925	0.637	3
Height	5	2	17	1.81352941	0.4823	2.285	2.198	2.023	1.527	0.83	3

Within a new DATA STEP, read a data set similar to the one in the previous example. In this DATA STEP the macro %ALIGN_STAT must be called for each statistic to be displayed. In the image below, there is an example in which a narrow data set is created. In addition to the macro calls, variables that will later serve for sorting purposes are added, and an observation per aligned statistic is output.

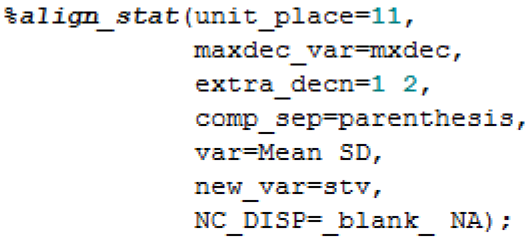
Figure 8. Sample code for the creation of a DATA SET with aligned values.



```
data stats_narrow(drop=n mean sd median min max rnd: fmt: mxdec);
  set stats;
  length stnm stv $ 200;

  stn=1;
  stnm='n';
  stv=put(n,12.0-r);
  output;

  stn=2;
  stnm='Mean (SD)';
  %align_stat(unit_place=11, maxdec_var=mxdec, extra_decn=1 2, comp_sep=parenthesis, var=Mean SD, new_var=stv, NC_DISP=_blank_ NA);
  output;

  stn=3;
  stnm='Median';
  %align_stat(unit_place=11, maxdec_var=mxdec, extra_decn=1, comp_sep=, var=Median, new_var=stv, NC_DISP=_blank_);
  output;

  stn=4;
  stnm='Q1, Q3';
  %align_stat(unit_place=11, maxdec_var=mxdec, extra_decn=0, comp_sep=comma, var=q1 q3, new_var=stv, NC_DISP=_blank_);
  output;

  stn=5;
  stnm='Min - Max';
  %align_stat(unit_place=11, maxdec_var=mxdec, extra_decn=0 0, comp_sep=hyphen, var=min max, new_var=stv, NC_DISP=_blank_);
  output;
run;
```

The following are brief descriptions of the parameters required by the macro %ALIGN_STAT, for more detailed information please refer to the Appendix A section, where the macro itself is presented along with instructions of use.

Figure 9. Sample macro call.



```
%align_stat(unit_place=11,
            maxdec_var=mxdec,
            extra_decn=1 2,
            comp_sep=parenthesis,
            var=Mean SD,
            new_var=stv,
            NC_DISP=_blank_ NA);
```

- UNIT_PLACE: It is the position, from left to right, at which the unit of the integer part of a number will be aligned for single-statistic values. For values comprised of 2 statistics, the separator will be aligned to the decimal point of single-statistic values.
- MAXDEC_VAR: The name of the variable containing the maximum number of decimals prior to summarization.
- EXTRA_DECN: The number of extra decimals that each statistic will have. Specify one value for single-statistics, and values separated by a space for composite values. Alternatively, if only one value is specified for composite statistics, then the same number of extra decimals will be used for both.
- COMP_SEP: This parameter should only be filled in for values comprised of two statistics. Currently the only permissible values are Parenthesis, Comma and Hyphen, which are the most common separators used. The values of used for the parameter are not case sensitive.
- VAR: The variable(s) containing the values to be aligned.

- NEW_VAR: The char variable that will contain the aligned value.
- NC_DISP: The text to be displayed when the statistic cannot be calculated. For single-statistic values just specify a text (only one word texts supported). For composite values specify two different texts, or specify just one text to be displayed for both statistics. If you need to have a blank instead of a text, then specify the keyword _BLANK_ (not case-sensitive).

THINGS TO CONSIDER WHEN USING THIS MACRO

Since the method of alignment used in this macro is reliant on the use of right justification in SAS formats, this macro will not work well in cases where the studies are setup in such a way that leading spaces are not allowed, for example, if the template for use in the reports (be they PDF, RTF, etc) does not support the option ASIS=ON within PROC REPORT.

There are only so many characters that can fit in a table column. As such we have to be mindful of the alignment position we assign each statistic, so it won't be scattered all over the place, or to insure statistics won't be wrapped incorrectly.

When using fonts that are not mono-spaced this macro may not be enough to obtain the desired results.

CONCLUSION

Summarization of continuous variables is a task that is performed quite often, and as such it is an area of opportunity for optimization through the use of the SAS MACRO facility. Use of this macro streamlines the process and makes it easier to customize the alignment of statistics in general.

ACKNOWLEDGMENTS

I would like to thank Thomas Kinghorn for his continuous support and providing feedback on this paper, Nadia Cardenas for her feedback, INC Research's Mexican statistical programming team for their feedback, and Beatriz Garcia for sharing her previous experiences on this conference.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Eric Pedraza
Enterprise: INC Research
Address: Via Gustavo Baz 2160 Caseta 3, Edificio 4, Zona Azul, Tlalnepantla, Estado de México
C.P. 54060
Phone: +52 55 7316 5323
E-mail: eric.pedraza@incresearch.com

APPENDIX A. CODE FOR MACRO %ALIGN_STAT

```
/*****************************************************************************
****************************/
/*  Macro align_stat
*/
/*
*/
/*  This macro serves the purpose of dynamically generating a format in order
to easily align           */
/*  descriptive statistics. You can also change the alignment of the
statistics by increasing or        */
/*  decreasing the value of the parameter UNIT_PLACE.
*/
/*
*/
/*  Considerarions before use: In order for it to work correctly, it must be
invoked within a data step */
/*  where the necessary variable(s) exist, said variables should contain the
descriptive statistics to  */
/*  be aligned, and a variable containing the maximum number of decimals for
the values before          */
/*  summarizing.
*/
/*
*/
/*  For compound statistics, by default the separator is aligned to the
decimal point of single sta-    */
/*  tistics (provided you specify the same value to the parameter UNIT_PLACE
of all statistics.)        */
/*
*/
/*  Parameters:
*/
/*    - UNIT_PLACE: The place at which the unit of the statistic will be
aligned to for single          */
/*                  statistics.
*/
/*                  For composite statistics, if the same number is
specified, the separator will be    */
/*                  aligned to the decimal point. (REQUIRED, DEFAULT is 8)
*/
/*    - MAXDEC_VAR: This is the name of the data set variable containing the
maximum number of decimals */
/*                  per group/data set. (REQUIRED)
*/
/*    - EXTRA_DECN: It is comoon that SAP specifies that some statistics
should have 1 or 2 more deci-  */
/*                  mal palces than the maximum number of decimal places
found in the values being      */
/*                  summarized. for single statistics specify just one value,
for compsite statistics   */
/*                  speicify one value if you want it to be used for both
statistics, or one value for  */
```

```
/*                   each statistic. (DEFAULT is 0).
*/
/*    - COMP_SEP:   Specify the separator for composite statistics. Valid
values are COMMA, HYPHEN and  */
/*                  PARENTHESIS (either lowecase, upcase or any combination).
(Required only for compo- */
/*                  statistics).
*/
/*    - VAR:        Specify the data set varible(s) that will supply the
values for the statistics.     */
/*                  Specify 2 varibles separated by a space for composite
statistics. (REQUIRED)        */
/*    - NEW_VAR:    Specify the name of a new varible to store
*/
/*    - NC_DISP:    Especify a value to be displayed when a value cannot be
calculated (i.e. SD when    */
/*                  when there is only one value. In single statistics, this
value is aligned to the    */
/*                  decimal point of other single statistics. In compound
statistics, the alignment is  */
/*                  dependent on the kind of separator used. IMPORTAN: Blank
spaces are not a valid     */
/*                  value. Instead specify the word _BLANK_ if you dont want
a label to be displayed.   */
/*                  For compound statistics specify two values separated by
an space, of if you spe-    */
/*                  cify only one value, it will be used for both statistics.
*/
/*
*/
/*****************************************************************************
****************************/


%macro align_stat( unit_place=8, maxdec_var=, extra_decn=0, comp_sep=, var=,
new_var=, NC_DISP=_BLANK_);

  %let bv=&unit_place.;

  %let w1=WAR;
  %let w2=NING;
  %let no1=NO;
  %let no2=TE;


  %* Parameter integrity checks (in order to make sure the parameters have
correct values, and to display a note or warning to the user
     or terminate the macro execution if needed);
  %if %sysfunc(countw(&unit_place)) ne 1 %then %do;
    %put &no.&no2.: More numbers than required or parameter UNIT_PLACE
missing.;
      %goto exit;
  %end;

  %if &maxdec_var=  %then %do;
    %put &w1.&w2.: Parameter MAXDEC_VAR required.;
    %goto exit;
  %end;

  %if &comp_sep ne   %then %do;
    %if %upcase(&comp_sep) ne COMMA and %upcase(&comp_sep) ne PARENTHESIS and
%upcase(&comp_sep) ne HYPHEN %then %do;
      %put &w1.&w2.: Invalid value for parameter COMP_SEP;
        %goto exit;
      %end;

      %if %sysfunc(countw(&extra_decn)) = 1 %then %do; %put &no1.&no2.: The
resulting value will be comprised of 2 statistis.;
                                                       %put &no1.&no2.- Only
one value for extradigits found.;

%put &no1.&no2.- It will be used for both statistics.;
    %end;
      %else %if %sysfunc(countw(&extra_decn)) ne 2 or  &extra_decn=   %then
%do;
        %put &w1.&w2.: Invalid or missing value for parameter EXTRA_DECN;
        %goto exit;
      %end;

      %if %sysfunc(countw(&NC_DISP)) = 1 %then %do; %put &no1.&no2.: The
resulting value will be comprised of 2 statistis.;
                                                       %put &no1.&no2.- Only
one value for NC_DISP found.;

%put &no1.&no2.- It will be used for both statistics.;
    %end;
      %else %if %sysfunc(countw(&NC_DISP)) ne 2 or  &NC_DISP=   %then %do;
        %put &w1.&w2.: Invalid or missing value for parameter NC_DISP;
        %goto exit;
      %end;

      %if %sysfunc(countw(&var.)) ne 2 or  &var.=   %then %do;
        %put &w1.&w2.: Invalid or missing value for parameter VAR.;
        %put &w1.&w2.- Two variables must be especified for composite
statistics.;
        %goto exit;
      %end;
  %end;

  %if &new_var=   %then %do;
    %put &w1.&w2.: The name of a new character variable to store the aligned
value must be specified;
      %goto exit;
  %end;


  %* check if the varible that will store the new statistic exists;

  call missing(&new_var);
  if vtype(&new_var)='N' then do;
    put "WAR" "NING: Either the variable &NEW_VAR. does not exist or its type
is numeric.";
    put "Please create it as a character variable before invoking this
macro";
```

```
  end;


  %* assign values from parameters to local macro variables for correct
execution of the macro ;
  %* local macro descriptions below: ;
  %* - UP1 and UP2: these varibles will store the same value as parameter
UNIT_PLACE separately in order to
       avoid confusion as to what parts of code are being used for each
statistic in compound values.;
  %* - ED1 and ED2: The same reasoning as UP1 and UP2 apply to these local
macro variables. In addition, if parameter EXTRA_DECN
       is assigned a single value, the same value will be used for both ED1
and ED2, if EXTRA_DECN is assigned 2 values, the first
       value will be assigned to ED1 and the second to ED3.;
  %* - NCD1 and NCD2: The same reasoning as UP1 and UP2 apply to these local
macro variables. In addition, if parameter NC_DISP
       is assigned a single value, the same value will be used for both NCD1
and NCD2, if NC_DISP is assigned 2 values, the first
       value will be assigned to NCD1 and the second to NCD3. If a no value
is to be displayed when a statistic cannot be computed
       then _blank_ should be specified (not case sensitive);

  %if &comp_sep ne %str() %then %do;
    %if %sysfunc(countw(&unit_place.))=1 %then %do; %let up1=&unit_place.;
%let up2=&unit_place; %end;

      %if %sysfunc(countw(&extra_decn.))=1 %then %do; %let ed1=&extra_decn.;
%let ed2=&extra_decn; %end;
      %else %if %sysfunc(countw(&extra_decn.))=2 %then %do; %let
ed1=%scan(&extra_decn.,1); %let ed2=%scan(&extra_decn,2); %end;

      %if %sysfunc(countw(&NC_DISP.))=1 %then %do;
      %if %index(%upcase(&NC_DISP.),_BLANK_) %then %do; %let ncd1=; %let
ncd2=; %end;
        %else %do; %let ncd1=&NC_DISP.; %let ncd2=&NC_DISP; %end;
      %end;
      %else %if %sysfunc(countw(&NC_DISP.))=2 %then %do iii=1 %to 2;
      %if %scan(%upcase(&NC_DISP.),&iii.)=_BLANK_ %then %let ncd&iii.=;
        %else %let ncd&iii.=%scan(&NC_DISP.,&iii.);
      %end;

    %* Align value for non calculable stats;
      %* The alignment of the value for non calculable stats is done as
follows:
         - Create local macro variable NC_VAL.
         - Subtract the length of NCD1 to the number in UNIT_PLACE
parameter+1 and add to the NC_VAL the same number of spaces as the
           resulting number from the subtraction, and then add the value of
NCD1.(Please note that if the same number is specified for UNIT_PLACE
           for all statistics, then the last character of the value in NCD1
will be aligned to the decimal point of sngle statistic values and to
           the separator in composite values.) ;
    %let NC_VAL=;
    %let UPL=%eval(&unit_place+1 - %length(&ncd1.));

    %do iii=1 %to &UPL;
```

```
      %if &iii < &upl. %then %let NC_VAL=&NC_VAL.%str( );
        %else %let NC_VAL=&NC_VAL.%str( )&ncd1.;
    %end;

      %* Local macro variables VAR1 and VAR2
       - For single-statistic values, VAR1 will contain the variable
specified in the parameter VAR. If parameter VAR contains
           the name of 2 variables, then VAR1 will be assigned the first and
VAR2 will be assigned the second.;
      %if %sysfunc(countw(&var.))=2 %then %do; %let var1=%scan(&var.,1); %let
var2=%scan(&var,2); %end;
      %else %if %sysfunc(countw(&var.))=1 %then %do; %let var1=&var; %let
var2=; %end;

      %* Create local macro variables OP and CL for composite-statistic
values. Their value will change as depending on the value of
         the parameter COMP_SEP.;
      %* Create DATA SET variables RND1 and RND2 that will contain the number
to be used within the ROUND function.;
      %* Create DATA SET variables FMT1 and FMT2 that will contain the
formats to be applied to each statistic.;


    fmt1='       ';
    fmt2='       ';


    %if %upcase(&comp_sep)=PARENTHESIS %then %do;
          %let op=%str( %();
          %let cl=%str(%));
        rnd1=10**-(&maxdec_var.+&ed1.);
        fmt1=strip(put(&up1.-
1,2.))||"."||strip(put(&maxdec_var.+&ed1.,10.))||"-r";
        rnd2=10**-(&maxdec_var.+&ed2.);

fmt2=strip(put(&up2.,2.))||"."||strip(put(&maxdec_var.+&ed2.,10.))||"-r";
    %end;
    %else %if %upcase(&comp_sep)=HYPHEN %then %do;
      %let op=%str( - );
        %let cl=;
        rnd1=10**-(&maxdec_var.+&ed1.);
        fmt1=strip(put(&up1.-
1,2.))||"."||strip(put(&maxdec_var.+&ed1.,10.))||"-r";
        rnd2=10**-(&maxdec_var.+&ed2.);

fmt2=strip(put(&up2.+1,2.))||"."||strip(put(&maxdec_var.+&ed2.,10.))||"-r";
      %end;
    %else %if %upcase(&comp_sep)=COMMA %then %do;
      %let op=%str(, );
        %let cl=;
        rnd1=10**-(&maxdec_var.+&ed1.);

fmt1=strip(put(&up1.,2.))||"."||strip(put(&maxdec_var.+&ed1.,10.))||"-r";
        rnd2=10**-(&maxdec_var.+&ed2.);

fmt2=strip(put(&up2.+1,2.))||"."||strip(put(&maxdec_var.+&ed2.,10.))||"-r";
    %end;
```

```
   if ^missing(&var1.) and missing(&var2.) then
&new_var=putn(round(&var1.,rnd1),fmt1)||"&op."||strip("&ncd2.")||"&cl.";
   else if cmiss(&var1.,&var2.)=2 then &new_var.="&nc_val.";
   else if cmiss(&var1.,&var2.)=0 then
&new_var=putn(round(&var1.,rnd1),fmt1)||"&op."||strip(putn(round(&var2.,rnd2)
,fmt2))||"&cl.";

  %end;
  %* For single statistic values we follow a similar procedure as for
composite statistics values, doing the appopiate chages ;
  %else %do;
    %let up1=&unit_place.;
      %let ed1=&extra_decn.;

      %* Align value for missing or non calculable stats;
      %if %index(%upcase(&NC_DISP.),_BLANK_) %then %let ncd1=;
      %else %let ncd1=&NC_DISP.;

      %let NC_VAL=;
    %let UPL=%eval(&unit_place+1 - %length(&ncd1.));

    %do iii=1 %to &UPL;
      %if &iii < &upl. %then %let NC_VAL=&NC_VAL.%str( );
        %else %let NC_VAL=&NC_VAL.%str( )&ncd1.;
    %end;

      rnd1=10**-(&maxdec_var.+&ed1.);
      fmt1=strip(put(&up1.+(&maxdec_var.>0 or
&ed1.>0)+&maxdec_var.+&ed1.,2.))||"."||strip(put(&maxdec_var.+&ed1.,10.))||"-
r";
      rnd2=.;
      fmt2='';

    if missing(&var.) then &new_var.="&NC_VAL.";
      else &new_var=putn(round(&var.,rnd1),fmt1);
  %end;



%exit: %mend align_stat;
```