%Paint Macro

The `%Paint` autocall macro interpolates between colors. This macro is not a part of the experimental design or marketing research family of macros. It exists for totally different reasons. This macro has two main uses.

- You can use it with ODS. The macro can create a macro that specifies an `ods cellstyle as` statement for use in PROC TEMPLATE for coloring the background around entries in a table.

- You can use it with PROC INSIGHT. The macro can read an input `data=` data set and create an `out=` data set with a new variable `_obstat_` that contains observation symbols and colors interpolated from the `colors=` list based on the values `var=` variable. Three interpolation methods are available with the `level=` option.

This macro creates temporary work data sets named `tempdata`, `tempdat2`, `tempdat3`, and `tempdat4`.

It makes a temporary macro called "`_`".

When a `cellstyle` macro is not created, the paint macro by default creates an `out=` data set named `colors`.

%Paint Macro Options

The following options can be used with the `%Paint` macro:

Option	Description
help	(positional) "help" or "?" displays syntax summary
values=*do-list*	input data values
var=*variable-name*	input variable
data=*SAS-data-set*	input SAS data set
out=*SAS-data-set*	output SAS data set
macro=*macro-name*	SAS macro name
colors=*<color-list> <data-value-list>*	colors list
level=*measurement-level*	measurement level of the data
symbols=*symbols-list*	list of symbols
format=*SAS-format*	nominal and ordinal variable format
order=*summary-order*	nominal and ordinal variable order
select=*n*	selection state
show=*n*	show/hide state
include=*n*	include/exclude state
label=*n*	label/unlabel state
rgbround=*rounding-list*	RGB value rounding instructions
missing=*missing-specification*	missing value handling
debug=*debug-string*	debugging information to display

You can specify either of the following to display the option names and simple examples of the macro syntax:

```
%paint(help)
%paint(?)
```

One of the following two options must be specified:

values= *do-list*
specifies the input data values. This option primarily exists for use with ODS. You can tell the macro to create input data set based on a `do` list provided with this option. Example: `values=0 to 10 by 0.25`.

var= *variable-name*
specifies a variable with the input data values. This option primarily exists for use with PROC INSIGHT. Use this option when there is an input `data=` data set.

All of these next options can optionally be specified:

data= *SAS-data-set*
specifies the input SAS data set (when `var=` but not `values=` is specified). This option primarily exists for use with PROC INSIGHT.

out= *SAS-data-set*
specifies the output SAS data set. The default, when `macro=` is not specified, is `out=colors`. This option primarily exists for use with PROC INSIGHT.

macro= *macro-name*
specifies the name of the SAS macro to create instead of creating an `out=` data set. This option primarily exists for use with ODS. By default, no macro is created.

colors= *<color-list> <data-value-list>*
specifies the colors list. The colors must be selected from: red, green, blue, yellow, magenta, cyan, black, white, orange, brown, gray, olive, pink, purple, violet. For other colors, specify the RGB color name (CX*rrggbb* where *rr* is the red value, *gg* is the green, and *bb* is blue, all three specified in hex, 00 to FF). When no list is specified, the default is `colors=blue magenta red`. The option `colors=red green 1 10`, interpolates between red and green, based on the values of the `var=` variable, where values of 1 or less map to red, values of 10 or more map to green, and values in between map to colors in between. The option `colors=red yellow green 1 5 10`, interpolates between red at 1, yellow at 5, and green at 10. If the data value list is omitted it is computed from the data.

level= *measurement-level*
specifies the measurement level of the data. Values include `interval`, `ordinal`, and `nominal`. The option `level=interval` (the default) interpolates on the actual values. The option `level=ordinal` interpolates on the ranks of the input values. This can even work with character variables; the ranks

are just the category numbers. The option `level=nominal` specifies that there is one color or symbol per category. Only the first three characters of the measurement level are checked.

symbols= *symbols-list*
provides a list of symbols from page 323 of the Version 6 PROC INSIGHT manual. Specify integers from 1 to 8 or select from: square plus circle diamond x up down star. Note that 'triangle' is not specified with 'up' and 'down'. Equivalent examples: `symbols=1 2 3 4 5 6`, `symbols=square plus circle diamond x up`, `symbols=1 plus circle 4 5 up`. Note that when `level=interval`, only the first symbol is used. With `level=nominal` and `level=ordinal`, the first symbol is used for the first category, the second symbol is used for the second category, ..., and if there are more categories than symbols, the last symbol is reused for all subsequent categories. Extra symbols are ignored. The default first symbol is circle, and the last symbol is substituted for invalid symbols.

format= *SAS-format*
specifies the format for the values used with `level=nominal` and `level=ordinal` variables.

order= *summary-order-option*
specifies the PROC SUMMARY `order=` option for `level=ordinal` and `level=nominal`.

These next four options specify the first four columns of _obstat_ variable from page 323 of the Version 6 PROC INSIGHT manual.

Column 5, the symbol, is specified with `symbols=`. The values must be numeric expressions including constants, variables, and arithmetic expressions. All nonzero expression results are converted to 1.

select= *n*
specifies the selection state. The value 0 means not selected. This option primarily exists for use with PROC INSIGHT.

show= *n*
specifies the show/hide state. The value 0 means hide. This option primarily exists for use with PROC INSIGHT.

include= *n*
specifies the include/exclude state. The value 0 means exclude. This option primarily exists for use with PROC INSIGHT.

label= *n*
specifies the label/unlabel state. The value 0 means unlabel. This option primarily exists for use with PROC INSIGHT.

rgbround= *rounding-list*
specifies instructions for rounding the RGB values. The first value is used to round the `colors=var` variable. Specify a positive value to have the variable rounded to multiples of that value. Specify a negative value n to have a maximum of abs(n) colors. For the other three values, specify positive

values. The last three values are rounding factors for the red, green, and blue component of the color. By default, when a value is missing, there is no rounding. The default is `rgbround=-99 1 1 1`, which creates a maximum of 99 colors and rounds the RGB values to integers. You could specify larger values for the last three values to have, for examples, the color values be multiples of two or four.

missing=

specifies how missing values in the `var=` variable are handled. By default, when `missing=` is null, the observation is not shown (`show=0` is set). The specified value is a color followed by a symbol. When only one value is specified in the `symbols=` list, the symbol is optional, and the `symbols=` symbol is used.

debug= *debug-string*

specifies types of debugging information to display. The option `debug=vars` displays macro options and macro variables for debugging. The option `debug=dprint` displays intermediate data sets. The option `debug=notes` suppresses the specification of `options nonotes` during most of the macro. The option `debug=time` displays the total macro run time. The option `debug=mprint` runs the macro with `options mprint`. Create a list for more than one type of debugging. Example: `debug=vars dprint notes time mprint`.