PharmaSUG 2024 - Paper DV- 438

Exploring DATALINEPATTERNS, DATACONTRASTCOLORS, DATASYMBOLS, the SAS System® REGISTRY procedure, and Data Attribute Maps (ATTRMAP) to assign invariant attributes to subjects and arms throughout a project.

Kevin R. Viel, Ph.D., Navitas Data Sciences and Histonis, Inc.

ABSTRACT

A subject, arm, or other grouping characteristic can appear in numerous figures, including in various deliveries with updating data, including new subjects or arms. Being able to assign a given subject invariant attributes such as a line pattern, marker, line color, and marker color may assist in the interpretation or review of the figures containing them without burdensome references to a legend. For instance, subjects in the first arm might be assigned blue lines in every plot and USUBJID 1 might be assigned a dashed line with an open diamond symbol. With care, if that subject is enrolled in a new trial, these attributes can remain with her or him, that is, be invariant across trials, or updates (interim and final deliveries). Further, using the SAS System® REGISTRY procedure and testing, one can determine which combinations of attributes might be standards for a company. Creating a data set for the compound or trial, one can assign a unique combination of attributes to a subject and use that data set to create an attribute map data set for the DATTRMAP= option in the Graph Template Language Template. The **goals** of this paper are to demonstrate 1) how to explore the attributes by graphing them, 2) how to explore the color values displayed by the REGISTRY procedure, and 3) how to create such as data set and use in the DATTRMAP= option.

INTRODUCTION

Programmatically creating figures (plots) can have a high learning curve. Often important issues such as range and scale and uniformity thereof across multiple related figures become clouded by appearance issues which can complicate or ease interpretation and inferences of the data conveyed by a figure. Initially appearing remedial, aspects such as line color or pattern and symbols can improve the aesthetics and readability of a figure or frustrate the programmers. While issues like assuring the same range of axes across separate plots by treatment arms might become obvious, assuring that the same arm or subject is represented by the same combination of attributes, like line color or marker symbol, might not be, especially if the programmer relies on the default cycling of attributes used by certain SAS System® procedures. Fortunately, SAS provides the functionality to assign the attributes for a given variable, such as USUBJID or ARM, using a **Discrete Attribute Map** (DATTRMAP) data set. With careful planning, such a data set can assure that the same entity is represented by the same attributes across multiple figures in the same project or even across a submission.

The variables in a DATTRMAP data set include LINECOLOR, MARKERCOLOR, MARKERSYMBOL, and LINEPATTERN. The valid values of these attributes are not necessarily obvious, an issue not confined to the use of a DATTRMAP data set. This paper introduces a SAS macro, **MAC_SG_ DATTRMAP**, that not only creates a DATTRMAP data set, but allows the user to explore the values available, perhaps limited by the operating system, and display and contrast the resulting figures so that the user can decide on the realm of standard values or present them to a manager in a concise, convincing manner.

MAC_SG_ DATTRMAP

Appendix 1 presents the MAC_SG_ DATTRMAP macro. Calling the macro with the macro parameter HELP = Y writes a brief description of the macro to the log, with a list of the macro parameters, a brief description of them, their default values, if applicable, and exits. Briefly, the purpose of the macro displayed is:

```
This statistical graphics (sg) macro:
    1)  Explores the Registry to determine the colors available on the system.
    2)  Displays the Line Patterns, Marker Symbols, and Colors for examination to make
        decisions about which should be selected for attributes in figures.
    3)  Creates a DATTRMAP (discrete attribute map) data set:

              Variable        Example Value
              ID              usubjid_by_arm
              value           001-091-0002
              markercolor     CX0000FF
              markersymbol    CircleFilled
              linecolor       CX0000FF
              linepattern     2

    4)  Creates a Cartesian join of the attribute arrays.  This allows the entire realm of
        distinct combinations attributes to be stored so replication is avoided, whether
        across a project or submission/delivery.
    5)  Displays a test of the DATTRMAP for visual inspection.
```

DISPLAYING ATTRIBUTES

The SAS documentation[1] provides a list of values for attributes and demonstrates them but seeing them in the output that one typically produces may be a better guide. The HELP section of the macro provides more details and citations. Some colors may not be in the Registry in the computing environment of the user. The following creates RTF files demonstrating the attributes shown in **Figure 1**:

```
%mac_sg_dattrmap
   ( display_colornames      = Y
   , colornames_path         = &path.
   , display_linepatterns    = Y
   , linepatterns_path       = &path.
   , display_marker_symbols  = Y
   , marker_symbols_path     = &path.
   ) ;
```

The lack of readability (discernibility) of the contents of Figure 1 emphasizes that restriction of values would be wise, especially when in combination (for instance, light colors with certain line patterns (ThinDot, for example, may not be distinguishable without enlarging the figure, something not available to a printed version. Further, other ways to control the appearance such as SIZE= or THICKNESS= options further enhance (or deteriorate) the ability to discern between entities represented by the combination of attributes. The macro provides the ability to test (display) the results of a DATTRMAP (see section DATTRMAP TEST).

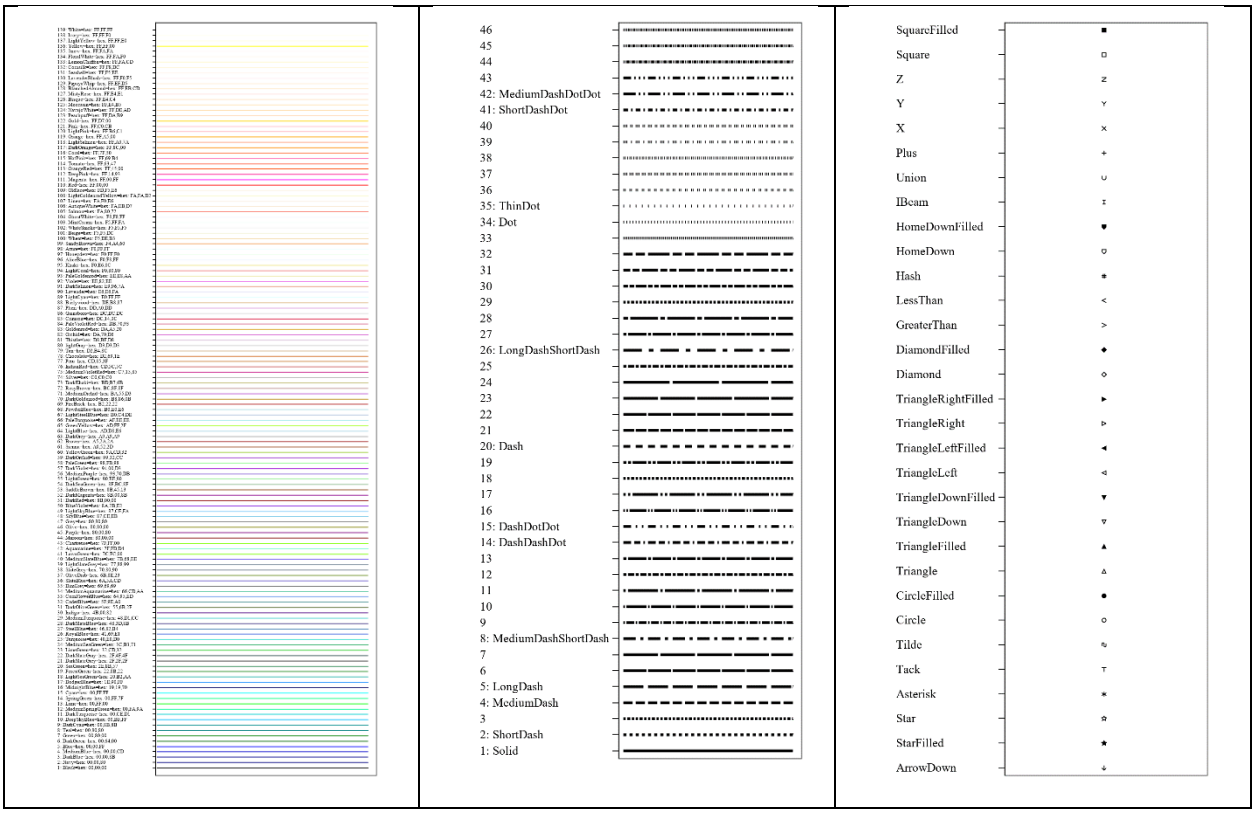

Figure 1. The display of contrast colors (left panel), the line patterns (middle panel), and the marker symbols (right panel).

CREATING A DATTRMAP DATA SET

The macro uses four arrays with default values. The users will want to update the defaults in the (autocall) macro once enterprise-wide conventions or SOP/WI are established. In most cases, the Line Color and Marker Color are expected to match, but this may not be possible in trials or submissions with a large number of subjects. Since it is impossible to predict what attributes might vary, the code to cycle through the arrays is “open”. For example, each arm might be assigned a standard color, line pattern, or marker symbol. Alternatively, each sex or age group might have a distinct marker symbol. Unfortunately, this might also have a learning curve or be a “matter of art”, but that might not concern the programmers in the trenches who just use the correct data set as the argument to the DATTRMAP= option of the appropriate statistical graphics procedure, such as SGPLOT. Finally, the author offers no guidance, but only a warning to select an assignment protocol that might not reveal details such as blinded information or, if it matters, order of enrollment. Consider consulting the biostatisticians for such matters.

Smaller initial value lists of the arrays and using the OUT_DS_ATTRIB_CYCLE option is helpful to demonstrate the cycling:

```
%mac_sg_dattrmap
   ( contrastcolor_list      = %str( "CX0000FF"
                                   , "CXFF0000"
                                   )
   , linestyle_list          = %str(  2
                                   ,  4
                                   )
   , markersymbol_list       = %str( "CircleFilled"
                                   , "TriangleFilled"
                                   , "SquareFilled"
                                   )
   , out_ds_attrib_cycle     = attrib_cycle
   ) ;
```

The results, using the default order of the indices, are in **Table 1**:

Table 1. The attribute cycle data set produced with restricted initial value lists.

linecolor	markercolor	markersymbol	linepattern	__1	__2	__3	__4
CX0000FF	CX0000FF	CircleFilled	2	1	1	1	1
CX0000FF	CX0000FF	CircleFilled	4	1	1	1	2
CX0000FF	CX0000FF	TriangleFilled	2	1	1	2	1
CX0000FF	CX0000FF	TriangleFilled	4	1	1	2	2
CX0000FF	CX0000FF	SquareFilled	2	1	1	3	1
CX0000FF	CX0000FF	SquareFilled	4	1	1	3	2
CXFF0000	CXFF0000	CircleFilled	2	2	1	1	1
CXFF0000	CXFF0000	CircleFilled	4	2	1	1	2
CXFF0000	CXFF0000	TriangleFilled	2	2	1	2	1
CXFF0000	CXFF0000	TriangleFilled	4	2	1	2	2
CXFF0000	CXFF0000	SquareFilled	2	2	1	3	1
CXFF0000	CXFF0000	SquareFilled	4	2	1	3	2

To test (visualize) this as a DATTRMAP data set, we add ID and VALUE variables and use the macro:

```
data attrib_cycle_dattrmap ;
  set attrib_cycle
        ( keep   = linecolor
                   markercolor
                   markersymbol
                   linepattern
        )
      ;
  retain id "test" ;
  value = put( _n_ , z2. ) ;
run ;

%mac_sg_dattrmap
   ( add_attr_names                 = Y
   , test_path                      = &path.
   , test_rtf_name                  = attrib_cycle_plot
   , test_dattrmap                  = attrib_cycle_dattrmap
   , test_id                        = test
   , test_valueattrs_size           = 6pt
   ) ;
```

Note that the option to add the attribute names has been used, providing perhaps more useful names such as “Red” for “CXFF0000” and (MedianDash) for 4 (**Figure 2**).

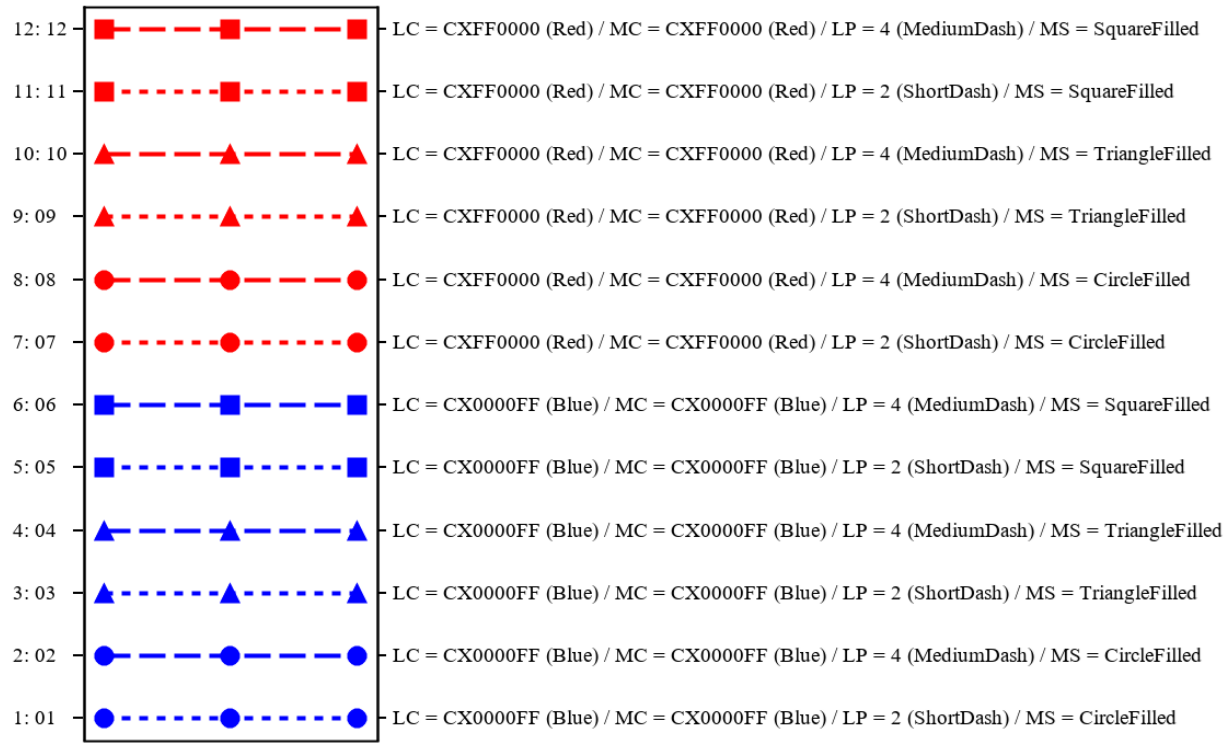

Figure 2. The visualization of a DATTRMAP data set created from the data set in Table 1.

In contrast to a restricted set of attributes, using the defaults:

```
%mac_sg_dattrmap
   ( out_ds_attrib_cycle     = attrib_cycle2 ) ;
```

The results (**Figure 3**) of the test (code not shown, but see the call above) demonstrate the SIZE= or THICKNESS= options (refer to the documentation or the macro) may be essential for discernibility.

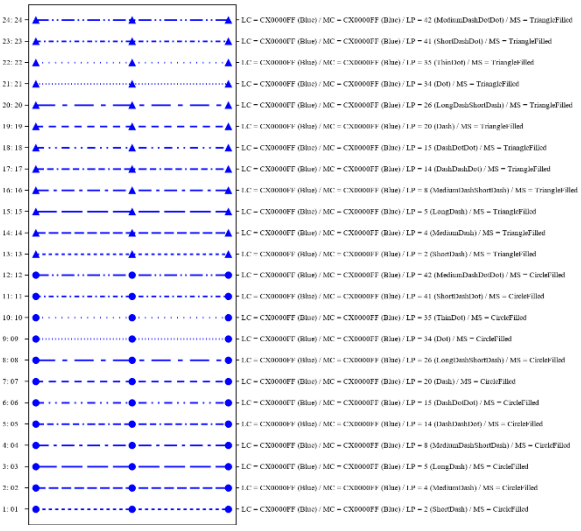

Figure 3. An attribute cycle data set created using the default values of the macro parameters.

Changing the values of the “Index” macro parameter demonstrate how to make another attribute “dominate”:

```
%mac_sg_dattrmap
   ( out_ds_attrib_cycle     = attrib_cycle3
   , first_index             = ls
   , second_index            = ms
   , third_index             = cc
   , fourth_index            = mc
   ) ;
```

The results (**Figure 4**) may be more readily discernible even though they use the same line pattern and only two marker symbols:

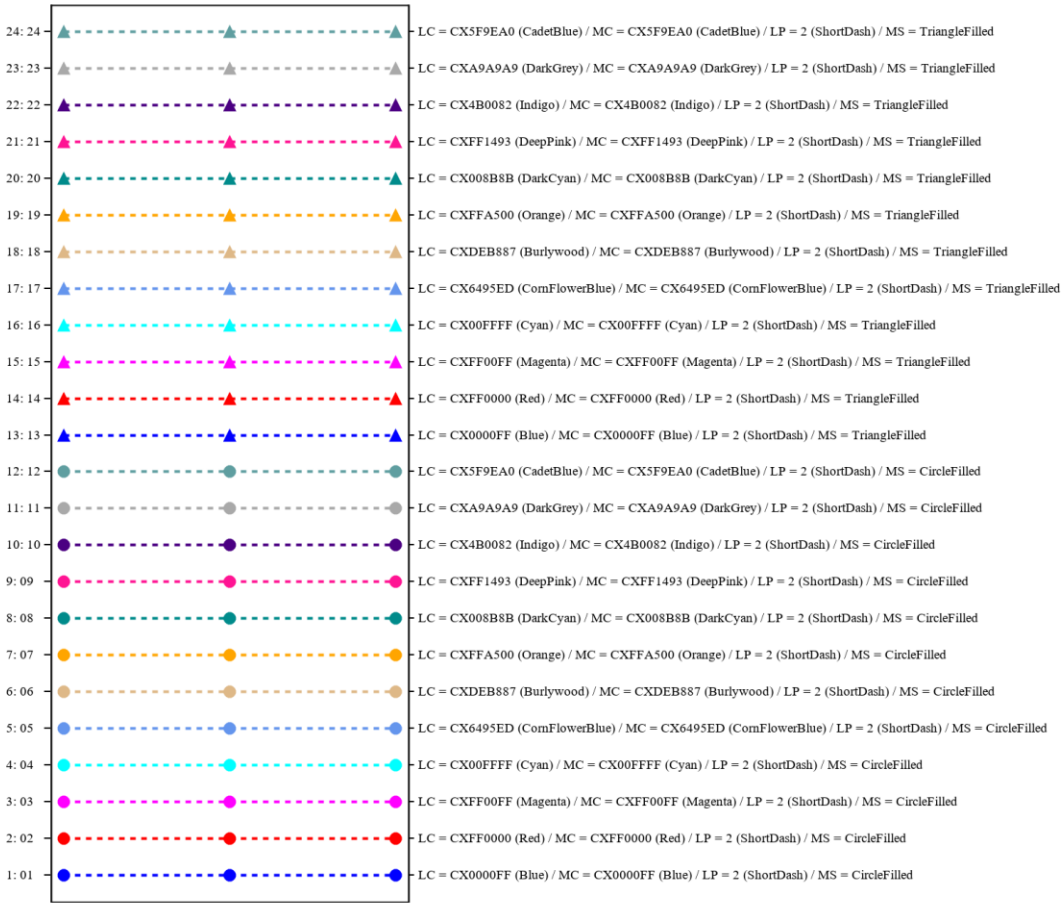

Figure 4. An attribute cycle data set created changing the default values of the “Index” macro parameters.

UPDATING A DATTRMAP DATA SET

Whether using the MAC_SG_DATTRMAP macro or SAS graphing procedures, the order of the observations can change with updates. **Table 2** demonstrates the change in assignments when a new subject is included in updated data (the same code, not shown, was used to call the macro, except for the value of the ID macro parameter).

Table 2. Two DATTRMAP created by the same call to MAC_SG_DATTRMAP, but the second (lower panel) included two new USUBJID's (updated data).

Obs	id	value	arm	linecolor	markercolor	linepattern	markersymbol
1	usubjid	001-001-0004	Arm 1	CX0000FF	CX0000FF	2	CircleFilled
2	usubjid	001-002-0004	Arm 1	CX0000FF	CX0000FF	4	TriangleFilled
3	usubjid	001-002-0007	Arm 1	CX0000FF	CX0000FF	5	SquareFilled
4	usubjid	001-003-0004	Arm 1	CX0000FF	CX0000FF	8	Plus
1	usubjid_updated	001-001-0001	Arm 1	CX0000FF	CX0000FF	2	CircleFilled
2	usubjid_updated	001-001-0004	Arm 1	CX0000FF	CX0000FF	4	TriangleFilled
3	usubjid_updated	001-002-0004	Arm 1	CX0000FF	CX0000FF	5	SquareFilled
4	usubjid_updated	001-002-0007	Arm 1	CX0000FF	CX0000FF	8	Plus

CARTESIAN JOIN

While it might be simple to enumerate pseudo-observations to create a permanent, invariant DATTRMAP data set, the macro provides the ability to create a cartesian join of the values of the four arrays:

```
%mac_sg_dattrmap
   ( cartesian_join                  = Y
   , cartesian_join_ds               = cart
   ) ;
```

The code above uses the defaults including the initial value list of the arrays and the Index variables. This results in 12 X 12 X 12 X 12 = 20,736 distinct combinations, which, by numbers typically suffices for the vast majority of clinical trials. Selection of the default values for the initial value lists (the macro parameters CONTRASTCOLOR_LIST, MARKERCOLOR_LIST, LINESTYLE_LIST, and MARKERSYMBOL_LIST) is vitally important. Further, some combinations might want to be excluded, for instance, requiring that the line and marker colors match, which will reduce the number of available combinations. **Table 3** shows a partial listing of the results.

The following snippet form the log demonstrates that the observations represent distinct combinations of these for attributes:

```
1924  proc sort
1925     data   = cart
1926     out    = cart_out
1927     dupout = cart_dupout
1928     nodupkey
1929     ;
1930    by linecolor
1931       markercolor
1932       markersymbol
1933       linepattern
1934       ;
1935  run ;

NOTE: There were 20736 observations read from the data set WORK.CART.
NOTE: 0 observations with duplicate key values were deleted.
NOTE: The data set WORK.CART_OUT has 20736 observations and 8 variables.
NOTE: The data set WORK.CART_DUPOUT has 0 observations and 8 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds
```

Table 3. Partial results of a cartesian join using the default macro parameter values.

linecolor	markercolor	markersymbol	linepattern	linecolor_order	markercolor_order	markersymbol_order	linepattern_order
CX0000FF	CX0000FF	CircleFilled	2	1	1	1	1
CX0000FF	CX0000FF	CircleFilled	4	1	1	1	2
CX0000FF	CX0000FF	CircleFilled	5	1	1	1	3
CX0000FF	CX0000FF	CircleFilled	8	1	1	1	4
CX0000FF	CX0000FF	CircleFilled	14	1	1	1	5
CX0000FF	CX0000FF	CircleFilled	15	1	1	1	6
CX0000FF	CX0000FF	CircleFilled	20	1	1	1	7
CX0000FF	CX0000FF	CircleFilled	26	1	1	1	8
CX0000FF	CX0000FF	CircleFilled	34	1	1	1	9
CX0000FF	CX0000FF	CircleFilled	35	1	1	1	10
CX0000FF	CX0000FF	CircleFilled	41	1	1	1	11
CX0000FF	CX0000FF	CircleFilled	42	1	1	1	12
CX0000FF	CX0000FF	TriangleFilled	2	1	1	2	1
CX0000FF	CX0000FF	TriangleFilled	4	1	1	2	2
CX0000FF	CX0000FF	TriangleFilled	5	1	1	2	3
CX0000FF	CX0000FF	TriangleFilled	8	1	1	2	4
CX0000FF	CX0000FF	TriangleFilled	14	1	1	2	5
CX0000FF	CX0000FF	TriangleFilled	15	1	1	2	6
CX0000FF	CX0000FF	TriangleFilled	20	1	1	2	7
CX0000FF	CX0000FF	TriangleFilled	26	1	1	2	8
CX0000FF	CX0000FF	TriangleFilled	34	1	1	2	9
CX0000FF	CX0000FF	TriangleFilled	35	1	1	2	10
CX0000FF	CX0000FF	TriangleFilled	41	1	1	2	11
CX0000FF	CX0000FF	TriangleFilled	42	1	1	2	12

CAPTURING THE SAS CODE GENERATED BY THE MACRO OR DEBUGGING

The macro is not complex, but it has multiple purposes (parts). Inadvertently generating an error may be easy. Even so, some regulatory agencies may not want externally defined SAS macros called in programs. The macro MAC_DEBUG_MFILE[2] is useful for this purpose:

```
%mac_debug_mfile
   ( mcall = mac_sg_dattrmap
               ( contrastcolor_list      = %str(   "CX0000FF"
                                                 , "CXFF0000"
                                               )
               , linestyle_list          = %str(    2
                                                  ,  4
                                               )
               , markersymbol_list       = %str(   "CircleFilled"
                                                 , "TriangleFilled"
                                                 , "SquareFilled"
                                               )

               , out_ds_dattrmap         = attrmap_usubjid
               , id                      = usubjid
               , debug                   = Y
               )
   ) ;
```

If run interactively in Windows, then the macro copies the generated SAS code to the clipboard. Typing Ctrl-V in a text editor will produce the following (unformatted) code:

```
data attrmap_usubjid ;
length id $ 32 value $ 50 linecolor $ 8 markercolor $ 8 linepattern 8 markersymbol $ 20 ;
array cc ( 2 ) $ 8 _temporary_ ( "CX0000FF" , "CXFF0000" ) ;
array mc ( 2 ) $ 8 _temporary_ ( "CX0000FF" , "CXFF0000" ) ;
array ls ( 2 ) _temporary_ ( 2 , 4 ) ;
array ms ( 3 ) $ 20 _temporary_ ( "CircleFilled" , "TriangleFilled" , "SquareFilled" ) ;
if _n_ = 1 then do ;
dim_cc = dim( cc ) ;
dim_mc = dim( mc ) ;
dim_ls = dim( ls ) ;
dim_ms = dim( ms ) ;
end ;
retain dim_cc dim_mc dim_ls dim_ms cc_index 0 mc_index 0 ls_index 0 ms_index 0 id
"usubjid" ;
set dattrmap ;
by studyid arm ;
if first.arm then do ;
cc_index + 1 ;
mc_index + 1 ;
ls_index = 1 ;
ms_index = 0 ;
end ;
ms_index + 1 ;
markercolor = cc( mc_index ) ;
linecolor = cc( cc_index ) ;
linepattern = ls( ls_index ) ;
markersymbol = ms( ms_index ) ;
if last.arm and cc_index = dim_cc then cc_index = 1 ;
if last.arm and mc_index = dim_mc then mc_index = 1 ;
if ms_index = dim_ms then do ;
ls_index + 1 ;
if ls_index > dim_ls then ls_index = 1 ;
ms_index = 0 ;
end ;
run ;
```

This may allow the user to experiment easier or provide the code if, for some reason, the macro appears in production code subject to submission to a regulatory agency that prohibits externally defined macros to be called in a submission.

CONCLUSION

This paper contributes to the ability of programmers to learn about and decide upon the values of the attributes that enhance the readability of figures and the discernability between the combinations of attributes that represent an entity or characteristic, such as a USUBJID or ARM. The macro MAC_SG_DATTRMAP allows the user create Discrete Attribute Map (DATTRMAP) data sets with the ability to assign to an entity and combinations of attributes (line color, marker color, line pattern, and marker symbol) that can be invariant between figures within a project or within a submission. The code of the macro might even provide a basis for creating figures or customizing aspects of them such as the SIZE= options to VALUEATTRS or MARKERATTRS options or the THICKNESS= to the LINEATTRS option. Importantly, by visualizing the attributes or combinations of them in output that the user is likely to create offers the user the opportunity to select attributes or combinations of them that provide the greatest enhancement to the appearance and interpretations of the figures.

REFERENCES

[1] SAS Institute Inc. 2016. SAS(R) 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc.

[2] Viel, K. 2016. "Capturing Macro Code when Debugging in the Windows Environment: The Power of MFILE and the Simplicity of Pasting." Proceedings of the PharmaSUG 2016 Conference, Denver, CO: PharmaSUG. https://www.pharmasug.org/proceedings/2016/TT/PharmaSUG-2016-TT17.pdf

CONTACT INFORMATION

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

Kevin R. Viel, Ph.D.
Navitas Data Sciences
kevin.viel@navitaslifesciences.com
www.navitasdatasciences.com

Histonis, Incorporated
kviel@histonis.org

Appendix 1. The macro MAC_SG_DATTRMAP.

```
%macro mac_sg_dattrmap
   ( in_ds                              = dattrmap
   , by                                 = studyid
                                          arm
   , out_ds_dattrmap                    = %str()
   , cartesian_join                     = N
   , cartesian_join_ds                  = %str()
   /***/
   , out_ds_attrib_cycle                = %str()
   , first_index                        = cc
   , second_index                       = mc
   , third_index                        = ms
   , fourth_index                       = ls
   , contrastcolor_list                 = %str( "CX0000FF"
                                              , "CXFF0000"
                                              , "CXFF00FF"
                                              , "CX00FFFF"
                                              , "CX6495ED"
                                              , "CXDEB887"
                                              , "CXFFA500"
                                              , "CX008B8B"
                                              , "CXFF1493"
                                              , "CX4B0082"
                                              , "CXA9A9A9"
                                              , "CX5F9EA0"
                                              )
   , markercolor_list                   = %str()
   , linestyle_list                     = %str(  2
                                              ,  4
                                              ,  5
                                              ,  8
                                              , 14
                                              , 15
                                              , 20
                                              , 26
                                              , 34
                                              , 35
                                              , 41
                                              , 42
                                              )
   , markersymbol_list                  = %str( "CircleFilled"
                                              , "TriangleFilled"
                                              , "SquareFilled"
                                              , "Plus"
                                              , "x"
                                              , "Asterisk"
                                              , "Circle"
                                              , "Triangle"
                                              , "Diamond"
                                              , "Square"
                                              , "StarFilled"
                                              , "Star"
                                              )
   , id                                 =
   , length_id                          = 32
   , length_value                       = $ 50
   , length_linepattern                 = 8

   , index_set_wrapper_top              = if first.arm then do ;
   , cc_index_set                       = cc_index + 1 ;
   , mc_index_set                       = mc_index + 1 ;
   , ls_index_set                       = ls_index = 1 ;
   , ms_index_set                       = ms_index = 0 ;
   , index_set_wrapper_bottom           = end ;

   , index_increment_wrapper_top        = %str()
   , cc_index_increment                 = %str()
   , mc_index_increment                 = %str()
   , ls_index_increment                 = %str()
   , ms_index_increment                 = ms_index + 1 ;
   , index_increment_wrapper_bottom     = %str()

   , index_reset_wrapper_top            = %str()
   , cc_index_reset                     = if last.arm and cc_index = dim_cc then cc_index = 1 ;
   , mc_index_reset                     = if last.arm and mc_index = dim_mc then mc_index = 1 ;
   , ls_index_reset                     = %str()
   , ms_index_reset                     = if ms_index = dim_ms
                                            then
                                              do ;
                                                 ls_index + 1 ;
                                                 if ls_index > dim_ls then ls_index = 1 ;
                                                 ms_index = 0 ;
                                              end ;
   , index_reset_wrapper_bottom         = %str()
   /****** Visualize attributes *****/
   , add_attr_names                     = N

   , display_colornames                 = N
   , colornames_path                    = %str()
   , colornames_ds                      = sas_colornames_registry
   , colornames_plot_ds                 = %str()
   , colornames_delete                  = Y
   /* LINE PATTERNS */
   , display_linepatterns               = N
   , linepatterns_path                  = %str()
   , linepatterns_ds                    = linepatterns
   , linepatterns_delete                = Y
   /* MARKER SYMBOLS */
   , display_marker_symbols             = N
   , marker_symbols_path                = %str()
   , marker_symbols_ds                  = marker_symbols
   , marker_symbols_delete              = Y

   /****** Visualize test *****/
   , test_path                          = %str()
   , test_rtf_name                      = test
   , test_obs_per_page                  = %str()
   , test_dattrmap                      = %str()
   , test_id                            = usubjid
   , test_ods_graphics_width            = 10in
   , test_sgplot_aspect                 = 2.5
   , test_valueattrs_size               = 8pt
   , test_lineattrs_thickness           = 2
   , test_markerattrs_size              = 8

   , debug                              = N
   , help                               = N
   ) ;

%if &help. = Y
%then
   %do ;
       %let mprint_orig = %sysfunc(getoption(mprint)) ;
       options nomprint ;

       skip ;
       skip ;
       %put                                                                                                                          ;
       %put Purpose of program:             This statistical graphics (sg) macro:                                                    ;
       %put %str(                                     )1) Explores the Registry to determine the colors available on the system.     ;
       %put %str(                                     )2) Displays the Line Patterns, Marker Symbols, and Colors for examination to make decisions about which ;
       %put %str(                                        )should be selected for attributes in figures.                              ;
       %put %str(                                     )3) Creates a DATTRMAP (discrete attribute map) data set:                      ;
       skip ;
       %put %str(                                           )Variable        Example Value                                           ;
       %put %str(                                           )ID              usubjid_by_arm                                          ;
       %put %str(                                           )value           001-091-0002                                            ;
       %put %str(                                           )markercolor     CX0000FF                                                ;
       %put %str(                                           )markersymbol    CircleFilled                                            ;
       %put %str(                                           )linecolor       CX0000FF                                                ;
       %put %str(                                           )linepattern     2                                                       ;
       skip ;
       %put %str(                                     )4) Creates a Cartesian join of the attribute arrays.  This allows the entire realm of distinct ;
       %put %str(                                        )combinations attributes to be stored so replication is avoided, whether across a project or ;
       %put %str(                                        )submission/delivery.                                                       ;
       skip ;
       %put %str(                                     )5) Displays a test of the DATTRMAP for visual inspection.                     ;
       skip ;
       %put Please review:                                                                                                           ;
       %put SAS Institute Inc. 2016. SAS(R) 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc.          ;
       %put January 31, 2024                                                                                                         ;
       skip ;
       %put Part 3 The Procedures                                                                                                    ;
       %put Chapter 14 Commonly Used Attribute Options Line                                                                          ;
       skip ;
       %put Table 14.1 Commonly Used Line Patterns page 1651                                                                         ;
       %put Pattern Number      Pattern Name                                                                                         ;
       %put %str(      )1             Solid                                                                                          ;
```

```
      %put %str(      )2          ShortDash                                                    ;
      %put %str(      )4          MediumDash                                                   ;
      %put %str(      )5          LongDash                                                     ;
      %put %str(      )8          MediumDashShortDash                                          ;
      %put %str(     )14          DashDashDot                                                  ;
      %put %str(     )15          DashDotDot                                                   ;
      %put %str(     )20          Dash                                                         ;
      %put %str(     )26          LongDashShortDash                                            ;
      %put %str(     )34          Dot                                                          ;
      %put %str(     )35          ThinDot                                                      ;
      %put %str(     )41          ShortDashDot                                                 ;
      %put %str(     )42          MediumDashDotDot                                             ;
      skip ;
      %put SAS Color Names and RGB Values in the SAS Registry page 1667                        ;
      %put RGB Value       SAS Color Names                 Hex                                 ;
      %put CX000000        Black                           00,00,00                            ;
      %put CX0000FF        Blue                            00,00,FF                            ;
      %put CXFF0000        Red                             FF,00,00                            ;
      %put CXFF00FF        Magenta                         FF,00,FF                            ;
      %put CX00FFFF        Aqua                            00,FF,FF                            ;
      %put CX6495ED        CornFlowerBlue                  64,95,ED                            ;
      %put CXDEB887        Burlywood                       DE,B8,87                            ;
      %put CXFFA500        Orange                          FF,A5,00                            ;
      %put CX008B8B        DarkCyan                        00,8B,8B                            ;
      %put CXFF1493        DeepPink                        FF,14,93                            ;
      %put CX4B0082        Indigo                          4B,00,82                            ;
      skip ;

      %put Marker Attributes and Symbols page 1657                                             ;
      %put Table 14.5 Supported Marker Symbols                                                 ;
      skip ;

      %put Macro Parameter                       Description                                                                                         ;
      %put _________________                     ________________________________________________________________________________________________  ;
      %put in ds                               = The input data set.                                                                                 ;
      %put %str(                                        )Default: dattrmap                                                                           ;
      %put by                                  = The BY variables.                                                                                   ;
      %put %str(                                        )Default: studyid                                                                            ;
      %put %str(                                                 )arm                                                                                ;
      %put out ds dattrmap                     = The name of the output DATTRMAP data set.  This data may be used as the argument to the DATTRMAP= option.   ;
      %put cartesian_join                      = Whether to create the cartesian join of the attribute arrays.                                       ;
      %put %str(                                        )Default: N                                                                                  ;
      %put cartesian_join_ds                   = The name of the output data set of the cartesian join.                                              ;
      %put %str(                                        )Default: %nrstr(%%)str()  (Not produced)                                                    ;
      skip ;
      %put out ds attrib cycle                 = The name of the output data sets.  The attributes are held in four arrays.  How to cycle through them, that is   ;
      %put %str(                                      )that is combine them, can be complicated and this data set displays their combination and the indices that ;
      %put %str(                                      )resulted in them.                                                                     ;
      %put %str(                                        )Default: %nrstr(%%)str()  (Not produced)                                                    ;
      %put first index                         = The first attribute through which to cycle. The outermost DO-LOOP.                                  ;
      %put %str(                                        )Default: cc                                                                                 ;
      %put second_index                        = The second attribute through which to cycle. The second DO-LOOP, nested in the first.               ;
      %put %str(                                        )Default: mc                                                                                 ;
      %put third_index                         = The third attribute through which to cycle. The third DO-LOOP, nested in the second.                ;
      %put %str(                                        )Default: ls                                                                                 ;
      %put fourth_index                        = The fourth attribute through which to cycle. The innermost DO-LOOP, nested in the third.            ;
      %put %str(                                        )Default: ms                                                                                 ;
      skip ;
      %put contrastcolor_list                  = The initial-value-list of the array CC, the line colors through which to cycle.                     ;
      %put %str(                                        )Default: %nrstr(%%)str(   "CX0000FF"                                                        ;
      %put %str(                                                       ), "CXFF0000"                                                                 ;
      %put %str(                                                       ), "CXFF00FF"                                                                 ;
      %put %str(                                                       ), "CX00FFFF"                                                                 ;
      %put %str(                                                       ), "CX6495ED"                                                                 ;
      %put %str(                                                       ), "CXDEB887"                                                                 ;
      %put %str(                                                       ), "CXFFA500"                                                                 ;
      %put %str(                                                       ), "CX008B8B"                                                                 ;
      %put %str(                                                       ), "CXFF1493"                                                                 ;
      %put %str(                                                       ), "CX4B0082"                                                                 ;
      %put %str(                                                       ), "CXA9A9A9"                                                                 ;
      %put %str(                                                       ), "CX5F9EA0"                                                                 ;
      %put %str(                                                     ))                                                                              ;
      %put markercolor_list                    = The initial-value-list of the array MC, the markers colors through which to cycle. If this value is missing,   ;
      %put %str(                                      )then it is assigned CONTRASTCOLOR LIST, which makes the line and marker color match.  ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put linestyle_list                      = The initial-value-list of the array LS, the list of line patterns through which to cycle.           ;
      %put %str(                                        )Default: %nrstr(%%)str(    2                                                                ;
      %put %str(                                                       ),  4                                                                         ;
      %put %str(                                                       ),  5                                                                         ;
      %put %str(                                                       ),  8                                                                         ;
      %put %str(                                                       ), 14                                                                         ;
      %put %str(                                                       ), 15                                                                         ;
      %put %str(                                                       ), 20                                                                         ;
      %put %str(                                                       ), 26                                                                         ;
      %put %str(                                                       ), 34                                                                         ;
      %put %str(                                                       ), 35                                                                         ;
      %put %str(                                                       ), 41                                                                         ;
      %put %str(                                                       ), 42                                                                         ;
      %put %str(                                                     ))                                                                              ;
      %put markersymbol_list                   = The initial-value-list of the array MS, the list of marker symbols through which to cycle.          ;
      %put %str(                                        )Default: %nrstr(%%)str(   "CircleFilled"                                                    ;
      %put %str(                                                       ), "TriangleFilled"                                                           ;
      %put %str(                                                       ), "SquareFilled"                                                             ;
      %put %str(                                                       ), "Plus"                                                                     ;
      %put %str(                                                       ), "x"                                                                        ;
      %put %str(                                                       ), "Asterisk"                                                                 ;
      %put %str(                                                       ), "Circle"                                                                   ;
      %put %str(                                                       ), "Triangle"                                                                 ;
      %put %str(                                                       ), "Diamond"                                                                  ;
      %put %str(                                                       ), "Square"                                                                   ;
      %put %str(                                                       ), "StarFilled"                                                               ;
      %put %str(                                                       ), "Star"                                                                     ;
      %put %str(                                                     ))                                                                              ;
      skip ;
      %put id                                  = The name of the ID to be used in the SG Procedures.                                                 ;
      %put length_id                           = The length of the ID variable.                                                                      ;
      %put %str(                                        )Default: 32                                                                                 ;
      %put length value                        = The length of the VALUE variable.                                                                   ;
      %put %str(                                      )Be sure to select an adequate length for updates that might result in a longer value.  ;
      %put %str(                                        )Default: $ 10                                                                               ;
      %put length_linepattern                  = The length of the LINEPATTERN variable.                                                             ;
      %put %str(                                        )Default: 8                                                                                  ;
      skip ;
      %put In the data step that creates OUT_DS_DATTRMAP, the following macro parameters appear, as they are below, after the BY statement and before the   ;
      %put first attribute assignment statement:                                                                                                       ;
      skip ;
      %put %str(   )index_set_wrapper_top                                                                                                              ;
      %put %str(   )cc index set                                                                                                                       ;
      %put %str(   )mc index set                                                                                                                       ;
      %put %str(   )ls_index_set                                                                                                                       ;
      %put %str(   )ms_index_set                                                                                                                       ;
      %put %str(   )index_set_wrapper_bottom                                                                                                           ;
      skip ;
      %put %str(   )index_increment_wrapper_top                                                                                                        ;
      %put %str(   )cc_index_increment                                                                                                                 ;
      %put %str(   )mc_index_increment                                                                                                                 ;
      %put %str(   )ls_index_increment                                                                                                                 ;
      %put %str(   )ms index increment                                                                                                                 ;
      %put %str(   )index_increment_wrapper_bottom                                                                                                     ;
      skip ;
      %put index_set_wrapper_top               = Default: if first.arm then do %str(;)                                                               ;
      %put cc index set                        = Default: cc index + 1 %str(;)                                                                       ;
      %put mc index set                        = Default: mc index + 1 %str(;)                                                                       ;
      %put ls_index_set                        = Default: ls_index = 1 %str(;)                                                                       ;
      %put ms_index_set                        = Default: ms_index = 0 %str(;)                                                                       ;
      %put index_set_wrapper_bottom            = Default: end %str(;)                                                                                ;
      skip ;
      %put index_increment_wrapper_top         = Default: %nrstr(%%)str()                                                                            ;
      %put cc_index_increment                  = Default: %nrstr(%%)str()                                                                            ;
      %put mc_index_increment                  = Default: %nrstr(%%)str()                                                                            ;
      %put ls_index_increment                  = Default: %nrstr(%%)str()                                                                            ;
      %put ms_index_increment                  = ms_index + 1 %str(;)                                                                                ;
      %put index_increment_wrapper_bottom      = Default: %nrstr(%%)str()                                                                            ;
      skip ;
      %put In the data step that creates OUT_DS_DATTRMAP, the following macro parameters appear, as they are below, after the last attribute assignment   ;
      %put and before the RUN statement:                                                                                                               ;
      skip ;
      %put %str(   )index_reset_wrapper_top                                                                                                            ;
      %put %str(   )cc_index_reset                                                                                                                     ;
      %put %str(   )mc_index_reset                                                                                                                     ;
      %put %str(   )ls_index_reset                                                                                                                     ;
      %put %str(   )ms_index_reset                                                                                                                     ;
      %put %str(   )index_reset_wrapper_bottom                                                                                                         ;
      skip ;
      %put index_reset_wrapper_top             = Default: %nrstr(%%)str()                                                                            ;
      %put cc_index_reset                      = Default: if last.arm and cc_index = dim_cc then cc_index = 1 %str(;)                                ;
      %put mc index reset                      = Default: if last.arm and mc index = dim mc then mc index = 1 %str(;)                                ;
      %put ls_index_reset                      = Default: Default: %nrstr(%%)str()                                                                   ;
      %put ms_index_reset                      = Default: if ms_index = dim_ms                                                                       ;
      %put %str(                                        )then                                                                                        ;
      %put %str(                                          )do %str(;)                                                                                ;
      %put %str(                                             )ls_index + 1 %str(;)                                                                   ;
      %put %str(                                             )if ls index > dim ls then ls index = 1 %str(;)                                         ;
      %put %str(                                             )ms_index = 0 %str(;)                                                                   ;
      %put %str(                                             )end %str(;)                                                                            ;
      %put index_reset_wrapper_bottom          = Default: %nrstr(%%)str()                                                                            ;
      skip ;
      /****** Visualize attributes *****/
      %put add_attr_names                           = Whether to add the attribute names, for example CX000000 (Black) or 2 (Solid), to the data sets and, thus,  ;
      %put %str(                                          )demonstration plots.                                                                      ;
      %put %str(                                        )Default: Y                                                                                  ;

      /* COLORS */
      %put display_colornames                  = Whether to display the colors in the SAS Color Registry.  This may allow a user to select optimal colors from   ;
      %put %str(                                      )those available.  The source is the log from a REGISTRY procedure with the options LIST and  ;
      %put %str(                                      )STARTAT = "COLORNAMES".                                                               ;
      %put %str(                                        )Default: N                                                                                  ;
      %put colornames_path                     = The path for the colornames plot.  If null, then this is assigned the path of the option SASUSER.   ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put colornames_ds                       = The name of the data set holding the colornames registry values.                                    ;
      %put %str(                                        )Default: sas colornames registry                                                            ;
      %put colornames_plot_ds                  = The name of the colornames data set of the registry values to be plotted as horizontal solid lines and a   ;
      %put %str(                                      )circlefilled marker. If null, then it is assigned the value in colornames_ds. This is also the name of   ;
      %put %str(                                      )the RTF file to which the plot is written.                                            ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put colornames_delete                   = Whether to delete COLORNAMES_DS and COLORNAMES_PLOT_DS.                                             ;
      %put %str(                                        )Default: Y                                                                                  ;
      /* LINE PATTERNS */
      %put display_linepatterns                = Whether to display the line patterns.  This may allow a user to select optimal line patterns.       ;
      %put %str(                                        )Default: N                                                                                  ;
      %put linepatterns path                   = The path for the linepatterns plot.  If null, then this is assigned the path of the option SASUSER.  ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put linepatterns_ds                     = The name of the data set holding the linepatterns.                                                  ;
      %put %str(                                        )Default: linepatterns                                                                       ;
      %put linepatterns_delete                 = Whether to delete LINEPATTERNS_DS.                                                                  ;
      %put %str(                                        )Default: Y                                                                                  ;
      /* MARKER SYMBOLS */
      %put display_marker_symbols              = Whether to display the marker symbols.  This may allow a user to select optimal marker symbols.     ;
      %put %str(                                        )Default: N                                                                                  ;
      %put marker_symbols_path                 = The path for the marker_symbols plot.  If null, then this is assigned the path of the option SASUSER.   ;
      %put %str(                                      )will be deleted at the end of the SAS session.                                        ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put marker_symbols_ds                   = The name of the data set holding the marker_symbols.                                                ;
      %put %str(                                        )Default: marker_symbols                                                                     ;
      %put marker_symbols_delete               = Whether to delete MARKER_SYMBOLS_DS.                                                                ;
      %put %str(                                        )Default: Y                                                                                  ;
      skip ;
      /************/
      %put test_path                         = The path for the test of the DATTRMAP plot.  If null, then this is assigned the path of the option SASUSER.   ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put test_rtf_name                     = The name of the RTF file of the plot of the test of DATTRMAP.                                         ;
      %put %str(                                        )Default: test                                                                               ;
      %put test_obs_per_page                 = The number of observations to plot per physical page.                                                 ;
      %put %str(                                        )Default: %nrstr(%%)str()                                                                    ;
      %put test_dattrmap                     = The DATTRMAP data set to test.                                                                        ;
      %put test ods graphics width           = The value of WIDTH option to the ODS GRAPHICS statement.                                              ;
      %put %str(                                        )Default: 10in                                                                               ;
      %put test_sgplot_aspect                = The value of the ASPECT= option to the SGPLOT statement.                                              ;
      %put %str(                                        )Default: 2.5                                                                                ;
      %put test_valueattrs_size              = The font size of the Y and Y2 axis labels.                                                            ;
      %put %str(                                        )Default: 8pt                                                                                ;
      skip ;
      %put debug                               = Whether to KEEP all variables in the OUT_DS_DATTRMAP dataset for debugging purposes.                ;
      %put %str(                                        )Default: N                                                                                  ;
      skip ;
      %put                                                                                                                                             ;
      %put End of help                                                                                                                                 ;

      options &mprint_orig. ;

      %goto __END ;
    %end ;


%if     &out_ds_dattrmap. ne %str()
    and &cartesian_join.  =  Y
%then
   %do ;
      %put E%str(RROR: ) Either OUT_DS_DATTRMAP should be null or CARTESIAN_JOIN cannot be Y. ;
      %put E%str(RROR: ) OUT_DS_DATTRMAP = &out_ds_dattrmap. ;
      %put E%str(RROR: ) CARTESIAN_JOIN  = cannot be &cartesian_join. ;
      %goto __END ;
   %end ;

/************************************************/
%if &add_attr_names. = Y
%then
   %do ;
      data commonly_used_line_patterns ;

         length linepattern_name $ 30 ;

         linepattern       = 1 ;
         linepattern_name  = " Solid" ;
         output ;
         linepattern       = 2 ;
         linepattern_name  = " ShortDash" ;
         output ;
```

```
            linepattern       = 4 ;
            linepattern_name  = " MediumDash" ;
            output ;
            linepattern       = 5 ;
            linepattern_name  = " LongDash" ;
            output ;
            linepattern       = 8 ;
            linepattern_name  = " MediumDashShortDash" ;
            output ;
            linepattern       = 14 ;
            linepattern_name  = "DashDashDot" ;
            output ;
            linepattern       = 15 ;
            linepattern_name  = "DashDotDot" ;
            output ;
            linepattern       = 20 ;
            linepattern_name  = "Dash" ;
            output ;
            linepattern       = 26 ;
            linepattern_name  = "LongDashShortDash" ;
            output ;
            linepattern       = 34 ;
            linepattern_name  = "Dot" ;
            output ;
            linepattern       = 35 ;
            linepattern_name  = "ThinDot" ;
            output ;
            linepattern       = 41 ;
            linepattern_name  = "ShortDashDot" ;
            output ;
            linepattern       = 42 ;
            linepattern_name  = "MediumDashDotDot" ;
            output ;

          run ;

          /***/
          filename temp
                   temp
                   ;

          /* Capture the set of color names and RGB values from the SAS Registry that you can use to specify colors. */
          proc printto
             log = temp
             new
             ;
          run ;

          proc registry
             list
             startat = "COLORNAMES"
             ;
          run ;

          proc printto ;
          run ;

          data colornames ;

            length registry    $ 100
                   color       $  50
                   hex_color   $   8
                   ;

            infile temp ;
            input ;
            put _infile_ ;

            if prxmatch( "/=hex:/i" , _infile_ )
            then
              do ;
                 registry    = strip( _infile_ ) ;
                 color       = scan( registry , 1 , "=" ) ;
                 hex_color   = cats( "CX"
                                   , compress( scan( registry , 2 , ":" ) , ", " )
                                   ) ;
                 output ;
              end ;

          run ;

          filename temp ;

          proc sort
             data = colornames ;
            by hex_color
               color
               ;
          run ;

          data colornames ;
            set colornames ;
            by hex_color
               color
               ;
            if     first.hex_color = 0
               or  last.hex_color  = 0
            then put "A" "LERT: "
                      hex_color=
                      color=
                      @50 registry=
                      ;
            if     first.hex_color = 0
               and last.hex_color  = 1
            then put ;

            if last.hex_color = 0 then delete ;

            group + 1 ;

          run ;

          %if &display_colornames. = Y
          %then
             %do ;
                 data &colornames_ds. ;
                   set colornames ;
                 run ;
              %end ;

      %end ; /* END OF add_attr_names = Y */


    /*************************************************/
    %if &display_colornames. = Y
    %then
       %do ;
          %if &colornames_path. = %str()
          %then %let colornames_path = %sysfunc( getoption( sasuser )) ;

          %if &colornames_plot_ds. = %str()
          %then %let colornames_plot_ds = &colornames_ds. ;

          /*******/
          %if &add_attr_names. =  N
          %then
             %do ;
                 filename temp
                          temp
                          ;

                 /* Capture the set of color names and RGB values from the SAS Registry that you can use to specify colors. */
```

```
                proc printto
                   log = temp
                   new
                   ;
                run ;

                proc registry
                   list
                   startat = "COLORNAMES"
                   ;
                run ;

                proc printto ;
                run ;

                data &colornames_ds. ;

                  length registry    $ 100
                         color       $  50
                         hex_color   $   8
                         ;

                  infile temp ;
                  input ;
                  put _infile_ ;

                  if prxmatch( "/=hex:/i" , _infile_ )
                  then
                    do ;
                       registry    = strip(  infile  ) ;
                       color       = scan( registry , 1 , "=" ) ;
                       hex_color   = cats( "CX"
                                         , compress( scan( registry , 2 , ":" ) , ", " )
                                         ) ;
                       output ;
                    end ;

                run ;

                filename temp ;

                proc sort
                   data = &colornames_ds. ;
                  by hex_color
                     color
                     ;
                run ;

                data &colornames_ds. ;
                  set &colornames_ds. ;
                  by hex color
                     color
                     ;
                  if     first.hex color = 0
                     or  last.hex_color  = 0
                  then put "A" "LERT: "
                           hex_color=
                           color=
                           @50 registry=
                           ;
                  if     first.hex_color = 0
                     and last.hex_color  = 1
                  then put ;

                  if last.hex_color = 0 then delete ;

                  group + 1 ;

                run ;

             %end ; /* END OF else: add_attr_names ne Y */

           /* Create the data set to plot */
           data &colornames_plot_ds.
                  ( drop   = index )
               ;

             length index  $ 3 ;

             set &colornames_ds. ;

             index = put( group , 8.0 -L ) ;
             index = right( index ) ;

             registry = cat( index
                           , ": "
                           , strip( registry )
                           ) ;

             y + 1 ;
             x = 0 ;
             output ;
             x = 1 ;
             output ;
             x = 2 ;
             output ;

           run ;

           /*****************************************/
           options orientation  = portrait
                   topmargin    = 0.75in
                   bottommargin = 0.75in
                   rightmargin  = 0.75in
                   leftmargin   = 0.75in
                   ;

           proc sql
              noprint ;
             select distinct color
                  , quote( strip( registry ))
                  , group
                      into : datacontrastcolors  separated by " "
                           , : valuesdisplay       separated by " "
                           , : null
             from &colornames_ds.
             order by group
             ;
           quit ;

           proc sql
              noprint ;
             select max( group )
                     into : max group separated by " "
             from &colornames_ds.
             ;
           quit ;

           ods listing close ;
           ods results ;

           ods rtf
               file  = "&colornames_path.\&colornames_plot_ds..rtf"
               nogtitle
               nogfootnote
               ;

           ods graphics on
             / height       = 10.5in
               width        = 5in
               border       = off
               outputfmt    = png
```

```
              attrpriority = none
              ;

          title1 height = 9pt "SAS Institute Inc. 2016. SAS(R) 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc." ;
          title2 height = 9pt "SAS Color Names and RGB Values in the SAS Registry page 1667" ;

          proc sgplot
             data = &colornames_plot_ds.
             noautolegend
             ;

            styleattrs datalinepatterns   = ( solid )
                       datacontrastcolors = ( &datacontrastcolors. )
                       datasymbols        = ( circlefilled )
                       ;

            series x = x
                   y = y
                 / group           = group
                   lineattrs       = ( thickness =  1 )
                   ;

            scatter x = x
                    y = y
                  / group          = group
                    markerattrs    = ( size      = 1 )
                    ;

            /****/
            xaxis label         = " "
                  values        = ( 0 1 2 )
                  display       = none
                  ;

            /****/
            yaxis  label         = " "
                   offsetmin     = 0.01
                   offsetmax     = 0.01
                   values        = ( 1 to &max_group. by 1 )
                   valuesdisplay = ( &valuesdisplay. )
                   valueshalign  = left
                   valueattrs    = ( size = 1pt )
                   type          = linear
                   fitpolicy     = none
                   ;

          run ;

          ods rtf close ;
          ods listing ;

          title " " ;

          %if &colornames_delete. = Y
          %then
             %do ;
                 %if %sysfunc( index( &colornames_ds. , . ))
                 %then
                    %do ;
                        %let __lb = %sysfunc( scan( &colornames_ds. , 1 , . )) ;
                        %let __ds = %sysfunc( scan( &colornames_ds. , 2 , . )) ;
                     %end ;
                  %else
                     %do ;
                         %let __lb = WORK ;
                         %let __ds = &colornames_ds. ;
                      %end ;

                 proc datasets
                    library = &__lb.
                    nolist
                    ;
                   delete &__ds. ;
                 quit ;

                 %if &colornames_ds. ne &colornames_plot_ds.
                 %then
                    %do ;
                        %if %sysfunc( index( &colornames_plot_ds. , . ))
                        %then
                           %do ;
                               %let __lb = %sysfunc( scan( &colornames_plot_ds. , 1 , . )) ;
                               %let __ds = %sysfunc( scan( &colornames_plot_ds. , 2 , . )) ;
                            %end ;
                         %else
                            %do ;
                                %let __lb = WORK ;
                                %let __ds = &colornames_plot_ds. ;
                             %end ;

                        proc datasets
                           library = &__lb.
                           nolist
                           ;
                          delete &__ds. ;
                        quit ;
                     %end ;
              %end ;

       %end ; /* END OF display_colornames = Y */

     /*************************************************/
     %if &display_linepatterns. = Y
     %then
        %do ;
            %if &linepatterns_path. = %str()
            %then %let linepatterns_path = %sysfunc( getoption( sasuser )) ;

            /* Create the data set to plot */
            data &linepatterns_ds. ;

              length label $ 23  ;

              do linepattern       = 1 to 46 ;

                select( linepattern ) ;
                   when (  1 ) label = " 1: Solid"               ;
                   when (  2 ) label = " 2: ShortDash"           ;
                   when (  4 ) label = " 4: MediumDash"          ;
                   when (  5 ) label = " 5: LongDash"            ;
                   when (  8 ) label = " 8: MediumDashShortDash" ;
                   when ( 14 ) label = "14: DashDashDot"         ;
                   when ( 15 ) label = "15: DashDotDot"          ;
                   when ( 20 ) label = "20: Dash"                ;
                   when ( 26 ) label = "26: LongDashShortDash"   ;
                   when ( 34 ) label = "34: Dot"                 ;
                   when ( 35 ) label = "35: ThinDot"             ;
                   when ( 41 ) label = "41: ShortDashDot"        ;
                   when ( 42 ) label = "42: MediumDashDotDot"    ;
                    otherwise label = put( linepattern , 8.0 -L ) ;
                end ;

                y + 1 ;
                x = 0 ;
                output ;
                x = 1 ;
                output ;
                x = 2 ;
                output ;
              end ;

            run ;
```

```
843
844           /*************************************************/
845           options orientation  = portrait
846                   topmargin    = 0.75in
847                   bottommargin = 0.75in
848                   rightmargin  = 0.75in
849                   leftmargin   = 0.75in
850                   ;
851
852           proc sql
853              noprint ;
854             select distinct linepattern
855                  , quote( strip( label ))
856                     into : datalinepatterns  separated by " "
857                          , : valuesdisplay   separated by " "
858             from &linepatterns_ds.
859             ;
860           quit ;
861
862           ods listing close ;
863           ods results ;
864
865           ods rtf
866               file  = "&linepatterns_path.\&linepatterns_ds..rtf"
867               nogtitle
868               nogfootnote
869               ;
870
871           ods graphics on
872             / height       = 10in
873               width        = 5in
874               border       = off
875               outputfmt    = png
876               attrpriority = none
877               ;
878
879           title1 height = 9pt "SAS Institute Inc. 2016. SAS(R) 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc." ;
880           title2 height = 9pt "Line Attributes and Patterns" ;
881           title3 height = 9pt "Table 14.2 Full List of Line Patterns page 1651" ;
882
883           proc sgplot
884              data = &linepatterns_ds.
885              noautolegend
886              ;
887
888             styleattrs datalinepatterns   = ( &datalinepatterns. )
889                        datacontrastcolors = ( black )
890                        ;
891
892             series x = x
893                    y = y
894                  / group           = linepattern
895                    lineattrs       = ( thickness =  3 )
896                    ;
897
898             /****/
899             xaxis label         = " "
900                   values        = ( 0 1 2 )
901                   display       = none
902                   ;
903
904             /****/
905             yaxis  label         = " "
906                    offsetmin     = 0.01
907                    offsetmax     = 0.01
908                    values        = ( 1 to 46 by 1 )
909                    valuesdisplay = ( &valuesdisplay. )
910                    valueshalign  = left
911                    valueattrs    = ( size = 9pt )
912                    type          = linear
913                    fitpolicy     = none
914                    ;
915
916           run ;
917
918           ods rtf close ;
919           ods listing ;
920
921           title " " ;
922
923           %if &linepatterns_delete. = Y
924           %then
925              %do ;
926                   %if %sysfunc( index( &linepatterns_ds. , . ))
927                   %then
928                      %do ;
929                           %let __lb = %sysfunc( scan( &linepatterns_ds. , 1 , . )) ;
930                           %let __ds = %sysfunc( scan( &linepatterns_ds. , 2 , . )) ;
931                       %end ;
932                    %else
933                       %do ;
934                           %let __lb = WORK ;
935                           %let __ds = &linepatterns_ds. ;
936                       %end ;
937
938                   proc datasets
939                      library = &__lb.
940                      nolist
941                      ;
942                     delete &__ds. ;
943                   quit ;
944              %end ;
945
946        %end ; /* END OF display_linepatterns = Y */
947
948      /*************************************************/
949      %if &display_marker_symbols. = Y
950      %then
951         %do ;
952           %if &marker_symbols_path. = %str()
953           %then %let marker_symbols_path = %sysfunc( getoption( sasuser )) ;
954
955           /* Create the data set to plot */
956           data &marker_symbols_ds. ;
957
958             length label  $ 20 ;
959
960             retain x 1 ;
961             do label   =   "ArrowDown"
962                          , "StarFilled"
963                          , "Star"
964                          , "Asterisk"
965                          , "Tack"
966                          , "Tilde"
967                          , "Circle"
968                          , "CircleFilled"
969                          , "Triangle"
970                          , "TriangleFilled"
971                          , "TriangleDown"
972                          , "TriangleDownFilled"
973                          , "TriangleLeft"
974                          , "TriangleLeftFilled"
975                          , "TriangleRight"
976                          , "TriangleRightFilled"
977                          , "Diamond"
978                          , "DiamondFilled"
979                          , "GreaterThan"
980                          , "LessThan"
981                          , "Hash"
982                          , "HomeDown"
983                          , "HomeDownFilled"
984                          , "IBeam"
```

```
                        , "Union"
                        , "Plus"
                        , "X"
                        , "Y"
                        , "Z"
                        , "Square"
                        , "SquareFilled"
                        ;
               y + 1 ;
               output ;
             end ;
          run ;

          /*********************************************/
          options orientation  = portrait
                  topmargin    = 0.75in
                  bottommargin = 0.75in
                  rightmargin  = 0.75in
                  leftmargin   = 0.75in
                  ;

          proc sql
             noprint ;
             select distinct label
                  , quote( strip( label ))
                  , y
                       into : datasymbols      separated by " "
                          , : valuesdisplay    separated by " "
                          , : null
             from &marker_symbols_ds.
             order by y
             ;
          quit ;

          proc sql
             noprint ;
             select max( y )
                    into : max_y separated by " "
             from &marker_symbols_ds.
             ;
          quit ;

          ods listing close ;
          ods results ;

          ods rtf
              file  = "&marker_symbols_path.\&marker_symbols_ds..rtf"
              nogtitle
              nogfootnote
              ;

          ods graphics on
            / height       = 10.5in
              width        = 5in
              border       = off
              outputfmt    = png
              attrpriority = none
              ;

          title1 height = 9pt "SAS Institute Inc. 2016. SAS(R) 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc." ;
          title2 height = 9pt "Marker Attributes and Symbols page 1657" ;
          title3 height = 9pt "Table 14.5 Supported Marker Symbols" ;

          proc sgplot
             data = &marker_symbols_ds.
             noautolegend
             ;

            styleattrs datacontrastcolors = ( black )
                       datasymbols        = ( &datasymbols. )
                      ;

            scatter x = x
                    y = y
                  / group           = label
                    markerattrs     = ( size      = 5 )
                    ;

            /****/
            xaxis label         = " "
                  values        = ( 0 1 2 )
                  display       = none
                  ;

            /****/
            yaxis  label         = " "
                   offsetmin     = 0.01
                   offsetmax     = 0.01
                   values        = ( 1 to &max_y. by 1 )
                   valuesdisplay = ( &valuesdisplay. )
                   valueshalign  = left
                   valueattrs    = ( size = 9pt )
                   type          = linear
                   fitpolicy     = none
                   ;

          run ;

          ods rtf close ;
          ods listing ;

          title " " ;

          %if &marker_symbols_delete. = Y
          %then
             %do ;
                %if %sysfunc( index( &marker_symbols_ds. , . ))
                %then
                   %do ;
                        %let   lb = %sysfunc( scan( &marker symbols ds. , 1 , . )) ;
                        %let __ds = %sysfunc( scan( &marker_symbols_ds. , 2 , . )) ;
                    %end ;
                 %else
                    %do ;
                        %let __lb = WORK ;
                        %let __ds = &marker_symbols_ds. ;
                    %end ;

                proc datasets
                   library = &__lb.
                   nolist
                   ;
                  delete &__ds. ;
                quit ;
              %end ;

       %end ; /* END OF display_marker_symbols = Y */

    %if     &display_colornames.     = Y
        or  &display_linepatterns.   = Y
        or  &display_marker_symbols. = Y
    %then %goto __END ;

    /*****************************************************************************************************/
    /* Obtain the dimensions of the arrays */
    %let cc_num = %eval( %sysfunc( countc( %nrbquote(&contrastcolor_list.) , %str(,))) + 1 ) ;
    %let ls_num = %eval( %sysfunc( countc( %nrbquote(&linestyle list.)     , %str(,))) + 1 ) ;
    %let ms_num = %eval( %sysfunc( countc( %nrbquote(&markersymbol_list.)  , %str(,))) + 1 ) ;

    %if %nrbquote(&markercolor_list.) = %str()
    %then
       %do ;
```

```
          %let markercolor_list = &contrastcolor_list. ;
          %let mc_index = cc ;
      %end ;
   %else %let mc_index = mc ;

   %let mc_num = %eval( %sysfunc( countc( %nrbquote(&markercolor_list.) , %str(,))) + 1 ) ;

   /*********/
   %if     &out_ds_attrib_cycle. ne %str()
        or  &cartesian_join.      =  Y
   %then
      %do ;
         %if %sysfunc( prxmatch( /^(?:cc|mc|ls|ms)$/i , &first_index. )) = 0
         %then
            %do ;
               %put E%str(RROR:) FIRST_INDEX = &first_index. must be one of CC (ContrastColor), MC (MarkerColor), LS (LineStyle), and MS (MarkerSymbol), case-insensitive. ;
               %goto __END ;
            %end ;
          %else %let first_index = &first_index. ;

         %if %sysfunc( prxmatch( /^(?:cc|mc|ls|ms)$/i , &second_index. )) = 0
         %then
            %do ;
               %put E%str(RROR:) SECOND_INDEX = &second_index. must be one of CC (ContrastColor), MC (MarkerColor), LS (LineStyle), and MS (MarkerSymbol), case-insensitive. ;
               %goto __END ;
            %end ;
          %else %let second_index = &second_index. ;

         %if %sysfunc( prxmatch( /^(?:cc|mc|ls|ms)$/i , &third_index. )) = 0
         %then
            %do ;
               %put E%str(RROR:) third_INDEX = &third_index. must be one of CC (ContrastColor), MC (MarkerColor), LS (LineStyle), and MS (MarkerSymbol), case-insensitive. ;
               %goto __END ;
            %end ;
          %else %let third_index = &third_index. ;

         /****/
         %if %sysfunc( prxmatch(
/^(?:ccmclsms|ccmcmsls|cclsmcms|cclsmsmc|ccmsmcls|ccmslsmc|mccclsms|mcccmsls|mclsccms|mclsmscc|mcmsccls|mcmslscc|lsccmcms|lsccmsmc|lsmcccms|lsmcmscc|lsmsccmc|lsmsmccc|msccmcls|mscc
lsmc|msmcccls|msmclscc|mslsccmc|mslsmccc)$/i , &first_index.&second_index.&third_index.&fourth_index. )) = 0
         %then
            %do ;
               %put E%str(RROR:) FIRST_INDEX  = &first_index.  ;
               %put E%str(RROR:) SECOND_INDEX = &second_index. ;
               %put E%str(RROR:) THIRD_INDEX  = &third_index.  ;
               %put E%str(RROR:) FOURTH_INDEX = &fourth_index. ;
               %put E%str(RROR:) must be a combination of one of CC (ContrastColor), MC (MarkerColor), LS (LineStyle), and MS (MarkerSymbol), ;
               %put E%str(RROR:) case-insensitive, with no repeats or missing values. ;
               %goto __END ;
            %end ;

          /*****************************/
          proc format ;
            value $ index
              "cc" = "linecolor"
              "mc" = "markercolor"
              "ls" = "linepattern"
              "ms" = "markersymbol"
              ;
         run ;

      %end ; /* END OF out_ds_attrib_cycle ne str() or cartesian_join = Y */

   %if &out_ds_attrib_cycle. ne %str()
   %then
      %do ;
         data &out_ds_attrib_cycle. ;

           retain %sysfunc( putc( &first_index.  , $index. ))
                  %sysfunc( putc( &second_index. , $index. ))
                  %sysfunc( putc( &third_index.  , $index. ))
                  %sysfunc( putc( &fourth_index. , $index. ))
                  ;

           length linecolor      $  8
                  markercolor    $  8
                  markersymbol   $ 20
                  linepattern    &length_linepattern.
                  ;

           array cc ( &cc_num. )
                 $ 8
                 _temporary_
                 ( &contrastcolor_list. )
                 ;

           array mc ( %if &mc_index. = mc %then &mc_num. ;
                        %else 1 ;
                    )
                 $ 8
                 _temporary_
                 ( %if &mc_index. = mc %then &markercolor_list. ;
                     %else " " ;
                 )
                 ;

           array ls ( &ls_num. )
                 _temporary_
                 ( &linestyle_list. )
                 ;

           array ms ( &ms_num. )
                 $ 20
                 _temporary_
                 ( &markersymbol_list. )
                 ;

           /************/
           do __1 = 1 to ( dim( &first_index. )) ;
             do   2 = 1 to ( dim( &second index. )) ;
               do __3 = 1 to ( dim( &third_index. )) ;
                 do __4 = 1 to ( dim( &fourth_index. )) ;

                   linecolor      = cc( %if &first_index. = cc %then __1 ;
                                        %else %if &second_index. = cc %then __2 ;
                                         %else %if &third_index. = cc %then __3 ;
                                          %else %if &fourth_index. = cc %then __4 ;
                                    ) ;


                   %if &mc_index. = mc
                   %then
                      %do ;
                          markercolor    = mc( %if &first index. = cc %then   1 ;
                                               %else %if &second_index. = cc %then __2 ;
                                                %else %if &third_index. = cc %then __3 ;
                                                 %else %if &fourth_index. = cc %then __4 ;
                                            ) ;
                      %end ;
                     %else markercolor = linecolor %str(;) ;

                   linepattern    = ls( %if &first_index. = ls %then __1 ;
                                        %else %if &second_index. = ls %then __2 ;
                                         %else %if &third index. = ls %then  3 ;
                                          %else %if &fourth_index. = ls %then __4 ;
                                    ) ;

                   markersymbol   = ms( %if &first_index. = ms %then __1 ;
                                        %else %if &second_index. = ms %then __2 ;
                                         %else %if &third_index. = ms %then __3 ;
                                          %else %if &fourth_index. = ms %then __4 ;
                                    ) ;
```

```
                  output ;

               end ;
             end ;
           end ;
         end ;

        run ;

    %end ; /* END OF out_ds_attrib_cycle ne str() */

  /*************************************************************/
  %if &out_ds_dattrmap. ne %str()
  %then
     %do ;
        data &out ds dattrmap.
               %if &debug. = N
               %then ( keep   = id
                               value
                               markercolor
                               markersymbol
                               linecolor
                               linepattern
                     )
              ;
           ;

         length id             $ &length id.
                value          &length_value.
                linecolor      $  8
                markercolor    $  8
                linepattern    &length_linepattern.
                markersymbol   $ 20
                ;

         array cc ( &cc_num. )
               $ 8
               _temporary_
               ( &contrastcolor_list. )
               ;

         array mc ( &mc_num. )
               $ 8
               _temporary_
               ( &markercolor_list. )
               ;

         array ls ( &ls_num. )
               _temporary_
               ( &linestyle_list. )
               ;

         array ms ( &ms_num. )
               $ 20
               _temporary_
               ( &markersymbol_list. )
               ;

         if _n_ = 1
         then
           do ;
              dim_cc = dim( cc ) ;
              dim_mc = dim( mc ) ;
              dim_ls = dim( ls ) ;
              dim_ms = dim( ms ) ;
           end ;

         retain dim_cc
                dim_mc
                dim_ls
                dim_ms
                cc_index  0
                mc_index  0
                ls_index  0
                ms_index  0
                id        "&id."
                ;

         set &in_ds. ;
         by &by. ;

         &index_set_wrapper_top.
         &cc_index_set.
         &mc_index_set.
         &ls_index_set.
         &ms_index_set.
         &index_set_wrapper_bottom.

         &index_increment_wrapper_top.
         &cc_index_increment.
         &mc_index_increment.
         &ls_index_increment.
         &ms_index_increment.
         &index_increment_wrapper_bottom.

         markercolor    = cc( mc_index ) ;
         linecolor      = cc( cc_index ) ;
         linepattern    = ls( ls_index ) ;
         markersymbol   = ms( ms_index ) ;

         &index_reset_wrapper_top.
         &cc_index_reset.
         &mc_index_reset.
         &ls_index_reset.
         &ms_index_reset.
         &index_reset_wrapper_bottom.

        run ;

    %end ; /* END OF out_ds_dattrmap ne str() */

  %else %if &cartesian_join. = Y
  %then
    %do ;
       data __linecolor ;
         array cc ( &cc_num. )
               $ 8
               _temporary_
               ( &contrastcolor_list. )
               ;
         do linecolor order = 1 to &cc num. ;
           linecolor = cc( linecolor_order ) ;
           output ;
         end ;
       run ;

       data __markercolor ;
         array mc ( &mc_num. )
               $ 8
               _temporary_
               ( &markercolor_list. )
               ;
         do markercolor order = 1 to &mc num. ;
           markercolor = mc( markercolor_order ) ;
           output ;
         end ;
       run ;

       data __linepattern ;
         array ls ( &ls_num. )
               _temporary_
```

```
                  ( &linestyle_list. )
                  ;
            do linepattern_order = 1 to &ls_num. ;
              linepattern = ls( linepattern_order ) ;
              output ;
            end ;
          run ;

          data __markersymbol ;
            array ms ( &ms_num. )
                  $ 20
                  _temporary_
                  ( &markersymbol_list. )
                  ;
            do markersymbol_order = 1 to &ms_num. ;
              markersymbol = ms( markersymbol_order ) ;
              output ;
            end ;
          run ;

          proc sql ;
            create table &cartesian_join_ds. as
            select a.%sysfunc( putc( &first_index.  , $index. ))
                 , b.%sysfunc( putc( &second_index. , $index. ))
                 , c.%sysfunc( putc( &third_index.  , $index. ))
                 , d.%sysfunc( putc( &fourth_index. , $index. ))
                 , a.%sysfunc( trim( %sysfunc( putc( &first_index.  , $index. ))))_order
                 , b.%sysfunc( trim( %sysfunc( putc( &second_index. , $index. ))))_order
                 , c.%sysfunc( trim( %sysfunc( putc( &third_index.  , $index. ))))_order
                 , d.%sysfunc( trim( %sysfunc( putc( &fourth_index. , $index. ))))_order
            from __%sysfunc( putc( &first_index.  , $index. )) as a
               , __%sysfunc( putc( &second_index. , $index. )) as b
               , __%sysfunc( putc( &third_index.  , $index. )) as c
               , __%sysfunc( putc( &fourth_index. , $index. )) as d
            order by a.%sysfunc( trim( %sysfunc( putc( &first_index.  , $index. ))))_order
                   , b.%sysfunc( trim( %sysfunc( putc( &second_index. , $index. ))))_order
                   , c.%sysfunc( trim( %sysfunc( putc( &third_index.  , $index. ))))_order
                   , d.%sysfunc( trim( %sysfunc( putc( &fourth_index. , $index. ))))_order
            ;
          quit ;

          proc datasets
             library = WORK
             nolist
             ;
            delete __linecolor
                   __markercolor
                   __linepattern
                   __markersymbol
                   ;
          quit ;

       %end ; /* END OF cartesian_join = Y */

    /*************************************************************/
    %if &test_dattrmap. ne %str()
    %then
       %do ;

          %let orientation_orig =  %sysfunc( getoption( orientation )) ;
          options orientation = landscape ;

          %if &colornames_path. = %str()
          %then %let colornames_path = %sysfunc( getoption( sasuser )) ;

          data   test
                 ( drop   = id ) ;
            set &test_dattrmap.
                 ( keep   = id
                            value
                   where  = ( id = "&test_id." )
                 )
                 end = end
               ;
            y = _n_ ;
            do x = 0 to 2 ;
              output ;
            end ;

            if end then call symputx( "max_obs" , put( y , 8. )) ;

          run ;

          %if &add_attr_names. =  Y
          %then
             %do ;
                proc sql ;
                  create table __dattrmap_names as
                  select a.*
                       , case when b.color ne " "
                              then cat( strip( a.linecolor )
                                      , " ("
                                      , strip( b.color )
                                      , ")"
                                      )
                               else a.linecolor
                         end as linecolor_name
                            length = 70
                  from ( select a1.value
                              , a1.markercolor
                              , a1.linecolor
                              , a1.linepattern
                              , a1.markersymbol
                         from &test_dattrmap. as a1
                         where a1.id = "&test_id."
                       ) as a
                       left join colornames as b
                          on a.linecolor = b.hex_color
                  ;
                quit ;

                proc sql
                   undo_policy = none ;
                  create table __dattrmap_names as
                  select a.*
                       , b.color
                       , case when b.color ne " "
                              then cat( strip( a.markercolor )
                                      , " ("
                                      , strip( b.color )
                                      , ")"
                                      )
                               else a.markercolor
                         end as markercolor_name
                            length = 70
                  from __dattrmap_names as a
                       left join colornames as b
                          on a.markercolor = b.hex_color
                  ;
                quit ;

                proc sql
                   undo_policy = none ;
                  create table __dattrmap_names as
                  select a.*
                       , case when b.linepattern_name ne " "
                              then cat( strip( put( a.linepattern , 8.0 ))
                                      , " ("
                                      , strip( b.linepattern_name )
                                      , ")"
                                      )
                               else put( a.linepattern , 8.0 -L )
```

```
                        end as linepattern_name
                           length = 30
                  from __dattrmap_names as a
                       left join commonly_used_line_patterns as b
                         on a.linepattern = b.linepattern
                  ;
                quit ;

                /*********/
                proc sql
                   undo_policy = none ;
                  create table __test as
                  select a.*
                       , b.linecolor_name
                       , b.markercolor_name
                       , b.linepattern_name
                       , b.markersymbol
                  from __test                      as a
                       left join __dattrmap_names  as b
                         on a.value = b.value
                  order by a.y
                         , a.x
                  ;
                quit ;

             %end ; /* END OF add_attr_names =  Y */

          /*****/
          %let max_pages = 1 ;

          %if &test_obs_per_page. ne %str()
          %then
             %do ;
                %if &max_obs. < &test_obs_per_page. %then %let test_obs_per_page = %str() ;
                 %else
                    %do ;
                        data __test ;
                          set __test
                                end = end
                              ;
                          __page = ( ceil( y / &test_obs_per_page. )) ;

                          if end then call symputx( "max_pages" , put( __page , 8. )) ;
                        run ;
                    %end ;
             %end ;

          ods listing close ;

          ods rtf
              file  = "&test_path.\&test_rtf_name..rtf"
              nogtitle
              nogfootnote
              ;

          ods graphics on
            / width        = &test_ods_graphics_width.
              border       = off
              outputfmt    = png
              attrpriority = none
              ;

          %do __j = 1 %to &max_pages. ;

             proc sql
                noprint ;
              select strip( put( min( y ) , 8. ))
                   , strip( put( max( y ) , 8. ))
                        into : min_y
                           , : max_y

              from __test
              %if &max_pages. > 1 %then where __page = &__j. ;
              ;

              select distinct y
                   , quote( catx( ": "
                                , put( y , 8. )
                                , value
                                )
                          )
                        into : null
                           , : valuesdisplay separated by " "
              from __test
              %if &max_pages. > 1 %then where __page = &__j. ;
              ;

              %if &add_attr_names. =  Y
              %then
                 %do ;
                     select distinct y
                          , quote( cat( "LC = "
                                      , strip( linecolor_name )
                                      , " / MC = "
                                      , strip( markercolor_name )
                                      , " / LP = "
                                      , strip( linepattern_name )
                                      , " / MS = "
                                      , strip( markersymbol )
                                      )
                                 )
                        into : null
                           , : valuesdisplay2 separated by " "
                  from __test
                  %if &max_pages. > 1 %then where __page = &__j. ;
                  ;
                 %end ;

             quit ;

             proc sgplot
                data     = __test
                           %if &max_pages. > 1
                           %then
                              ( where  = (__page = &__j. ))
                           ;
                dattrmap = &test_dattrmap.
                noautolegend
                aspect = &test_sgplot_aspect.
                ;

               series  x = x
                       y = y
                     /  group            = value
                        lineattrs        = ( thickness = &test_lineattrs_thickness. )
                        attrid           = &test_id.
                        ;

               scatter  x = x
                        y = y
                      /  group           = value
                         markerattrs     = ( size = &test_markerattrs_size. )
                         attrid          = &test_id.
                         ;

               /****/
               xaxis label         = " "
                     values        = ( 0 1 2 )
                     display       = none
                     ;

               /****/
               yaxis  label        = " "
```

```
                    offsetmin     = 0.03
                    offsetmax     = 0.03
                    values        = ( &min_y. to &max_y. by 1 )
                    valuesdisplay = ( &valuesdisplay. )
                    valueshalign  = left
                    valueattrs    = ( size = &test_valueattrs_size. )
                    type          = linear
                    fitpolicy     = none
                    ;

             /***/
             %if &add_attr_names. =  Y
             %then
                %do ;
                    series  x  = x
                            y = y
                         /  group           = value
                            lineattrs       = ( thickness = 0 )
                            y2axis
                            ;

                    scatter  x  = x
                             y = y
                          /  group           = value
                             markerattrs     = ( size = 0 )
                             y2axis
                             ;

                    y2axis  label         = " "
                            offsetmin     = 0.03
                            offsetmax     = 0.03
                            values        = ( &min_y. to &max_y. by 1 )
                            valuesdisplay = ( &valuesdisplay2. )
                            valueshalign  = left
                            valueattrs    = ( size = &test_valueattrs_size. )
                            type          = linear
                            fitpolicy     = none
                            ;
                %end ; /* END OF add_attr_names =  Y */

            run ;

         %end ; /* CYCLED THROUGH __J */

         ods rtf close ;
         ods listing ;

         options orientation = &orientation_orig. ;

         proc datasets
            library = WORK
            nolist
            ;
           delete   test
                    __dattrmap_names
                    ;
         quit ;

      %end ; /* END OF test = Y */

   %__END:

%mend  mac_sg_dattrmap ;
```