Building a Better Dashboard Using Base SAS® Software

Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California
Joshua M. Horstman, Nested Loop Consulting, Indianapolis, Indiana
Roger D. Muller, Data-To-Events, Indianapolis, Indiana

Abstract
Organizations around the world develop business intelligence dashboards, sometimes referred to as enterprise dashboards, to display the current status of “point-in-time” metrics and key performance indicators. Effectively designed dashboards extract real-time data from multiple sources for the purpose of highlighting important information, numbers, tables, statistics, metrics, performance scorecards and other essential content on a single screen. This presentation explores the basic rules for “good” dashboard design, the metrics frequently used in dashboards, and the use of best practice programming techniques in the design of highly interactive, filterable, and drill-down dashboards using Base-SAS® software. Attendees learn how to create a real-world dashboard using Base-SAS® programming techniques including the use of the DATA step, PROC FORMAT, PROC PRINT, PROC MEANS, PROC SQL, Enterprise Guide, ODS, ODS Statistical Graphics, PROC SGRENDER, PROC SGPLOT, PROC SGSCATTER, PROC SGPANEL, and PROC TEMPLATE.

Introduction
In a world of big data where data repositories and the demand placed on them are growing at explosive levels, organizations are faced with a number of decisions related to their information requirements:

1) What are the best ways to handle large amounts of information?
2) How should analytical data be processed?
3) What are the choices for constructing the most effective information delivery mechanisms?
4) How should analytical data and results be displayed?

To help answer these and other questions, this paper explains what a dashboard is, the dashboard’s elements, the do’s and don’ts for constructing effective dashboards, dashboard design techniques, an investigation of the various types of dashboards, the merits and strengths of using the base-SAS® software to construct dashboards, and an illustration of a few dashboard examples along with the base-SAS code used in their construction.

Example Table
The dashboard examples displayed in this paper reference a Movies table consisting of a number of movies that I’ve viewed over the years. The Movies table consists of six columns: Title, Length, Category, Year, Studio, and Rating. Title, Category, Studio, and Rating are defined as character columns, and the Length and Year are defined as numeric columns. The data contained in the Movies table is illustrated below.

<table>
<thead>
<tr>
<th>Title</th>
<th>Length</th>
<th>Category</th>
<th>Year</th>
<th>Studio</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brave Heart</td>
<td>177</td>
<td>Action Adventure</td>
<td>1995</td>
<td>Paramount Pictures</td>
<td>R</td>
</tr>
<tr>
<td>Casablanca</td>
<td>102</td>
<td>Drama</td>
<td>1942</td>
<td>MGM / UA</td>
<td>F</td>
</tr>
<tr>
<td>Christmas Vacation</td>
<td>97</td>
<td>Comedy</td>
<td>1990</td>
<td>Warner Bros</td>
<td>F8-13</td>
</tr>
<tr>
<td>Coming to America</td>
<td>116</td>
<td>Comedy</td>
<td>1988</td>
<td>Paramount Pictures</td>
<td>R</td>
</tr>
<tr>
<td>Dracula</td>
<td>130</td>
<td>Horror</td>
<td>1992</td>
<td>Columbia Tri-Star</td>
<td>R</td>
</tr>
<tr>
<td>Die Hard</td>
<td>110</td>
<td>Drama</td>
<td>1990</td>
<td>Paramount Pictures</td>
<td>R</td>
</tr>
<tr>
<td>Forrest Gump</td>
<td>142</td>
<td>Drama</td>
<td>1994</td>
<td>Paramount Pictures</td>
<td>F8-13</td>
</tr>
<tr>
<td>Ghost</td>
<td>127</td>
<td>Drama Romance</td>
<td>1990</td>
<td>Paramount Pictures</td>
<td>F8-13</td>
</tr>
<tr>
<td>jaws</td>
<td>135</td>
<td>Action Adventure</td>
<td>1977</td>
<td>Universal Studios</td>
<td>F9</td>
</tr>
<tr>
<td>Jurassic Park</td>
<td>127</td>
<td>Action</td>
<td>1993</td>
<td>Universal Pictures</td>
<td>F8-13</td>
</tr>
<tr>
<td>Michael</td>
<td>100</td>
<td>Drama</td>
<td>1997</td>
<td>Warner Bros</td>
<td>F8-13</td>
</tr>
<tr>
<td>National Lampoons Vacation</td>
<td>98</td>
<td>Comedy</td>
<td>1980</td>
<td>Warner Bros</td>
<td>F8-13</td>
</tr>
<tr>
<td>Peter Pan</td>
<td>115</td>
<td>Horror</td>
<td>1992</td>
<td>MGM / UA</td>
<td>F8</td>
</tr>
<tr>
<td>Rocky</td>
<td>130</td>
<td>Action Adventure</td>
<td>1976</td>
<td>MGM / UA</td>
<td>F8</td>
</tr>
<tr>
<td>Scroog</td>
<td>120</td>
<td>Action Crime &amp;</td>
<td>1985</td>
<td>Universal Studios</td>
<td>R</td>
</tr>
<tr>
<td>Silence of the Lambs</td>
<td>110</td>
<td>Drama Suspense</td>
<td>1991</td>
<td>Orion</td>
<td>R</td>
</tr>
<tr>
<td>Shaolin</td>
<td>124</td>
<td>Action Sci-Fi</td>
<td>1997</td>
<td>Lucas Film Ltd</td>
<td>F9</td>
</tr>
<tr>
<td>The Hunt for Red October</td>
<td>125</td>
<td>Action Adventure</td>
<td>1995</td>
<td>Paramount Pictures</td>
<td>F8</td>
</tr>
<tr>
<td>The Wizard of Oz</td>
<td>101</td>
<td>Adventure</td>
<td>1935</td>
<td>MGM / UA</td>
<td>G</td>
</tr>
<tr>
<td>Titanic</td>
<td>194</td>
<td>Drama Romance</td>
<td>1997</td>
<td>Paramount Pictures</td>
<td>F8-13</td>
</tr>
</tbody>
</table>
“Brief” History of Dashboards
In the world of information technology, a dashboard serves as a user interface to organize and display information visually in the simplest way possible. Dashboards originated in the 1970’s as decision support tools and systems that served management, operations, and organizational planning. In the 1980’s, dashboards came of age as executive information systems emphasizing graphical displays and simple user interfaces to assist with management decision making. In the 1990’s, dashboards experienced a growing interest with the rise of the Internet. As information technology and the Internet entered the 2000’s, vendors including SAS Institute, and others, offered high-end easy-to-use products for the development of comprehensive “custom” dashboards. The dashboards being built today offer users the ability to monitor key metrics, information summaries, and reports in a single easy-to-use user interface. As a result, dashboards are designed to alert users to key business issues that impact an organization’s tactics and strategies by facilitating improved decision making activities.

So exactly what is a dashboard? In the paper, “Building Your First Dashboard Using the SAS® 9 Business Intelligence Platform: A Tutorial,” by Gregory S. Nelson (2009), Nelson describes a dashboard as a visualization technique that provides an immediate view or snapshot of exactly where you are in a specific process relative to your stated goals and objectives. He adds that, Visual indicators, such as temperature gauges, traffic lights and speedometers, help give a real-world sense of present progress and assists in making decisions, adapting to current conditions or drilling into more detailed information. As a user interface, dashboards display performance indicators (PIs), key performance indicators (KPIs), and other relevant information.

Types of Dashboards
The first step in dashboard design is to understand the purpose and type of dashboard you will need. With three types of dashboard designs available, users are encouraged to select the dashboard type that best meets your needs. The following table describes the three types of enterprise dashboards and their purpose.

<table>
<thead>
<tr>
<th>Dashboard Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategic Dashboards</td>
<td>Strategic dashboards provide executives and managers with visual information to determine and support goals and objectives within an organization. This type of dashboard facilitates monitoring an organization’s health, progress, performance, and areas where improvement can be made. There is typically no need for interactive features with this type of dashboard. Strategic dashboard examples include: Sales, Human Resources, Manufacturing, and Services.</td>
</tr>
<tr>
<td>Analytical Dashboards</td>
<td>Analytical dashboards provide users with visual information to help gain a better understanding with historical, present and future data; understand trends; allow comparisons to be made; and determine the type of adjustments that are needed. Analytical dashboards should allow interactive features such as drill-down capabilities, as needed, to access more detailed information. Dashboard examples include: obtaining real-time data and information, determining why some things are working and others are not, identifying patterns and opportunities with your data, and aligning strategic objectives with performance initiatives.</td>
</tr>
<tr>
<td>Operational Dashboards</td>
<td>Operational dashboards provide users with visual information to concentrate on performance monitoring and measurements, monitor the efficiency and effectiveness of their organization. There is typically a need to update information displayed in an operational dashboard frequently to make it relevant to the users’ needs. Dashboard examples include: improved understanding of performance, better focus and alignment, and faster and better decision making.</td>
</tr>
</tbody>
</table>

Dashboard Elements
In Malik Shadan’s (2007) paper, Elements for an Enterprise Dashboard, he mentions that there are basic and advanced characteristics specific to an enterprise dashboard. The basic characteristics encompass the acronym, SMART, and the advanced characteristics of an enterprise dashboard encompass the acronym, IMPACT. The elements associated with each acronym appear in the following tables.

SMART Basic Elements

<table>
<thead>
<tr>
<th>Synergetic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synergetic</td>
<td>Synergize information in a single screen view.</td>
</tr>
<tr>
<td>Monitor KPIs</td>
<td>Display critical KPIs for effective decision making.</td>
</tr>
<tr>
<td>Accurate</td>
<td>Dashboard must be well tested and validated, and information must be accurate.</td>
</tr>
<tr>
<td>Responsive</td>
<td>Respond to user alerts and visual content to draw immediate attention to critical matters.</td>
</tr>
<tr>
<td>Timely</td>
<td>Display information that is real-time and right-time for effective decision making.</td>
</tr>
</tbody>
</table>
Building a Better Dashboard Using Base SAS® Software, continued

13 Common Pitfalls to Avoid when Designing Dashboards

Successful dashboard design involves the transformation of quantitative data into meaningful and effective visual displays including graphs, maps, gauges and summary information. In his paper, “Common Pitfalls in Dashboard Design,” Stephen Few (2006) proposes 13 common mistakes many make when designing dashboards. Instead of concentrating on what should be done when designing dashboards, Mr. Few’s body of work espouses the most common mistakes along with detailed explanations to help educate current and future designers alike. I have listed the 13 common pitfalls from Mr. Few’s seminal work, below, but readers are encouraged to read his entire paper, see the References section, for a complete perspective.

Stephen Few’s 13 Common Pitfalls in Dashboard Design (cited from reference)

<table>
<thead>
<tr>
<th>Pitfall</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitfall #1</td>
<td>Exceeding the Boundaries of a Single Screen</td>
</tr>
<tr>
<td>Pitfall #2</td>
<td>Supplying Inadequate Context for the Data</td>
</tr>
<tr>
<td>Pitfall #3</td>
<td>Displaying Excessive Detail or Precision</td>
</tr>
<tr>
<td>Pitfall #4</td>
<td>Expressing Measures Indirectly</td>
</tr>
<tr>
<td>Pitfall #5</td>
<td>Choosing Inappropriate Media of Display</td>
</tr>
<tr>
<td>Pitfall #6</td>
<td>Introducing Meaningless Variety</td>
</tr>
<tr>
<td>Pitfall #7</td>
<td>Using Poorly Designed Display Media</td>
</tr>
<tr>
<td>Pitfall #8</td>
<td>Encoding Quantitative Data Inaccurately</td>
</tr>
<tr>
<td>Pitfall #9</td>
<td>Arranging the Data Poorly</td>
</tr>
<tr>
<td>Pitfall #10</td>
<td>Ineffectively Highlighting What’s Important</td>
</tr>
<tr>
<td>Pitfall #11</td>
<td>Cluttering the Screen with Useless Decoration</td>
</tr>
<tr>
<td>Pitfall #12</td>
<td>Misusing or Overusing Color</td>
</tr>
<tr>
<td>Pitfall #13</td>
<td>Designing an Unappealing Visual Display</td>
</tr>
</tbody>
</table>

Steps to Creating a Dynamic Dashboard using Base-SAS®

Follow these basic steps to successfully construct a dynamic dashboard using the Base-SAS software.

1. Connect to desired data sources using Libname statement.
2. Create user-defined formats containing URL links for dashboard and drill-down results.
3. Create Graphics, Bar Chart, Box Plot, Histogram, Pie, etc.
4. Produce Graph Template using PROC SGRENDER as a static or dynamic display.
5. Produce Drill-down Detail Reports, Summary Reports, Statistics, Tables, etc.

At the end of this paper are several examples of static and drill-down (dynamic) dashboards.
**Dashboard Examples**

1. **Static Dashboard – PROC PRINT Dashboard using PROC FORMAT**

   ![Distribution of Length](image)

   **Key Points about Code**

   1. Base SAS now provides users with several procedures for creating graphical output. The three procedures are: SGPLOT, SGSCATTER and SGPANEL.
   2. The type of dashboard created in this approach is a static, or non-drill-down, dashboard.
   3. A template definition, created with PROC TEMPLATE, provides details to define the chart including the chart title, the layout of an overlay scenario, the creation of a histogram chart, the specification of the “crisp” dataskin, and the specification of a true border.
   4. The specification of the ODS HTML and GRAPHICS statements to define the dashboard’s dimensions.
   5. PROC SGRENDER is used to produce a graph from an input SAS data set and an ODS graph template.

   **Base-SAS Code:**

   ```sas
   **************************************************************************
   ***** Program Name: Dashboard with Histogram and PROC SGRENDER.SAS *****;
   ***** Purpose.....: Create and display a simple dashboard interface using PROC *****;
   *****               TEMPLATE and PROC SGRENDER.                                    *****;
   ***** Author......: Kirk Paul Lafler, Software Intelligence Corporation *****;
   ***** Date Written: 02/22/2015 ***;
   ***** Input Files.: Workshop Data *****;
   ***** Output Files: HTML Output (1 File) *****;
   ***** Subroutines.: None *****;
   ***** User-defined Formats: None *****;
   ***** Macro Variables: None *****;
   ***** Includes....: None *****;
   ***** Modification History: 02/23/2015 KPL Added Header information.*****;
   **************************************************************************;
   ```
libname mydata 'e:\workshops\workshop data' ;

*STEP 1 - Create DashboardHistogram Template Overlay with PROC TEMPLATE ;
proc template ;
  define statgraph DashboardHistogram ;
  dynamic VAR ;
  begingraph ;
    entrytitle "Distribution of " VAR ;
    layout overlay / xaxisopts=(griddisplay=on) ;
    histogram VAR / scale=percent
dataSkin=crisp ;
    layout gridded / columns=2
      autoalign=(topleft topright) border=true opaque=true backgroundcolor=GraphWalls:color ;
      entry halign=left "N" ;
      entry halign=left eval(strip(put(n(VAR),12.0))) ;
      entry halign=left "Mean" ;
      entry halign=left eval(strip(put(mean(VAR),12.2))) ;
      entry halign=left "Std Dev" ;
      entry halign=left eval(strip(put(stddev(VAR),12.2))) ;
    endlayout ;
  end ;
endgraph ;
end ;
run ;

*STEP 2 - Produce Graphical Output from DashboardHistogram Template using PROC SGRENDER ;
ods html file='Dashboard-SGRender-Histogram.htm' path='e:\' (url=none) ;
ods graphics / reset imagemap=on width=10in height=6in imagename='Dashboard-SGRender-Histogram' ;
title 1 h=7 color=Blue "Classic Movies" ;
proc sgrender data=mydata.movies template=DashboardHistogram ;
dynamic VAR="Length" ;
run ;
quit ;
title ;
ods html close ;

2. Drill-down Dashboard – Bar Chart Dashboard

![BarChart of Movies Data Rating by Movie Length](image)

- **G-rated Movies**
  
<table>
<thead>
<tr>
<th>Title</th>
<th>Length</th>
<th>Category</th>
<th>Year</th>
<th>Studio</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Wizard of Oz</td>
<td>131</td>
<td>Adventure</td>
<td>1939</td>
<td>MGM/LA</td>
<td>G</td>
</tr>
</tbody>
</table>
  
  N = 1

- **PG-rated Movies**

<table>
<thead>
<tr>
<th>Title</th>
<th>Length</th>
<th>Category</th>
<th>Year</th>
<th>Studio</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>139</td>
<td>Drama</td>
<td>1942</td>
<td>RKO</td>
<td>PG</td>
</tr>
</tbody>
</table>
  
  
  N = 6

- **PG-13-rated Movies**

<table>
<thead>
<tr>
<th>Title</th>
<th>Length</th>
<th>Category</th>
<th>Year</th>
<th>Studio</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinatown</td>
<td>91</td>
<td>Crime</td>
<td>1922</td>
<td>Warner</td>
<td>PG-13</td>
</tr>
</tbody>
</table>
  
  
  N = 1

- **R-rated Movies**

<table>
<thead>
<tr>
<th>Title</th>
<th>Length</th>
<th>Category</th>
<th>Year</th>
<th>Studio</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Godfather</td>
<td>144</td>
<td>Crime</td>
<td>1972</td>
<td>Warner</td>
<td>R</td>
</tr>
</tbody>
</table>
  
  
  N = 5
Key Points about Code

1. The first step illustrates the data preparation process. In this example, the data is sorted in ascending order by the Movie Rating variable.

2. The second step assigns an HTML link, for drill-down purposes, to a URLLINK variable with a series of IF-THEN/ELSE statements.

3. The third step creates a template definition using PROC TEMPLATE by providing details to define the chart including the chart title, the layout of an overlay scenario, the creation of a vertical bar chart, the specification of the “crisp” dataskin, a transparency value of .3 to allow the vertical bar chart to show through the overlay, a vertical orientation, and the specification of the URL= parameter to allow drill-down capabilities.

4. The fourth step specifies the ODS HTML and GRAPHICS statements, the dimensions of the dashboard, title and footnote statements, and PROC SGRENDER to produce the graphical image from an input SAS data set and an ODS graph template.

5. The fifth step specifies the type of output to produce when the user clicks the vertical bar chart’s bars displayed on the dashboard.

Base-SAS Code:

```sas
libname mydata 'e:\workshops\workshop data';

proc sort data=mydata.movies out=work.sorted_movies; by rating; run;

data Classic_Movies;
length URLLINK $50;
max=200;
set work.sorted_movies;
if upcase(rating) = 'G' then URLLINK='e:\SGRender-BarChart-G-Movies.htm';
else if upcase(rating) = 'PG' then URLLINK='e:\SGRender-BarChart-PG-Movies.htm';
else if upcase(rating) = 'PG-13' then URLLINK='e:\SGRender-BarChart-PG13-Movies.htm';
else if upcase(rating) = 'PG-17' then URLLINK='e:\SGRender-BarChart-PG17-Movies.htm';
else if upcase(rating) = 'R' then URLLINK='e:\SGRender-BarChart-R-Movies.htm';
run;
```
else
  if upcase(rating) = 'R'
    then URLLINK='e:\SGRender-BarChart-R-Movies.htm';
run;

/*****************************/
/* STEP 3 - Create BarChart Template Overlay with PROC TEMPLATE */
/*****************************/
proc template;
  define statgraph BarChartDashboard;
  begingraph;
    entrytitle "Drill-down BarChart Dashboard";
    layout overlay / xaxisopts=(display=(label tickvalues line))
                   dataskin=crisp
                   datatransparency=0.3
                   orient=vertical
                   url=urllink
                 
      barchart x=Rating y=length /
                   data=Classic_Movies
                   datatransparency=0.3
                   orient=vertical;
      endlayout;
  endgraph;
run;

/*****************************/
/* Step 4 - Produce Graphical Output from BarChart Template using PROC SGRENDER */
/*****************************/
ods html file='SGRender-BarChart-Dashboard.htm' path='e:' url=none;
ods graphics / reset imagemap=on width=7in height=5in imagename='SGRender-BarChart-Dashboard';
title1 "BarChart of Movies Data";
title2 "Rating by Movie Length";
proc sgrender data=Classic_Movies template=BarChartDashboard;
run;
title;
ods html close;

/*****************************/
/* STEP 5 - Create Drill-down Output for Each Rating Group with PROC PRINT */
/*****************************/
ods html body="SGRender-BarChart-G-Movies.htm" path='e:';
PROC PRINT DATA=classic_movies(DROP=urllink max) NOOBS N;
  title "G-rated Movies";
  WHERE UPCASE(rating) = "G";
RUN;
ods html close;

ods html body="SGRender-BarChart-R-Movies.htm" path='e:';
PROC PRINT DATA=classic_movies(DROP=urllink max) NOOBS N;
  title "R-rated Movies";
  WHERE UPCASE(rating) = "R";
RUN;
ods html close;
3. Drill-down Dashboard – SGPanel Dashboard using PROC FORMAT

Key Points about Code

1. The first step illustrates any data preparation process including data manipulation techniques. In this example, a PROC FORMAT is specified to create a URL variable to be associated with the Movie Rating variable.

2. The second step assigns the HTML link, for drill-down purposes, to a URLLINK variable with the user-defined format created in step 1, above.

3. The third step specifies the ODS HTML and GRAPHICS statements, the dimensions of the dashboard, title and footnote statements, PROC SGPANEL to produce the graphical vertical bar chart from an input SAS data set, and the specification of the URL= parameter to allow drill-down capabilities.

4. The fourth step specifies the type of output to produce when the user clicks the vertical bar chart’s bars displayed on the dashboard.
LIBNAME mydata 'e:\workshops\workshop data' ;

*STEP 1 - Create User-defined Rating Format ;
PROC FORMAT ;
   VALUE $RATFMT$ 'G' = 'e:\SGPanel-1-Drill-down-G-Movies.htm'
     'PG' = 'e:\SGPanel-1-Drill-down-PG-Movies.htm'
     'PG-13' = 'e:\SGPanel-1-Drill-down-PG13-Movies.htm'
     'PG-17' = 'e:\SGPanel-1-Drill-down-PG17-Movies.htm'
     'R' = 'e:\SGPanel-1-Drill-down-R-Movies.htm' ;
RUN ;

*STEP 2 - Assign HTML Link to URLLINK Variable ;
data Classic_Movies ;
   set mydata.movies ;
   urllink = put(rating,$RATFMT$. ) ;
RUN ;

*STEP 3 - Create Drill-down HTML File with PROC SGPANEL and User-defined Format ;
ODS html file="SGPanel-Dashboard-1.htm" path='e:\' (url=none) ;
ods graphics / reset imagemap=on width=8in height=6in imagename='SGPanel-Dashboard-1' ;
TITLE1 h=2in color=blue "SGPanel Dashboard" ;
TITLE2 color=blue "by Movie Rating" ;
FOOTNOTE COLOR=blue "(Click any vertical bar to drill down.)" ;
OPTIONS NODATE ;
proc sgpghanlax =Classic_Movies ;
   panelby category / onepanel uniscale=row ;
   vbar rating / url=urllink ;
RUN ;
ODS html close ;

*STEP 4 - Create Drill-down Output for Each Rating Group with PROC PRINT ;
TITLE ;
FOOTNOTE ;
ODS html body="e:\SGPanel-1-Drill-down-G-Movies.htm" ;
PROC PRINT DATA=Classic_Movies(DROP=urllink) NOOBS N ;
   TITLE "G-rated Movies" ;
   WHERE UPCASE(rating) = "G" ;
RUN ;
ODS html close ;

ODS html body="e:\SGPanel-1-Drill-down-PG-Movies.htm" ;
PROC PRINT DATA=Classic_Movies(DROP=urllink) NOOBS N ;
   TITLE "PG-rated Movies" ;
   WHERE UPCASE(rating) = "PG" ;
RUN ;
4. **Drill-down Dashboard – SGPanel Dashboard with HBAR using PROC FORMAT**

Key Points about Code

1. The first step illustrates any data preparation process including data manipulation techniques. In this example, a PROC FORMAT is specified to create a URL variable to be associated with the Movie Rating variable.

2. The second step illustrates using another PROC FORMAT to create a URL variable to be associated with the Movie Category variable.

3. The third step assigns the HTML links, for drill-down purposes, to the URLLINK variables created with the user-defined formats in step 1 and 2, above.
4. The fifth step specifies the type of output to produce when the user clicks the horizontal bar chart’s bars displayed on the dashboard.

**Base-SAS Code:**

```sas
LIBNAME mydata 'e:\workshops\workshop data';

*STEP 1 - Create User-defined Rating Format ;
PROC FORMAT ;
VALUE $RATFMT
   'G'  = 'e:\SGPanel-2-Drill-down-G-Movies.htm'
   'PG' = 'e:\SGPanel-2-Drill-down-PG-Movies.htm'
   'PG-13' = 'e:\SGPanel-2-Drill-down-PG13-Movies.htm'
   'PG-17' = 'e:\SGPanel-2-Drill-down-PG17-Movies.htm'
   'R' = 'e:\SGPanel-2-Drill-down-R-Movies.htm';
RUN ;

*STEP 2 - Create User-defined Category Format ;
PROC FORMAT ;
VALUE $CATFMT
   'Action' = 'e:\SGPanel-2-Drill-down-Act-Movies.htm'
   'Action Adventure' = 'e:\SGPanel-2-Drill-down-AA-Movies.htm'
   'Action Cops & Robber' = 'e:\SGPanel-2-Drill-down-ACR-Movies.htm'
   'Action Sci-Fi' = 'e:\SGPanel-2-Drill-down-ASF-Movies.htm'
   'Adventure' = 'e:\SGPanel-2-Drill-down-Adv-Movies.htm'
   'Comedy' = 'e:\SGPanel-2-Drill-down-C-Movies.htm'
   'Drama' = 'e:\SGPanel-2-Drill-down-D-Movies.htm'
   'Drama Mysteries' = 'e:\SGPanel-2-Drill-down-DM-Movies.htm'
   'Drama Romance' = 'e:\SGPanel-2-Drill-down-DR-Movies.htm'
   'Drama Suspense' = 'e:\SGPanel-2-Drill-down-Ds-Movies.htm'
   'Horror' = 'e:\SGPanel-2-Drill-down-H-Movies.htm';
RUN ;

*STEP 3 - Assign HTML Link to URLLINK Variable ;
data Classic_Movies ;
   set mydata.movies ;
   urllink1 = put(rating,$RATFMT.);
   urllink2 = put(category,$CATFMT.);
run ;

*STEP 4 - Create Drill-down HTML File with PROC SGPANEL and User-defined Format ;
ODS html file="SGPanel-Dashboard-2.htm" path='e:\' (url=none);
ods graphics / reset imagemap-on width=8in height=6in imagename='SGPanel-Dashboard-2' ;
TITLE1 h=.2in color=blue "SGPanel Dashboard" ;
TITLE2 color=blue "by Movie Rating and Movie Category" ;
FOOTNOTE COLOR=blue "(Click any vertical/horizontal bar to drill down.)" ;
OPTIONS NODATE ;
proc sgpanel data=Classic_Movies ;
   panelby rating category / layout=panel onepanel novarname columns=5 ;
hbar category / url=urllink2 ;
run ;
ods html close ;

*STEP 5 - Create Drill-down Output for Each Rating Group with PROC PRINT ;
TITLE ;
FOOTNOTE ;
ods html body="e:\SGPanel-2-Drill-down-G-Movies.htm" ;
proc print data=Classic_Movies(DROP=urllink1) NOOBS N ;
   title "G-rated Movies" ;
   where upcase(rating) = "G" ;
run ;
ods html close ;
ods html body="e:\SGPanel-2-Drill-down-PG-Movies.htm" ;
proc print data=Classic_Movies(DROP=urllink1) NOOBS N ;
   title "PG-rated Movies" ;
   where upcase(rating) = "PG" ;
run ;
ods html close ;
ods html body="e:\SGPanel-2-Drill-down-PG13-Movies.htm" ;
proc print data=Classic_Movies(DROP=urllink1) NOOBS N ;
   title "PG-13-rated Movies" ;
   where upcase(rating) = "PG-13" ;
run ;
ods html close ;
ods html body="e:\SGPanel-2-Drill-down-PG17-Movies.htm" ;
proc print data=Classic_Movies(DROP=urllink1) NOOBS N ;
   title "PG-17-rated Movies" ;
   where upcase(rating) = "PG-17" ;
run ;
ods html close ;
ods html body="e:\SGPanel-2-Drill-down-R-Movies.htm" ;
proc print data=Classic_Movies(DROP=urllink1) NOOBS N ;
   title "R-rated Movies" ;
   where upcase(rating) = "R" ;
run ;
ods html close ;
5. **Drill-down Dashboard – Pie / Bar Chart Dashboard**

Key Points about Code

1. The first step specifies a series of IF-THEN/ELSE statements to create a URL variable to be associated with the Movie Rating variable and another URL variable to be associated with the Movie Category variable.

2. The second step uses a PROC TEMPLATE to create a template overlay to be used in step 3, below.

3. The third step assigns the HTML links, for drill-down purposes, to the URLLINK variables created with the URL variables created in step 1, above, and the user-defined template in step 2, above. It also specifies the ODS HTML and GRAPHICS statements, the dimensions of the dashboard, title and footnote statements, PROC SGPANEL to produce the graphical pie chart and vertical bar chart from an input SAS data set, and the specification of the URL= parameters to allow drill-down capabilities.

4. The fourth step specifies the type of output to produce when the user clicks on one the pieces of the pie chart displayed on the dashboard.

5. The fifth step specifies the type of output to produce when the user clicks on the bars of the vertical bar chart displayed on the dashboard.
**Base-SAS Code:**

```sas
libname mydata 'e:\workshops\workshop data';
/* Step 1 - Assign PieChart URL= to URLRATING Variable */
data Classic_Movies;
  length URLRATING URLCATEGORY $60;
  max=200;
  set mydata.movies;
  if upcase(rating) = 'G' then URLRATING='e:\PieChart-G-Movies.htm';
  else if upcase(rating) = 'PG' then URLRATING='e:\PieChart-PG-Movies.htm';
  else if upcase(rating) = 'PG-13' then URLRATING='e:\PieChart-PG13-Movies.htm';
  else if upcase(rating) = 'R' then URLRATING='e:\PieChart-R-Movies.htm';
run;
/* Step 1b - Assign BarChart URL= to URLCATEGORY Variable */
if upcase(category) = 'ACTION' then URLCATEGORY='e:\BarChart-Action-Movies.htm';
else...
else if upcase(category) = 'HORROR' then URLCATEGORY='e:\BarChart-Horror-Movies.htm';
run;
/* Step 2 - Create PieChart / BarChart Template Overlay with PROC TEMPLATE */
proc template;
define statgraph PieChartBarChartDashboard;
begingraph;
  entrytitle "Drill-down PieChart/BarChart Dashboard";
  layout gridded / columns=1;
  layout lattice / columns=2;
  cell;
    layout region;
    piechart category=Rating / url=URLRATING dataskin=crisp datatransparency=0.3;
  endlayout;
  endcell;
  cell;
    layout overlay / width=250px xaxisopts=(display=(label tickvalues line));
    barchart x=Category y=length / url=URLCATEGORY dataskin=crisp
endgraph;
endtemplate;
```
data transparency=0.3
orient=vertical barwidth=0.8;

endlayout;
endcell;
endlayout;
endlayout;
endgraph;
end;
run;
quit;

/*****************************************************/
/* Step 3 - Produce Graphical Output from PieChart BarChart */
/***************************
Template using PROC SGRENDER
***************************
ods html file='SGRender-PieChart-BarChart-Dashboard.htm'
path='e:\'(url=none);
ods graphics / reset imagemap=on width=8in height=5in
imagename='PieChartBarChart-Dashboard';
title1 "PieChart/BarChart of Movies Data";
title2 "Movie Ratings/Categories";
proc sgrender data=Classic_Movies
   template=PieChartBarChartDashboard;
run;
quit;
title;
ods html close;

/*******************************/
/* Step 4 - Create Pie Chart Drill-down Output for Each Rating Group with */
/* PROC PRINT */
/*******************************/
ods html body="PieChart-G-Movies.htm" path='e:\';
proc print data=classic_movies(drop=urlrating urlcateg max) noobs n;
   title "G-rated Movies";
   where upcase(rating) = "G";
run;
ods html close;

ods html body="PieChart-R-Movies.htm" path='e:\';
proc print data=classic_movies(drop=urlrating urlcateg max) noobs n;
   title "R-rated Movies";
   where upcase(rating) = "R";
run;
ods html close;

*******************************/
/* Step 5 - Create Bar Chart Drill-down Output for Each Category Group with */
/* PROC PRINT */
*******************************/
ods html body="BarChart-Action-Movies.htm" path='e:\';
proc print data=classic_movies(drop=urlrating urlcateg max) noobs n;
   title "Action Movies";
   where upcase(category) = "ACTION";
run;
ods html close;

ods html body="BarChart-Horror-Movies.htm" path='e:\';
proc print data=classic_movies(drop=urlrating urlcateg max) noobs n;
   title "Horror Movies";
   where upcase(category) = "HORROR";
run;
ods html close;
References

Acknowledgments
The authors thank Rachel Straney and Jeffrey Kromrey, Building Blocks Section Chairs, for accepting our abstract and paper; Brian Varney, SESUG 2016 Academic Chair; Kristen Harrington, SESUG 2016 Operations Chair; the SouthEast SAS Users Group (SESUG) Executive Board; and SAS Institute for organizing and supporting a great conference!

Trademarks Citations
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.
Author Information
Kirk Paul Lafler is an entrepreneur, consultant and founder of Software Intelligence Corporation, and has been using SAS since 1979. Kirk is a SAS Certified Professional, provider of IT consulting services, professor at UC San Diego Extension and educator to SAS users around the world, mentor, and emeritus sasCommunity.org Advisory Board member. As the author of six books including Google® Search Complete (Odyssey Press. 2014) and PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press. 2013); Kirk has written more than five hundred papers and articles; been an Invited speaker and trainer at five hundred-plus SAS International, regional, special-interest, local, and in-house user group conferences and meetings; and is the recipient of 23 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Joshua M. Horstman is an independent statistical programmer with 18 years experience using SAS in the pharmaceutical industry. He has experience working for GlaxoSmithKline, Eli Lilly and Company, inVentiv Health, and others. Josh has presented numerous papers at PharmaSUG, SAS Global Forum, and other SAS User Group meetings. He holds a bachelor’s degree in mathematics and computer science, and a master’s degree in statistics from Colorado State University.

Roger D. Muller, Ph.D. has been as SAS user for over 40 years, primarily in the life sciences, business marketing, and educational areas. His current interests are in creative display of textual and graphical information using a wide variety of SAS products.

Comments and suggestions can be sent to:

Kirk Paul Lafler  
Senior SAS® Consultant, Application Developer, Data Analyst, Educator and Author  
E-mail: KirkLafler@cs.com  
LinkedIn: http://www.linkedin.com/in/KirkPaulLafler  
Twitter: @sasNerd

Joshua M. Horstman  
Senior SAS® Consultant, Application Developer and Author  
E-mail: jmhorstman@gmail.com  
LinkedIn: https://www.linkedin.com/in/joshuahorstman

Roger D. Muller  
Senior SAS® Consultant, Application Developer and Author  
E-mail: rdmuller@hotmail.com  
LinkedIn: https://www.linkedin.com/in/roger-muller-b42a9929