

Using ODS and the Macro Facility to Construct Color Charts and Scales for SAS® Software Applications

Perry Watts, Independent Consultant, Elkins Park, PA

Abstract

Highly accessible HTML files have increased the importance and use of color as means for enhancing reports and summarizing data. The reference manual for SAS/GRAPH® software contains an entire chapter devoted exclusively to color. In fact, the chapter lists 292 predefined SAS colors by their NAMES, RGB (red/green/blue) and HLS (hue/light/saturation) codes. Nevertheless, the list is hampered by the absence the colors themselves. VPAPB for "very pale purplish blue" continues to remain in the eye of the beholder.

With ODS it is now possible to construct a fully annotated color chart from the list of predefined SAS colors found in *SAS OnlineDoc*. Actual colors are displayed in an HTML format along with their NAMES, RGB, and HLS codes.

A color scale new to SAS software and developed from the color chart is also described in the paper. The color scale uses a decimal conversion macro in a looping structure to display one hue in varying degrees of luminescence. The paper provides illustrations, instructions for construction, and applications of both the color chart and the color scale. These visual aids should promote the effective use of color for communicating information in SAS Software applications.

Constructing a SAS Color Chart

To obtain a listing of the predefined SAS colors appearing on pages 146-154 in Version 8 of the SAS/GRAPH manual, download the table from *SAS OnlineDoc* in HTML format into a text file by completing the following steps:

- 1) Access the table of contents and drill down to:
SAS/GRAPH
SAS/GRAPH COLORS
Color-naming schemes
- 2) Select *All* and then *Copy* and *Paste* to insert the material into a Microsoft Word file.
- 3) Delete extraneous text and annotations keeping only the table.
- 4) Select the table, and press *Convert Table to Text* specifying comma delimited output.
- 5) Save the output to a TXT file that is used as input to a SAS program.

The SAS program that processes the downloaded text file uses a macro together with traffic lighting to add color blocks to output from PROC Report. The macro that is invoked from a compute block in PROC Report assigns the correct background color to Column #2 with a SELECT statement:

```
%macro selectRGB;
  select (_C2_);
  %do i = 1 %to &NColors;
    when (&i.) CALL DEFINE(_COL_, "STYLE",
      "STYLE=[ BACKGROUND=&1&&RgBN&i.]");
  %end;
  otherwise;
  end;
%mend selectRGB;
```

- 1 Macro variable &&RgBN&i is created in a Data _NULL_ step that processes all 292 colors. It contains an RGB code for a particular color.

Macro variables have to be used for assigning background colors, because the syntax for the STYLE option in the ODS DEFINE statement calls for unquoted constant text, not a SAS variable. Here is how the macro is invoked in PROC REPORT:

```
ods listing close;
ods html
  body='c:\SUGI27\SASColorChrt.html';
proc report data=X nowindows headline
  headskip split='*' ls=100 ps=40;
  columns cGroup 1 CNum cName cDesc RGB HLS;
  define cGroup / order width=15
    'Color*Group';
  define cNum / display width=2
    @format=missf. 'C';
  define cName / display width=10
    'SAS*Color Name';
  define cDesc / display width=30
    'Color Description';
  define RGB / display width=8 'RGB';
  define HLS / display width=8 'HLS';
  compute CNum;
    @%selectRGB;
  endcomp;
  title1 'SAS Color Chart';
run;
ods html close;
ods listing;
```

- 1 The variable CNUM is column #2, the argument of the SELECT statement in the SELECTRGB macro.
- 2 CNUM is set to missing with the MISSF format so that only the background color is displayed. The MISSF format is defined with a single range:
value missf low-high=' ';
- 3 The SELECTRGB macro is invoked from the COMPUTE block.

Selected Displays of the Color Chart

A partial listing of uniquely defined SAS colors not belonging to a larger color group is displayed in Figure 1. The color description is omitted from the display, since it is identical to the SAS color name. Otherwise the SAS code used for generating this table is identical to that described in the previous section.

Different SAS Color NAMES can map to the same color defined by the RGB code. Figure 2 lists all duplicates by name, RGB and HLS codes. The displayed shades confirm the presence of synonyms in the predefined color list.

There is also one error in the predefined color list that has remained undetected up to now. Technically, the HLS and RGB codes for the same color should display identical shades. In Figure 3, however, the two color blocks are distinctly different for *vivid purple*.

Color Group	C	SAS Color Name	RGB	HLS
unique		CYAN	CX00FFFF	H12C80FF
		STEEL	CX3883A8	H1407080
		CHARCOAL	CX4F4F4F	H0784F00
		MAROON	CX700000	H07838FF
		LIME	CXC0FF81	H0D2C0FF
		LILAC	CXE06090	H062A0AC
		TAN	CXE0A860	H09AA0AC
		CREAM	CXE8D898	H0A8C0A2
		SALMON	CXFF0055	H06480FF
		MAGENTA	CXFF00FF	H03C80FF
		ROSE	CXFF6060	H078B0FF
		GOLD	CXFFAA00	H0A080FF

Figure 1. A subset of unique predefined SAS colors along with their actual shades.

Duplicate RGB Listings of SAS Defined Colors

Color Group	RGBc	HLSc	SAS Color Name	Color Description	RGB#	HLS#
black			RBK	reddish black	CX191919	H06D1904
			GBL	greenish black	CX191919	H0F81900
			OLBL	olive black	CX191919	H0CB1900
			PBL	purplish black	CX191919	H0341901
			BBL	bluish black	CX191919	H0011901
gray			DAGGR	dark greenish gray	CX595959	H0F85900
			DAGR	dark gray	CX595959	H0005900
			GGR	greenish gray	CX8C8C8C	H0F88C01
			MEGR	medium gray	CX8C8C8C	H0008C00
			LIGGR	light greenish gray	CXBFBFBF	H0F8BF01
			LIGR	light gray	CXBFBFBF	H000BF00
green			GRLG	grayish yellow-green	CX8B9985	H0DD8F17
			GROLG	grayish olive green	CX8B9985	H0DD8F17

Figure 2. A list of predefined SAS Colors with different NAMES that map to the same RGB codes and shades. Note that a single RGB code can appear with different HLS codes. Even so, the HLS shades are identical to their RGB counterparts in every instance.

Duplicate HLS Listings of SAS Defined Colors

Color Group	RGBc	HLSc	SAS Color Name	Color Description	RGB#	HLS#
green			GRLG	grayish yellow-green	CX8B9985	H0DD8F17
			GROLG	grayish olive green	CX8B9985	H0DD8F17
purple			VIP	vivid purple	CX6F0980	H0347194
			BIP	brilliant purple	CXA030B2	H0347194

Figure 3. HLS and RGB colors should be identical for each SAS color name in the predefined list. RGB and HLS codes produce different color blocks for *vivid purple*.

The Color Scale

The idea for a SAS color scale comes from the outstanding user-interface supplied by Microsoft PowerPoint for adjusting colors. The caption in Figure 4 shows how to get to the *Custom* window in PowerPoint to define a color for any object on a slide. The color scale in the window is the narrow vertical band to the right of the main color display. The triangle next to it makes it work like a slider bar. By sliding the triangle up or down the color scale, the *New* upper half of the square in the lower right corner becomes lighter or darker reflecting the adjacent shade on the bar.

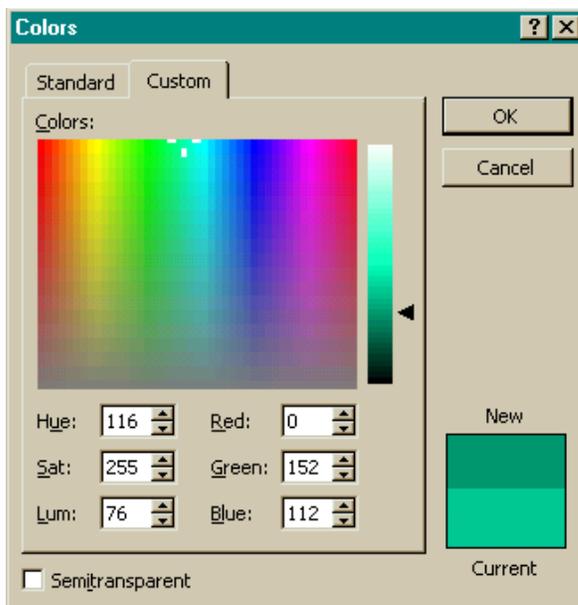


Figure 4. To get to the *Custom* window, press **Format:Colors and Lines: Color: More Colors: Custom** on the menu bar in PowerPoint.

When the slider bar is operated in the *Custom* window, all three RGB numbers change. In contrast, only the *lum(inescence)* or *light* number changes in the HLS scale, and the change reflects the position of triangle alongside the bar. The value 76 for *lum* in Figure 4, for example, is nearly one third of the distance between 0 and 255. Because only one component is needed from the HLS code to adjust the shade of a given color, it rather than the RGB code should be used for deriving a color scale in SAS.

Unfortunately, though, the HLS scale is configured differently in SAS. Some of the differences can be easily remedied. A decimal conversion macro changes a decimal number to hexadecimal, and code components can simply be rearranged to change a Microsoft *HSL* into a SAS *HLS*.

The story is not so simple for the *hue* component, however. In Microsoft, values for *hue* range from 0 to 255 whereas the maximum value for *hue* in SAS is 360 reflecting the Tektronix Color Standard. Even when a correction is made with interpolation, the same HLS code produces different colors in the two software packages. In contrast RGB codes are the same, but they cannot be used in this instance, because their components change unpredictably when a color scale is scanned.

The Color Scale is derived from the Color Chart

A SAS macro PLTCOLSC is used to create a color scale from the list of predefined SAS colors stored locally in a permanent SAS data set. The color scale in Figure 5 was made possible by matching the macro parameter GOLD to its SAS defined HLS code in the data set. Using a fully annotated color chart as a guide is the only way to get an exact color scale in SAS.

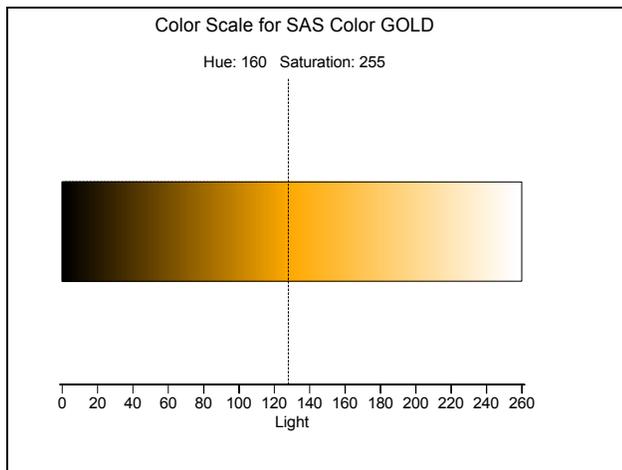


Figure 5. A color scale for GOLD. The vertical line identifies the LIGHT component of HLS code for the predefined SAS color.

All hexadecimal numbers in SAS are converted to decimal not only for display but because program looping is impossible otherwise. Thus for GOLD, H0A080FF in Figure 1 becomes D160128255 in Figure 5. The structure for the HLS code is identified below:

HLS (Hue, Light Saturation) Code		
Code	Hexadecimal	Decimal
Code	H0A080FF	D160128255
Identifying Prefix	H	D
Hue (hhh) (ddd)	0A0	160
Light (LL) (LLL)	80	128
Saturation (SS) (SSS)	FF	255

Numeric conversion macros play an important role in working with SAS colors. The programmer works and thinks in decimal, but the programs must convert to both hexadecimal and decimal when needed. This conversion and re-conversion process is apparent in the following code fragment from the PLOTCOLSC macro:

```
%macro PltColSc(SASCol);
proc sql noprint;
  select hls, cDesc into &hls, :cDesc
  from rgb.SASColors
  where cName eq "%upcase(&SASCol)";
quit;
%let hlsdec=&hlsdec(&hls);
%let DHue=%substr(&hlsdec,2,3);
%let DLum=%substr(&hlsdec,5,3);
%let DSat=%substr(&hlsdec,8,3);
%macro scale;
data anno;
  %system(2,2,2);
  %do i=&DLum %to 255 %by 5;
    ylow=10; yhigh=20; xleft=&i;
    xright=xleft+5;
    %bar(xleft,ylow,xright,yhigh,
    &hlsdec(&DHue,&i,&DSat),0,solid);
  %end;
endmacro;
```

```
output;
%end;
etc. etc. etc. ...
run;
%mend scale;
%scale;
etc. etc. etc. ...
%mend PltColSc;
%PltColSc(GOLD);
```

- ① HLS here is a macro variable containing the hexadecimal value for HLS that corresponds to SAS color sent as a parameter to the macro.
- ② HLSDEC is the decimal equivalent for HLS. This value is not used directly. Rather, it is parsed into decimal values for HUE, LIGHT, and SATURATION.
- ③ HUE and SATURATION remain fixed, but LIGHT is varied from 0 to 255, minimum and maximum values, in increments of 5 (visible as slight ridges in Figure 5). Decimal notation is required here so that the looping statement will work properly. Increments of 5 are chosen, because there is a danger of exceeding the value of MAXCOLS for a given graphics device. When the limit is exceeded, SAS just supplies default colors in random order thereby destroying the color scale that is being created.
- ④ The decimal number must be converted back to hexadecimal so that SAS can plot the correct color for a given increment of the looping variable.

Working with the RGB Code

If only a single color needs to be defined for an application, then RGB is the code of choice, since it is configured identically in both SAS and Microsoft. In this instance, the choice of a color does not need to be confined to a predefined list. Instead, a color can be defined by simply manipulating the controls in the *Custom* color window of a Microsoft application. Once the color is selected, its RGB code must be recorded for processing by the RGBHEX macro inside SAS.

The RGBHEX macro converts Microsoft decimal numbers to hexadecimal notation that SAS software understands. The macro contains only one statement:

```
%macro RGBHex(rrr,ggg,bbb);
  %sysfunc(compress(CX%sysfunc(putn(&rrr,hex2.))
  %sysfunc(putn(&ggg,hex2.))
  %sysfunc(putn(&bbb,hex2.)))));
%mend RGBHex;
```

The SAS function COMPRESS is used rather than the macro function %COMPRESS to remove all internal blanks from the returned hexadecimal color code. For example, %RGBHEX(0,204,153) returns CX00CC99. If %COMPRESS were used instead, the returned string would be unusable as CX00 CC 99. Note that the macro statement is missing a semicolon. If one were present, it would be returned as a portion of the string constant causing an error.

Gray is unique

The color gray is automatically produced if the three components in the RGB code have the same value. Consequently, the conversion macro GRAYHEX is a simplified version of RGBHEX:

```
%macro GrayHex(gray);
  %sysfunc(compress(CX%sysfunc(putn(&gray,hex2.))
    %sysfunc(putn(&gray,hex2.))
    %sysfunc(putn(&gray,hex2.))))
%mend GrayHex;
```

Because of this property, the RGB scale can be used to construct a color scale for gray. Here is an adjusted version of the SCALE macro that works with RGB rather than HLS to define the gray scale displayed in Figure 6:

```
%macro scale;
  data anno;
  %system(2,2,2);
  %do i=0 %to 255 %by 5;
    ylow=10; yhigh=20; xleft=&i; xright=xleft+5;
    %bar(xleft,ylow,xright,yhigh,%GrayHex(&i),
      0,solid);
  output;
  %end;
  %bar(0,10,260,20,black,0,empty); /*outline*/
  stop;
  run;
%mend scale;
```

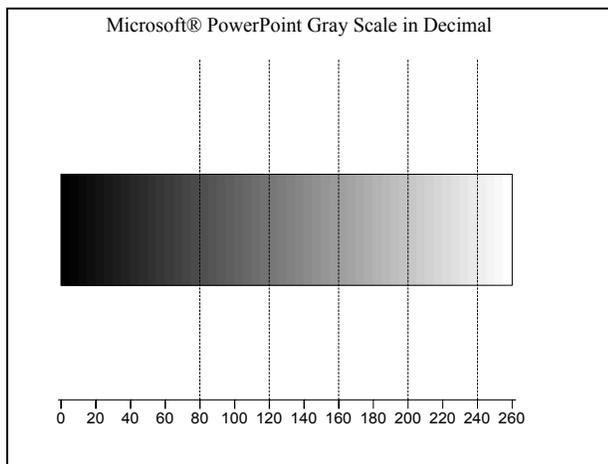


Figure 6. A gray scale based on the more intuitive decimal notation enables a developer to define a precise shade for an application.

A Table of Conversion Macros

Since the conversion macros are typically brief, they are being grouped together in Table 1 below. If the source code has already been listed in the paper, a reference is made to it.

Name	GRAYHEX
Description	Convert an RGB decimal code into Hexadecimal format.
Parms	GRAY, a 3-digit zero-filled decimal number.
Source Code	See section <i>Gray is Unique</i>
Name	HLSDEC
Description	Convert a SAS HEX HLS color to its decimal equivalent.
Parms	HHHLLSS a hexadecimal number with a leading character H.
Source Code	
<pre>%macro HLSDec(Hhhllss); %let hue=%substr(&Hhhllss,2,3);</pre>	

<pre>%let ll=%substr(&Hhhllss,5,2); %let ss=%substr(&Hhhllss,7); %sysfunc(compress (D%sysfunc(putn(%sysfunc(inputn(&hue,hex3.)),z3.) %sysfunc(putn(%sysfunc(inputn(&ll,hex2.)),z3.) %sysfunc(putn(%sysfunc(inputn(&ss,hex2.)),z3.)))))) %end HLSDec;</pre>	
Name	HLSHEX
Description	Convert an HLS decimal code into Hexadecimal format.
Parms	HUE: a 3-digit code for hue LUM: a 3-digit code for light SAT: a 3-digit code for saturation
Source Code	
<pre>%macro HLSHex(hue,lum,sat); %sysfunc(compress (H%sysfunc(putn(&hhh,hex3.)) %sysfunc(putn(&ll,hex2.)) %sysfunc(putn(&ss,hex2.)))) %end HLSHex;</pre>	
Name	RGBHEX
Description	Convert an RGB decimal code into Hexadecimal format.
Parms	RRR: a 3-digit code for Red GGG: a 3digit code for Green BBB: a 3-digit code for Blue
Source Code	See section <i>Working with the RGB Code</i>

Table 1. A listing of all the conversion macros used for processing and defining SAS colors.

Applications

• *Matching SAS/GRAPH and PowerPoint backgrounds*

The presence of too many colors can lead to “garish, difficult to interpret and confusing interfaces” according to Preece (89). The potential for clashing colors is very great when the output from SAS/GRAPH is pasted into a PowerPoint Presentation. By matching the PowerPoint background color in SAS, it is easy to produce the color-coordinated tree graph in Figure 7.

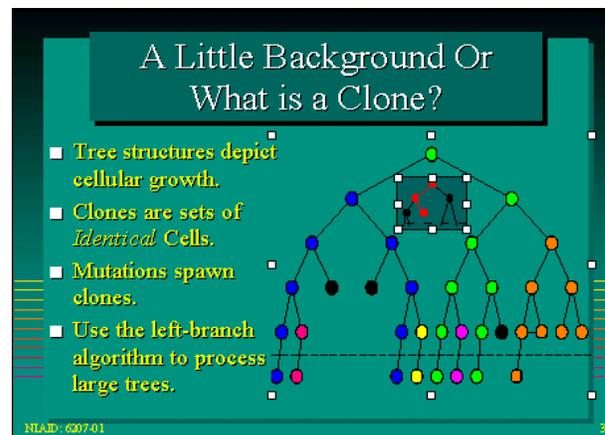


Figure 7. Binary tree plots with white selection handles match background colors in PowerPoint. The decimal RGB code is copied from PowerPoint and then matched in SAS by adding CBACK="&rgbhex(0,150,136)" to the GOPTIONS statement.

•Producing a bar chart where the shades of gray can be distinguished from each other

By referencing the gray scale in Figure 6, the bar charts in Figure 8 that depict relative speeds of format resolution times in graduated shades of gray can be produced with the following source code:

```
select (srctype);
  when ('S') color="%GRAYHEX(220)";
  when ('SE') color="%GRAYHEX(160)";
  when ('BE') color="%GRAYHEX(80)";
  when ('B') color='black';
end;
```

With decimal notation, it is easy to remember that low numbers are dark and high numbers are light.

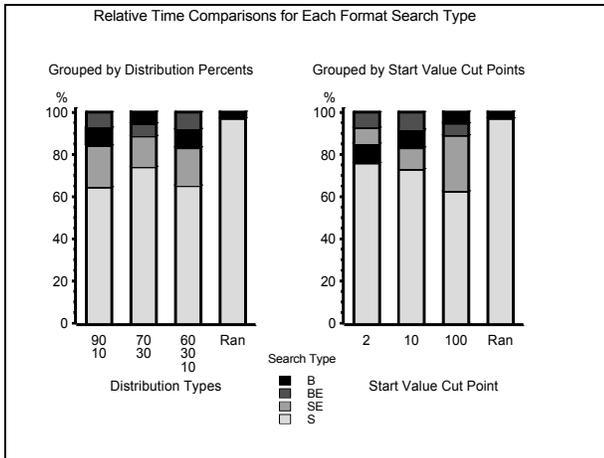


Figure 8. Relative resolution times for B(inary), B(inary) (E)mbedded, (S)equential (E)mbedded, and (S)equential formats in ranked order. A discussion of the issues that motivated this graph can be found in the paper about the relationship between format structure and efficiency cited in the reference section.

•Producing a visually appealing, more informative contour plot

Preece's quote about too many colors is particularly appropriate when SAS default colors are used for generating contour plots. Two versions of the same contour plot are displayed in Figures 9 and 10 at the end of the paper. The source code and data come from the SAS Sample Library.

A simple loop in the source code is used to plot the contours in varying shades of blue and brown. Since the colors are related to each other, the HLS scale must be used with a color obtained from the annotated list of predefined SAS colors. This particular version of the list also lists HLS codes in decimal. Again the decimal number is converted to hexadecimal inside the loop so that the correct SAS color is assigned. The relevant code fragment appears below:

```
/* color the contour plot */
%macro c19;
```

```
%let j=1;
%do i=80 %to 190 %by 10; *water;
  pattern&j value=msolid
    color=%hlsheX(001,&i,145) ;
  %let j=%eval(&j+1);
%end;
%do i=180 %to 60 %by -20; *land;
  pattern&j value=msolid
    color=%hlsheX(152,&i,045) ;
  %let j=%eval(&j+1);
%end;
%mend c19;
```

Summary and Conclusions

Two visual aids, the color chart and color scale, have been presented in this paper. The color chart is used to enhance the SAS list of predefined colors by associating an actual color with its NAME, HLS and RGB code. The color scale developed from the color chart displays one hue in varying degrees of luminescence. The color scale is particularly helpful for designing graphs that depict magnitude through a meaningful arrangement of related colors. Finally, here is the actual color associated with VPAPB - hardcopy in a shade of gray.

VPAPB	very pale purplish blue	CXC0A8E5
-------	-------------------------	----------

References

- Haworth, Lauren E., *Output Delivery System: The Basics*. Cary, NC: SAS Institute Inc., 2001.
- Preece, Jenny et. al. *Human-Computer Interaction*. Harlow, England: Addison-Wesley, 1994.
- SAS Institute Inc. *SAS Online Doc*. Cary, NC: SAS Institute Inc., 1999.
- SAS Institute Inc. *SAS/GRAPH® Software: Reference, Version 8, Volume 1*. Cary, NC: SAS Institute Inc., 1999.
- SAS Sample Library. Name: GR2401. Title: *Simple Contour Plot*. Product: GRAPH. System: ALL.
- Watts, Perry. *On the Relationship between Format Structure and Efficiency in SAS*. Proceedings of the 14th Annual NorthEast SAS® Users Group Conference. Baltimore, MD. September 30-October 3, 2001, pp. 697-705.

Contact Information

Perry Watts, Independent Consultant
wattsp@dca.net

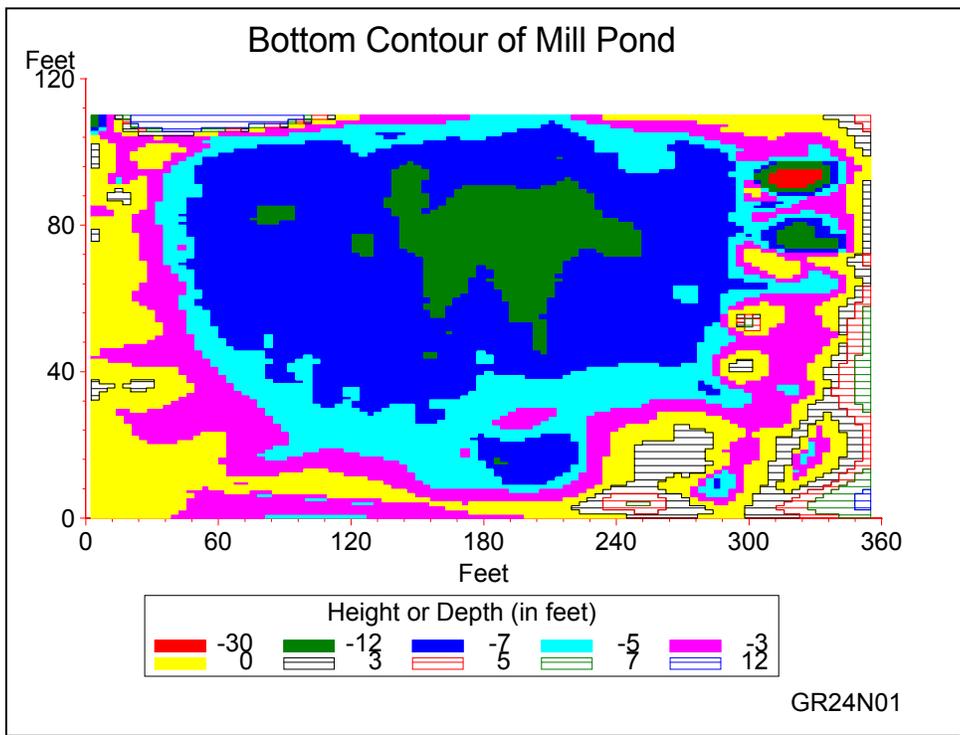


Figure 9. First rendition of Mill Pond is garish in full color. There is no rationale behind coloring the deepest part of the pond red, and the shallowest part yellow. Some study is also required to realize that dashed lines are used for depicting dry land.

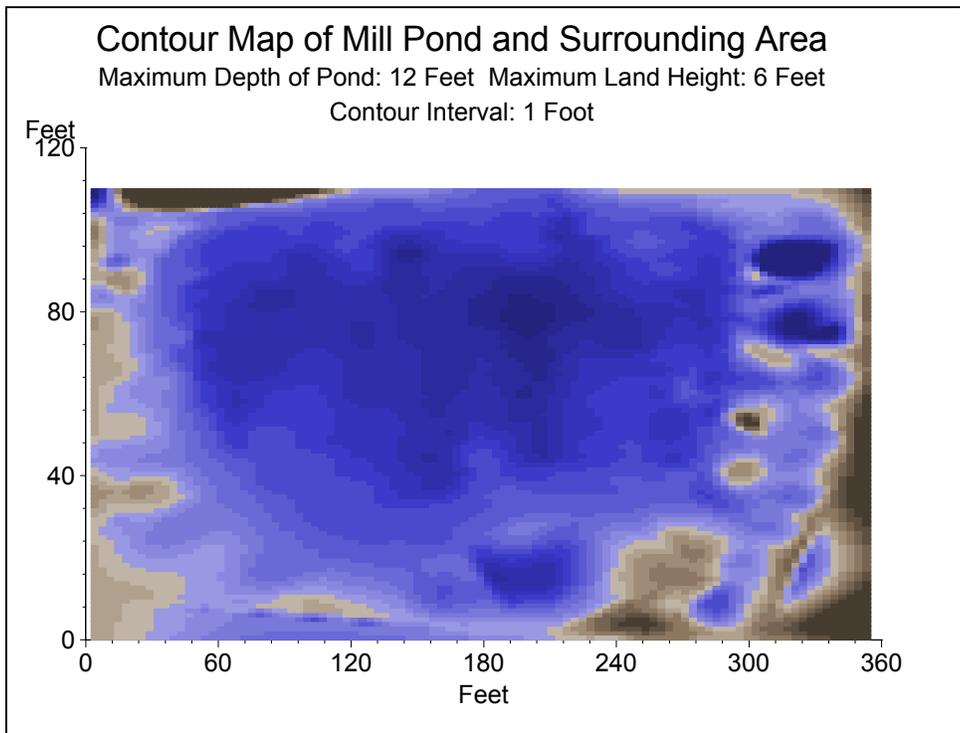


Figure 10. In the revised contour plot, water depth is depicted in varying shades of blue and land height takes on darker shades of brown at higher elevations. There is no need for a legend, since all communication is conveyed in the title.