

Stop Light Reporting with PROC REPORT

Ron Richardson, Regions Bank, Birmingham, AL

ABSTRACT

Frequently in banking users want "red/yellow/green" colors in matrix reports to aid in illustrating to management the risks in certain decisions. Using **proc report** coupled with custom compute blocks and formats, SAS® can deliver the presentation quality information normally associated with WISWIG spreadsheet output. The key is layering multiple SAS techniques to achieve the desire result.

INTRODUCTION

This paper will illustrate how to employ various SAS techniques to produce a Stop Light style report. We will touch on the reporting dataset layout. Then we will review the three SAS techniques, when employed together, support the stop light report. The techniques are; custom formats, ODS styles & output, **proc report** statements and compute blocks. Each of these topics is covered in the following sections.

THE REPORTING DATASET

For the process to work, the reporting dataset must contain the results of all the summary calculations. That is, the values in each row of the matrix report must be present in the dataset before beginning. This may require considerable dataset manipulation. However, it should not be outside the abilities of most intermediate SAS programmers. Why go through all the heroics? Because **proc report** is the tool that supports the conditional compute blocks required to make this process work. Note that missing columns are handled in **proc report** with the custom format technique below. Missing rows must be corrected / filled-in using a dataset or SQL procedure. I like to employ a lookup table to insert any needed observations. The code below produces a Stop Light report ready dataset.

```
data work.Summary_Report_Data;
  input H_Axis:$1. V_Axis:$1. Num_Loans:8.;
  datalines;
2 A 1
3 A 1
T A 2
2 B 2
3 B 4
T B 6
2 C 3
3 C 5
T C 8
2 T 6
3 T 10
T T 16
;
run;
```

THE CUSTOM FORMAT

The custom format plays a vital role in the output. Leveraging **proc report's** format pre-load function, this custom format ensures that every column in the matrix is displayed regardless of the data being reported. This keeps your report framework consistent.

```
proc format lib=work;
  value $H_Axis_Fmt
    '1' = '1'
    '2' = '2'
    '3' = '3'
  ;
run;
```

ODS CONTROLS

Exploiting the properties of the output delivery system (ODS) is fundamental to producing the Stop Light report. The main ODS feature used is the **style** function which produces the red, yellow and green cells in the report.

Additionally, by leveraging an existing template we can control various style elements of the report's framework (i.e. the frame and space around the cell values). The ODS PATH statement below initializes the style search path. The **proc template** routine below that takes an existing style and uses it as a basis for the new style to be utilized in the report.

```
ods path (prepend) work.template(update);

proc template;
  define style work.mystyle;
    parent=styles.printer
    ;
    /** control the system titles */
    style systemtitle from titlesandfooters /
      font_face    = arial
      font_style   = roman
      font_weight  = bold
    ;

    /** control row headers */
    style header from header /
      background    = grey
      foreground    = white
      font_face     = arial
      font_size     = 8pt
      cellpadding  = 2pt
    ;

    /** control left column */
    style data from cell /
      background    = grey
      foreground    = white
      font_face     = arial
      font_size     = 8pt
    ;
  end;
run;
```

Additional ODS options are specified below. Note the creation of the "ODS ESCAPECHAR" and its usage in the **title** statement below. This technique supports insertion of special graphic files into the report lending a more professional look.

```
/** stop the normal output listing */
ods listing close;

/** redirect it to a PDF file using a certain style template */
ods pdf file = "c:\Stop_Light_Report.pdf" style=work.mystyle;

/** specify a certain character to support special formatting */
ods escapechar = '^';

/** creates custom PDF bookmarks */
ods proclabel = "Volume Dist";

title1 j=l'^S={preimage="C:\saslogo.gif"}'
      j=c height = 18pt "SESUG 2009"
      ;
```

THE REPORT PROCEDURE

The three major features of **proc report** being leveraged to produce the Stop Light report are, **column spanning**, **format pre-loading & compute blocks to change ODS styles**. Rather than present a detailed narrative of each part of the procedure, I have provided comments throughout the program. Hopefully, the placement of remarks near the code will prove more instructive.

```
proc report
  data = work.Summary_Report_Data
  /* options */
  contents="" /* suppress normal PDF bookmark */
  nowindows  /* suppress normal report window */
;

/** Column spanning and headers used to describe data. */
/** Note the comma between variables forces proc report */
/** to display contents of the variable. */
column
  ("^S={font_size=12pt}Vertical" V_Axis)
  ("^S={font_size=12pt}Horizontal" (H_Axis, Num_Loans))
  Cell_Color_Alias /** this is part of the color driver */
;

/* group rows by V_Axis values, centers & suppress col header*/
define V_Axis / group center "";

/* group columns by H_Axis, pre-loads the custom format */
define H_Axis / across format = $H_Axis_Fmt. preloadfmt center "";

/* analysis keyword works in conjunction with column statement */
define Num_Loans / analysis center "";

/* computed value support colors in cells */
define Cell_Color_Alias / noprint;

/** This compute block drives the stoplight report colors. */
/** NOTE: the columns referenced in the first position of the */
/** "call define" statement begin with 1st report column (1). */

compute Cell_Color_Alias;

/** when the value of v_axis is "a", paint the cells in columns... */
if lowercase(V_Axis) = 'a' then do;

  /** for columns 2 & 3, red/white, for column 4, yellow/black */
  call define(2,'style', 'style=[background=red foreground=white]');
  call define(3,'style', 'style=[background=red foreground=white]');
  call define(4,'style', 'style=[background=lightyellow foreground=black]');
end;

if lowercase(V_Axis) = 'b' then do;
  call define(2,'style', 'style=[background=red foreground=white]');
  call define(3,'style', 'style=[background=lightyellow foreground=black]');
  call define(4,'style', 'style=[background=lightgreen foreground=black]');
end;

if lowercase(V_Axis) = 'c' then do;
  call define(2,'style', 'style=[background=lightyellow foreground=black]');
  call define(3,'style', 'style=[background=lightgreen foreground=black]');
  call define(4,'style', 'style=[background=lightgreen foreground=black]');
end;

endcomp;
run;
```

```
ods pdf close;  
ods listing;  
title1;  
title2;
```

CONCLUSION

Although a challenging variety of techniques are required, the program above illustrates that SAS can produce presentation quality output that rivals the spreadsheets. The custom format coupled with **proc report's** format pre-load feature ensures a consistent horizontal axis. Exploiting the output delivery system's support of styles and colors in conjunction with the **reporting procedure's** ability to create and conditionally modify the display attributes for a variable are the drivers for painting the cell's colors.

REFERENCES

Funny ^Stuff~ in My Code: Using the ODS ESCAPECHAR. SAS Global Forum paper 099-2007. Cynthia L. Zender, SAS Institute Inc., Cary, NC

ACKNOWLEDGMENTS

SAS technical support provided considerable expertise in turning this challenge into a solution.

CONTACT INFORMATION

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

Ron Richardson
Regions Bank
2050 parkway Office Circle
Hoover, AL 35244
Work Phone:
E-mail: ron.richardson@regions.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

#