Paper 260-2012

A Different Point of View with ODS PDF in SAS® 9.3

Scott Huntley and Woody Middleton, SAS, Cary, NC, USA

ABSTRACT

The ODS PDF statement in SAS 9.3 is giving you new ways to change how you view and display your output. Several enhancements in ODS PDF 9.3 are sure to be crowd pleasers. Topics in this paper will include: how to change orientation mid-file, how to drill down from your PDF file, how a stronger use of vector-based graphics will save memory and time, and much more. Be the first to find out how to change your output and get the different point of view you've been wanting.

INTRODUCTION

A different point of view often provides people a new perspective on a topic. A different point of view generated by the ODS PDF statement can make people see your output in a new light. In this paper we will discuss how to format your PDF output and give it some new looks. Responding to customer requests and Technical Support input, the development team has added some new functionality to the ODS PDF statement that will provide you with better looking output, more options, and flexibility. All the syntax in this paper is available using SAS 9.3. Feel free to cut and paste the code examples so you can quickly see some new points of views in your output.

ORIENTATION SWAP

A common request is the ability to change orientation within a PDF file. In the past, when using SAS you would have to create two separate PDF files with ODS and then concatenate them together using a third-party product. Now with SAS 9.3 you can swap orientation every page if desired. To swap all you have to do is use the global OPTIONS statement and set the desired value for the option ORIENTATION.

```
options orientation=portrait;
```

Here is official documentation on the ORIENTATION option:

ORIENTATION=PORTRAIT | LANDSCAPE | REVERSEPORTRAIT | REVERSELANDSCAPE

Syntax Description

PORTRAIT specifies the paper orientation as portrait. This is the default.

LANDSCAPE specifies the paper orientation as landscape.

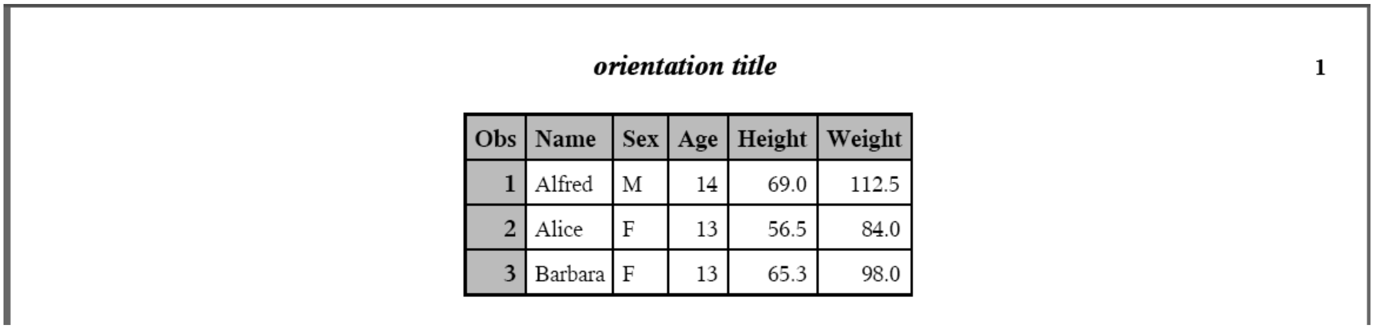
REVERSEPORTRAIT specifies the paper orientation as reverse portrait. Use this value to control the top of the page relative to how the paper is inserted into the input paper tray.

REVERSELANDSCAPE specifies the paper orientation as reverse landscape. Use this value to control the top of the page relative to how the paper is inserted into the input paper tray.

Now that we know the option let's see how it looks inside some code. In the following example I am going to create four PROC PRINT tables with varying observations. The varying observations will help us see how each unique PROC PRINT table will look in a different orientation view. You will see that I am resetting the options statement before each PROC PRINT call to set the new orientation. Also note that once you reset a new orientation a new page is automatically given. Each page can handle only one orientation per page. In the code, once we see an orientation swap, we perform an explicit page for the new orientation. Here is a quick snippet of code to show how it works:

```
options nodate orientation=portrait;
title 'orientation title';
footnote 'orientation footnote';
ods pdf file="test.pdf";

proc print data=sashelp.class(obs=3);run;

options orientation=landscape;
proc print data=sashelp.class(obs=5);run;

options orientation=reverseportrait;
proc print data=sashelp.class(obs=7);run;

options orientation=reverselandscape;
proc print data=sashelp.class(obs=9);run;
ods pdf close;
```

First we get page one of the output:

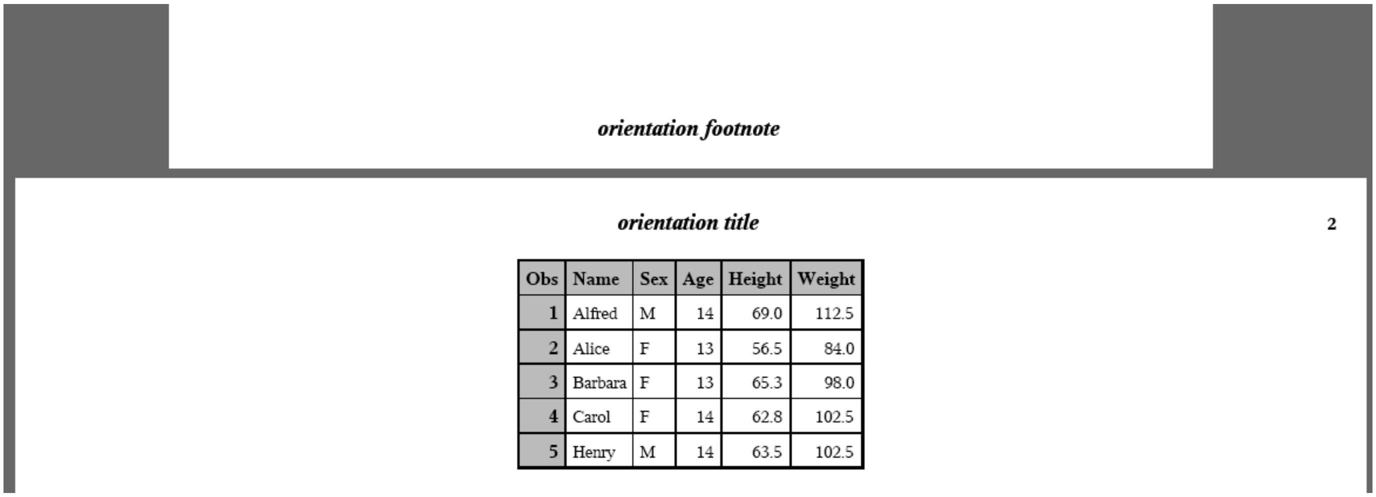


orientation title 1

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0

Display 1. Page 1 Showing Portrait Orientation

This is just normal portrait output showing the first three observations of sashelp.class. Next we reset the orientation to landscape.



orientation footnote

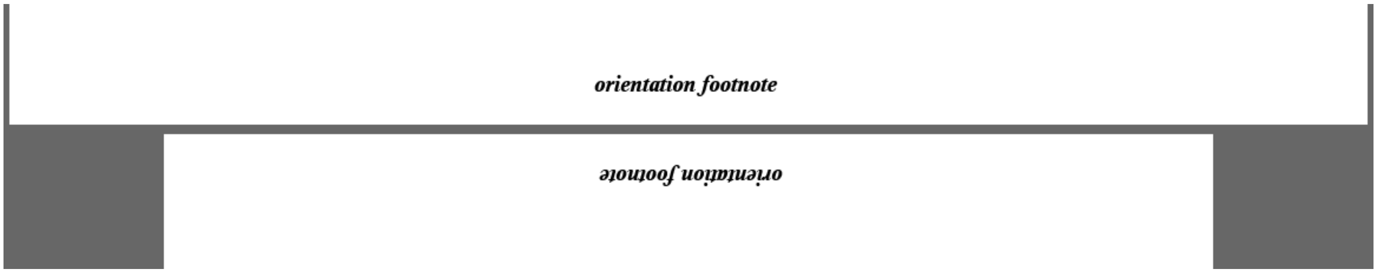
orientation title

2

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5

Display 2. Page 2 of Output Showing Landscape Orientation

You can see the footnote at the end of page 1 is a portrait orientation. At the start of page 2 you see we have now swapped to a landscape orientation and now display the first five observations of sashelp.class. Once again, I want to reiterate that once you change orientations a new page will immediately occur. With one orientation I could easily put both tables on one page using the “startpage=no” option. Changing orientation overrides the STARTPAGE option and forces a new page even if you asked not to start one. Before we move onto our third page I want to show the bottom of page 2 first. Since I’m changing orientation now to reverse portrait the page will look unusual at first. Let me show you the bottom of page 2 and the start of page 3.

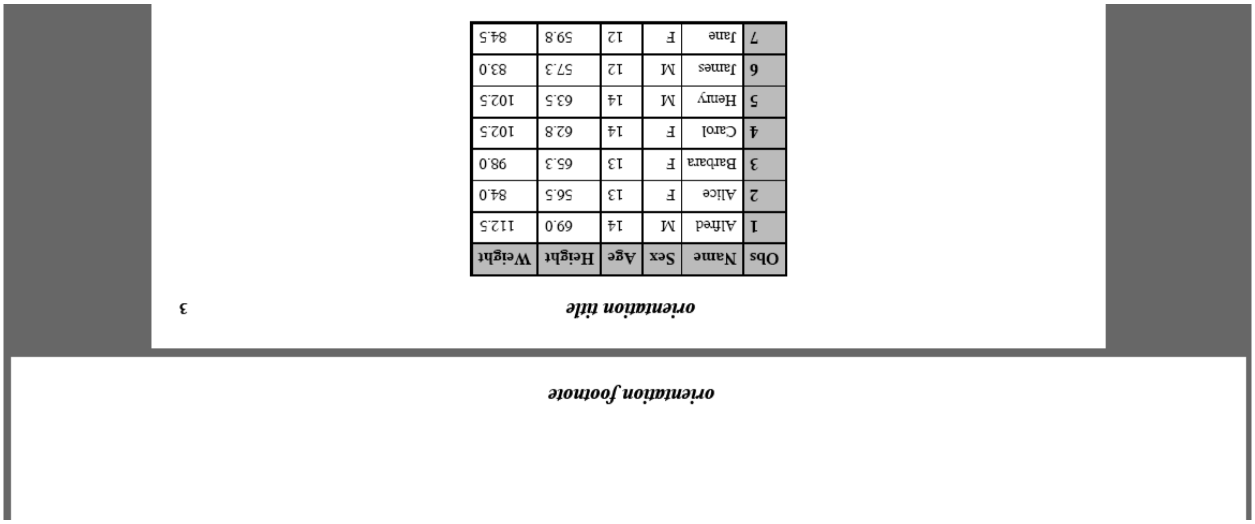


orientation footnote

orientation footnote

Display 3. Page 3 of Output Showing Orientation Reverse Portrait

You can see the footnote of page 2 which has the orientation of landscape. Now we switch to reverse portrait. It has the same dimensions of regular portrait but the layout of the page is reversed 180 degrees. Here is the bottom of the page. Now we are showing the first seven observations of sashelp.class in the new orientation.



orientation title

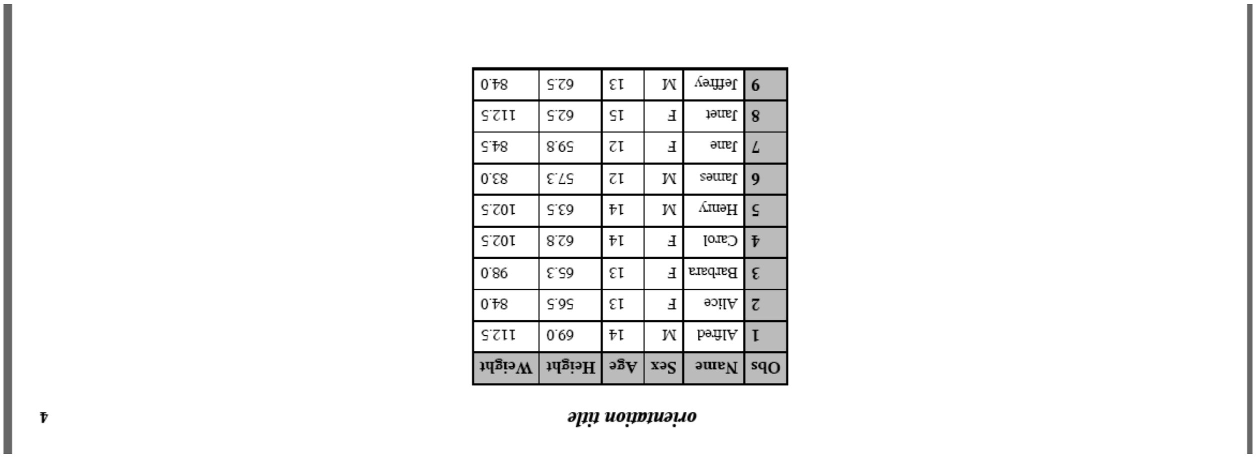
3

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5

orientation footnote

Display 4. Bottom of Page 3 of Output Showing Orientation Reverse Portrait

Finally, our last page is showing the first nine observations of sashelp.class in reverse landscape. Just like reverse portrait the output is flipped 180 degrees from what is normally presented. Here is the bottom of page 4:



orientation title

4

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0

Display 5. Bottom of Page 4 of Output Showing Reverse Landscape Orientation

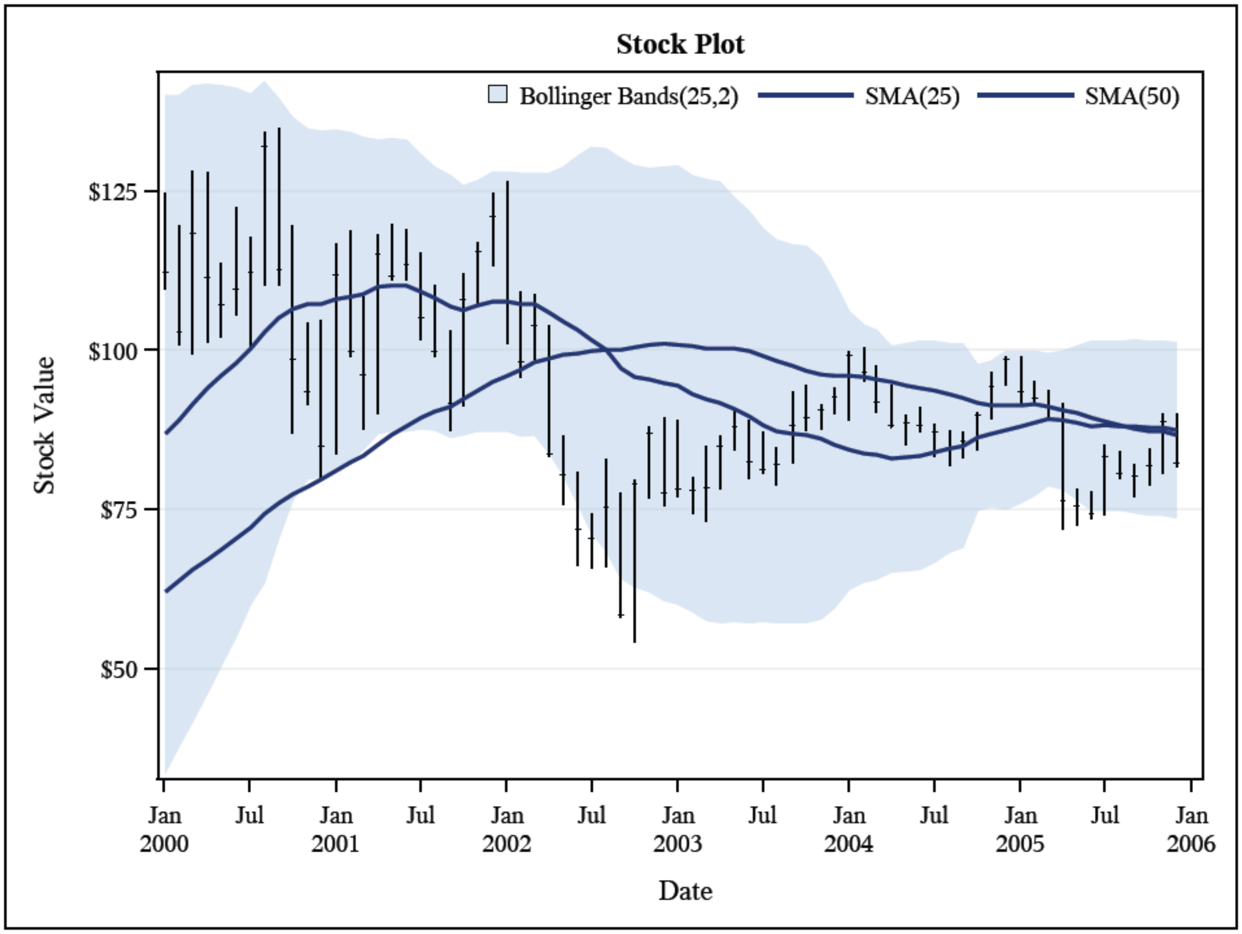
The option is simple to use and shows your output in a new way. Actually, you now have four new orientations to show you output. This isn't just restricted to tables. Any output objects like titles, footnotes, text, tables, or graphs can be presented in these new orientations. The first new point of view discussed is using a new orientation to view your output. Now let's move to a new way to show off statistical graphics output.

VECTOR-BASED GRAPHICS

Graphs are a common output type directed to PDF files from SAS. In previous releases, ODS PDF would interface with STATGRAPH to incorporate its wonderful output into PDF files. Basically ODS PDF would request output from STATGRAPH and then ODS PDF would create an image of the graph and place it on a page of output. While the approach at the time it was developed was satisfactory, current technology allows us better output today. When you convert a graph to an image you lose resolution and crispness of the original graph. Also putting images into PDF files can become costly memory wise while processing, leading to longer processing time to create PDF files. The file size is greater with images that are stored inside the PDF as well. To the rescue comes vector-based graphics. It allows us to tackle all those problems and give us sharper output. Let's start with some code creating an SGPLOT with and without vector-based graphics. Below is a code snippet. I will put the complete code at the end of this paper.

Vector graphics is the default in SAS 9.3. Here is sample code:

```
ods graphics / reset imagename="HighLow";
ods pdf file='nv_highlow.pdf' notoc dpi=300;

Title1 "Stock Plot";
proc sgplot data=moveavg (where=(date>='01jan2000'd));
   yaxis grid label="Stock Value";
   band x=date upper=bolupper lower=bollower /
            transparency=0.5 legendlabel="Bollinger Bands(25,2)" name="boll";
   vector x=date y=high / yorigin=low xorigin=date noarrowheads;
   scatter x=date y=close / markerattrs=(symbol=plus size=3);
   series x=date y=avg25 / lineattrs=GraphFit legendlabel="SMA(25)" name="d25";
   series x=date y=avg50 / lineattrs=GraphFit2(pattern=solid) legendlabel="SMA(50)"
name="d50";
   keylegend "boll" "d25" "d50" / across=4 noborder position=TopRight location=inside;
run;

ods pdf close;
```

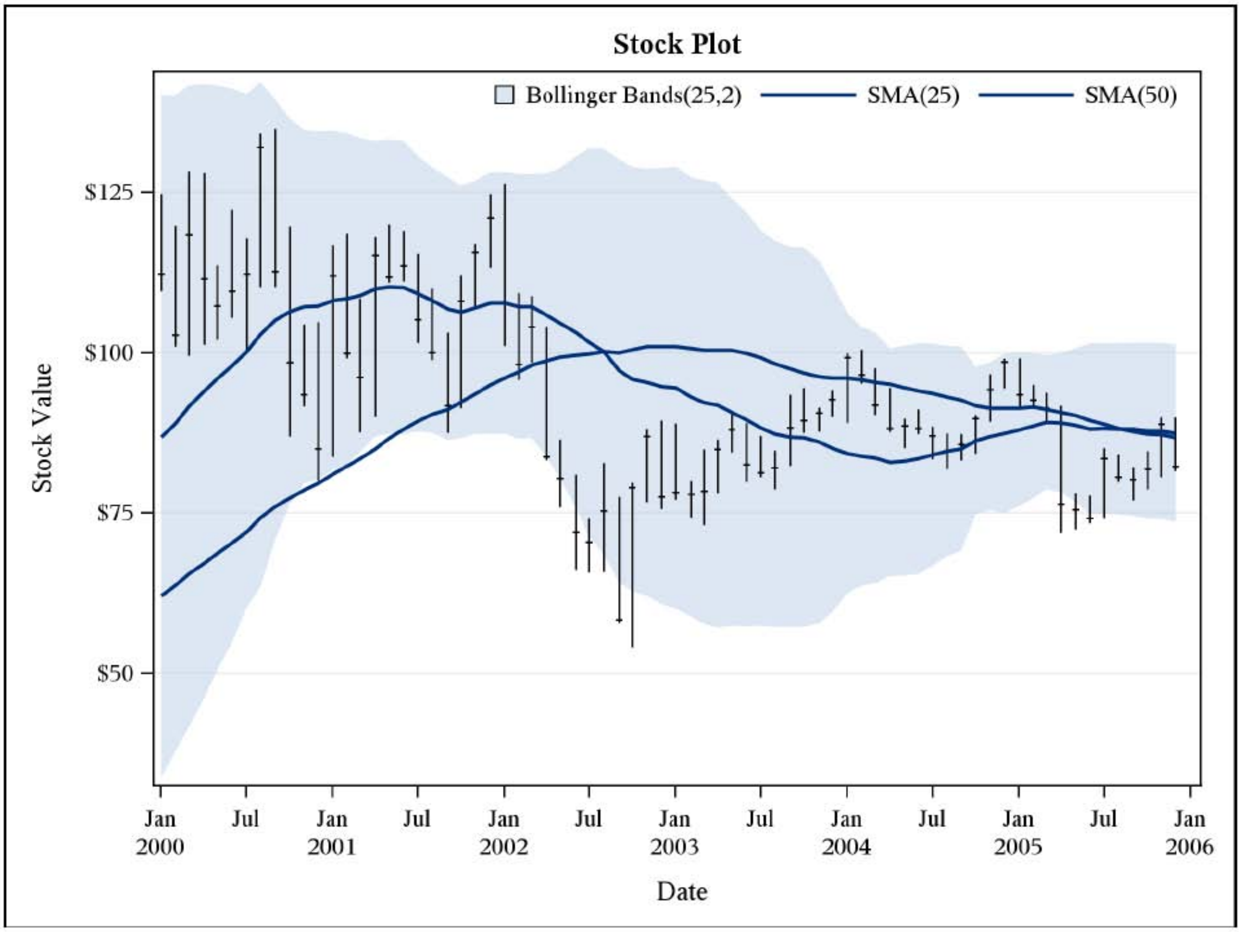


Display 6. SGPLOT Using Vector-Based Graphics

Using the IMAGEFMT option in the ODS GRAPHICS statement means the output requested or generated is not using vector graphics:

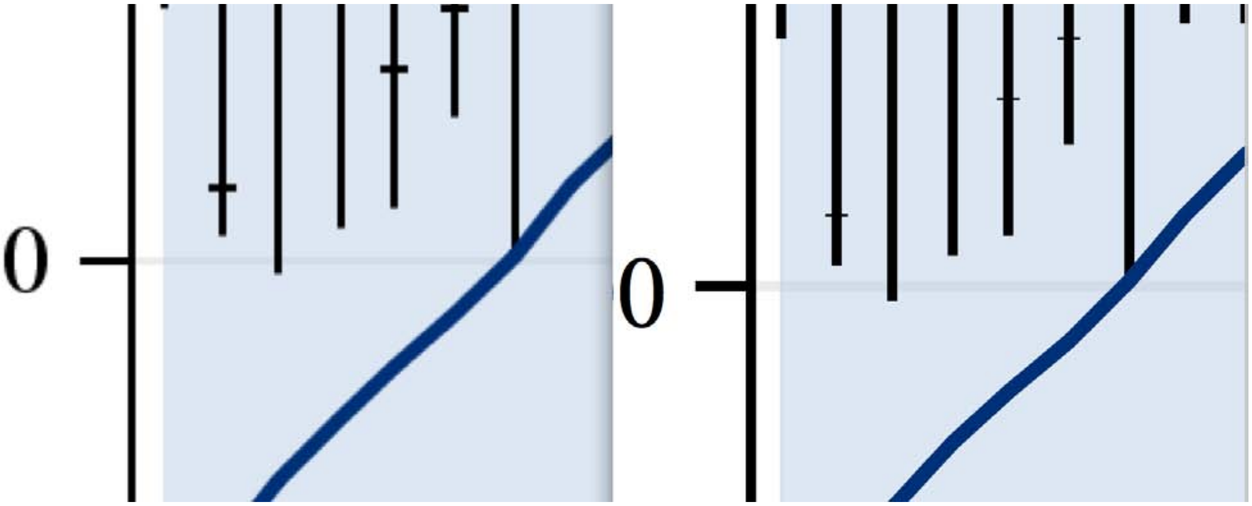
```
ods graphics / reset imagename="HighLow" imagefmt=png;
ods pdf file='nv_highlow.pdf' notoc dpi=300;

Title1 "Stock Plot";
proc sgplot data=moveavg (where=(date>='01jan2000'd));
   yaxis grid label="Stock Value";
   band x=date upper=bolupper lower=bollower /
            transparency=0.5 legendlabel="Bollinger Bands(25,2)" name="boll";
   vector x=date y=high / yorigin=low xorigin=date noarrowheads;
   scatter x=date y=close / markerattrs=(symbol=plus size=3);
   series x=date y=avg25 / lineattrs=GraphFit legendlabel="SMA(25)" name="d25";
   series x=date y=avg50 / lineattrs=GraphFit2(pattern=solid) legendlabel="SMA(50)"
name="d50";
```

```
   keylegend "boll" "d25" "d50" / across=4 noborder position=TopRight location=inside;
run;

ods pdf close;
```



Display 7. SGPLOT Not Using Vector-Based Graphics (IMAGEFMT=PNG)

At first glance they look very similar. Look closely at the displays again and you will see the crisp clean lines in Display 6. The words and numbers in the graph are sharper. Instead of an image embedded inside a PDF file you are now seeing the PDF file itself rendering the graph. With vector-based graphics you get more than a visual improvement. Here is a quick look at our output when we zoom in. The output on the left is the non-vector SGPLOT. The right is vector-based graphics. Zooming in like this shows how crisp the lines are. The details are clearer with the vector-based graphics. The lines in the graph look bigger, wider, and bolder.



Display 8. Zoomed in Look at SGPLOTs

As you can see in the code snippet, I used the option `dpi=300` in the ODS PDF statement to get a more dots per inch in the image. Remember the default DPI for the ODS PDF statement is 150. Using a higher DPI will always make your graphics look better. However, if I had tried `dpi=600`, the non-vector test case would have run out of memory trying to convert the image to a larger resolution. Vector-based graphics use much less memory so the job will run to completion without error. Less memory usage while running allows for more graphs to be embedded inside a PDF file. In addition, the run time to create these graphs will be less. The biggest win, in my opinion, other than appearance, is the decrease in file size. To prove that point I ran the above SGPLOT snippet 20 times in one job to create a PDF file. The resulting size was 2,325 KB. Then I used vector-based graphics and re-ran with those same 20 SGPLOTs. The second run had a file size of 203 KB. That is a huge difference which surely makes everyone happy. Vector-based graphics are a great way to show off your STATGRAPH output with a cleaner appearance, while saving on memory and file space at the same time. Next let’s discuss the GTITLE and GFOOTNOTE options that are new options in the ODS PDF statement in SAS 9.3.

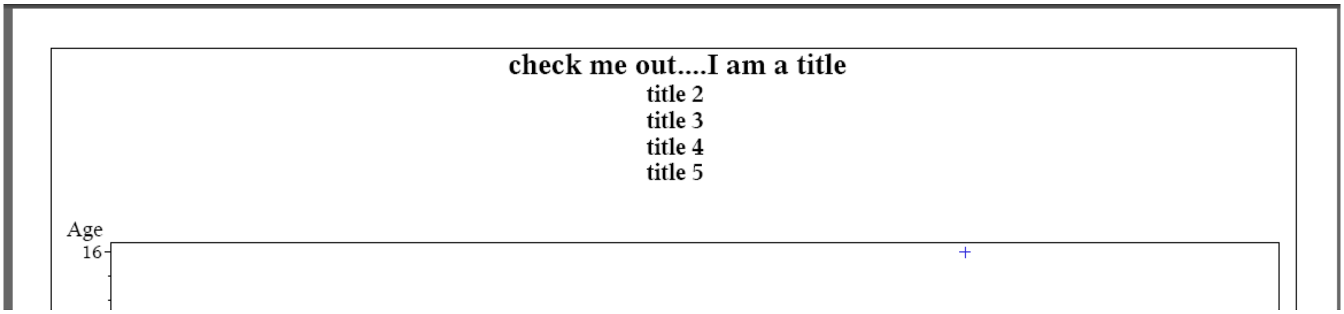
GRAPH TITLES AND FOOTNOTES

By default in all destinations (LISTING, HTML, RTF, and PDF), graph titles and footnotes appear within the graph output/image. RTF and HTML have the ability to use the options GTITLE/NOGTITLE and GFOOTNOTE/NOGFOOTNOTE to toggle this behavior. These options enablesyou to put the titles and footnotes inside or outside the graph output. The ODS PDF statement can now use these same options starting in SAS 9.3. Let’s start by showing the SAS documentation for the GTITLE option:

> GTITLE | NOGTITLE
>
> controls the location where titles are printed in the graphics output.
>
> GTITLE
>
> prints the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.
>
> NOGTITLE
>
> prints the title that is created by ODS, which appears outside of the graph borders.
>
> Default: GTITLE

It's a toggle switch. GTITLE puts the titles in the graphs, while NOGTITLE puts the titles outside the graph. The same is true for GFOOTNOTE and NOGFOOTNOTE. Here is some sample SAS code to show off this option:

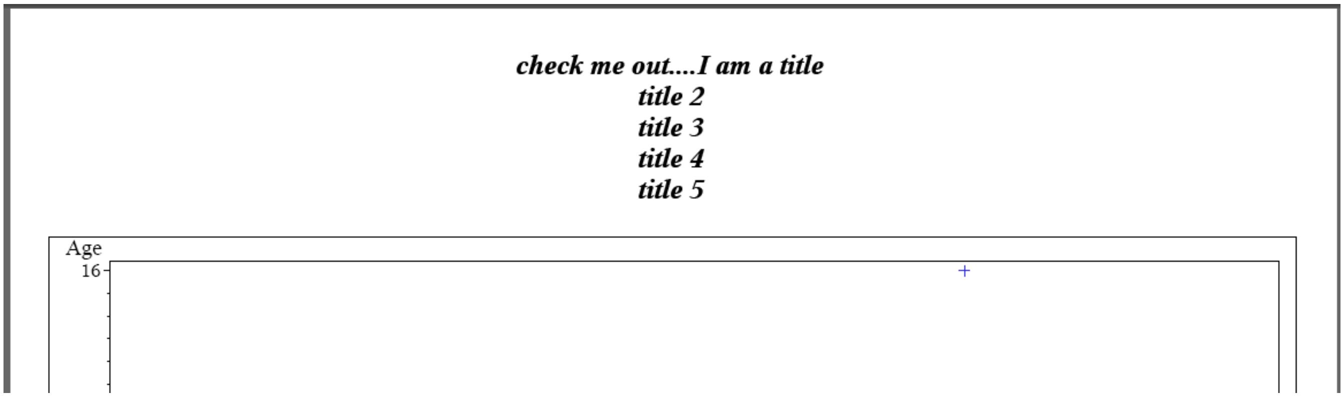
```
options nodate nonumber;
goptions reset=all border;
title 'check me out....I am a title';
title2 'title 2';
title3 'title 3';
title4 'title 4';
title5 'title 5';
footnote 'check me out...I am a footnote';
footnote2 'footnote 2';
footnote3 'footnote 3';
footnote4 'footnote 4';
footnote5 'footnote 5';
ods pdf file="test.pdf" notoc;
proc gplot data=sashelp.class; plot age*height; run; quit;
ods pdf close;
```



Display 9. GPLOT Output Using the Default Option GTITLE

Since GTITLE is the default, I don't need to set anything in the ODS PDF statement. The titles will automatically be placed inside the graph. Notice how I set the border option using GOPTIONS. This puts a border around the graph output. You can clearly see the titles are inside the graph output. Now making one small change on your PDF statement you can get your graph titles outside the graph image.

```
ods pdf file="test.pdf" nogtitle notoc;
```



check me out....I am a title
title 2
title 3
title 4
title 5

Display 10. GPLOT Output Using the Option NOGTITLE

Now since the titles are on the outside of the graph you can now take advantage of ODS styles. Notice how they are bold and italicized, which is default for all titles not managed by the LISTING ODS statement. You could perform some simple inline functions or styles to add color or other font attributes to the titles. The use of inline styles has been documented and discussed in a past paper (Huntley 2008). This new option works with both SAS/GRAPH and ODS output. I’ve briefly mentioned styles and how they can change the appearance of titles. Now let’s discuss another new visual change that is available when using the ODS PDF statement in SAS 9.3

TRANSPARENCY USING RGBA

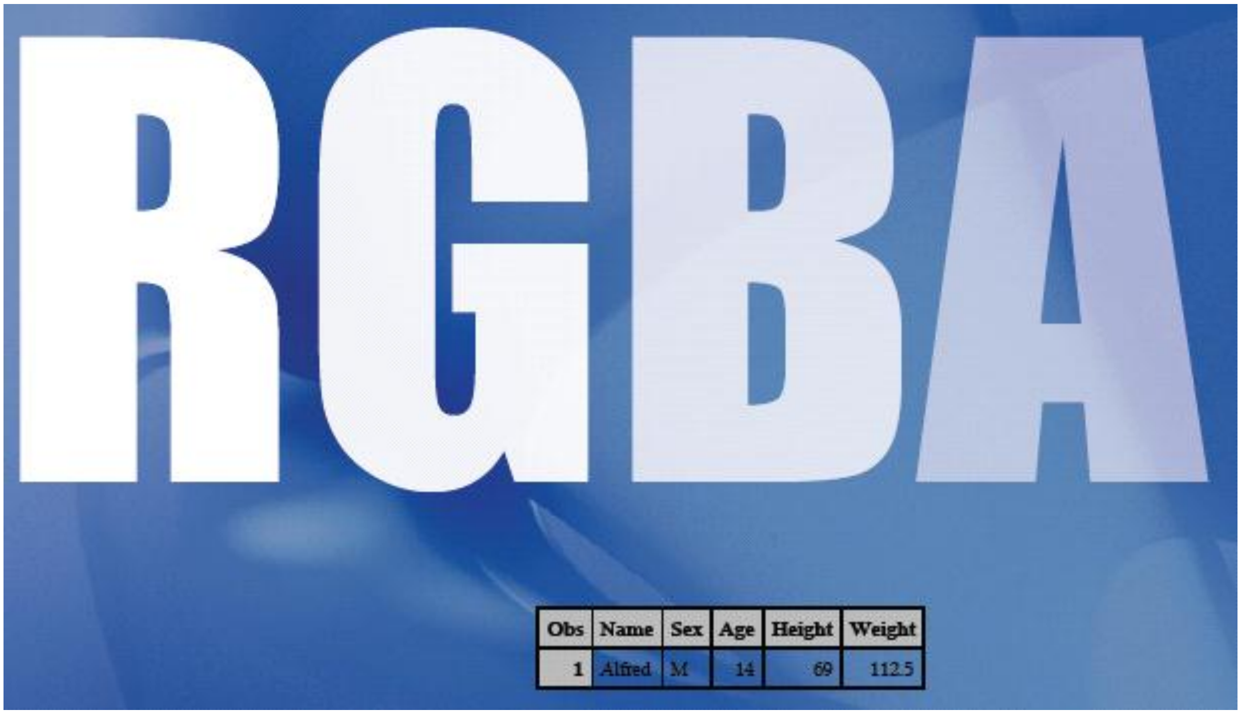
Styles in ODS provide you a way to add a personalized touch to your output. Styles touch all aspects of your output such at text strings, cells in tables, and page backgrounds. We are going to focus on color and how you can take advantage of a new transparency feature of the ODS PDF statement in SAS 9.3. RGB Color models combine red, green, and blue lights together to produce a huge array of colors. To quickly recap RGB, you specify a hexadecimal number between 0 and 255. Each value is then used to determine how much of that one color you want in your final color. By combining all three colors you will create a new color. The Web is full of information about hexadecimal numbers and colors. There are color maps that can give you the exact hexadecimal numbers you need to display the perfect color in your output, so I will not offer one here.

To achieve transparency with the color model we need to add some more information to the existing RGB model. We now can use the RGBA color model. **RGBA** stands for **R**ed **G**reen **B**lue **A**lpha …an RGB model with extra information. The Alpha channel determines opacity. Complete transparency is 0% (or no color at all) while 100% is fully opaque (like current colors without alpha information). Let’s start with a simple example using RGBA to see how it works with the ODS PDF statement.

```
proc template;
define style rgba;
parent = styles.printer;
class body / backgroundimage = 'c:\temp\background.jpg’;
style big /
fontsize = 4in
fontfamily = impact
fontweight = bold ;
end;
run;
ods listing close;
```

```
title;
options orientation=landscape nodate;
ods escapechar '^';

ods pdf file='rgba.pdf' style=rgba;
ods pdf text = "^{style big[color=RGBAffffffff]R}^{style
big[color=RGBAffffffcc]G}^{style big[color=RGBAffffff99]B}^{style
big[color=RGBAffffff66]A}";
proc print data=sashelp.class(obs=1); run;
run;
ods _all_ close;
```

There is lot to discuss in this code. Let's view the output first and then go through the code in smaller segments. I will explain what is happening in each segment.



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5

Display 11. Text Output Using Transparency

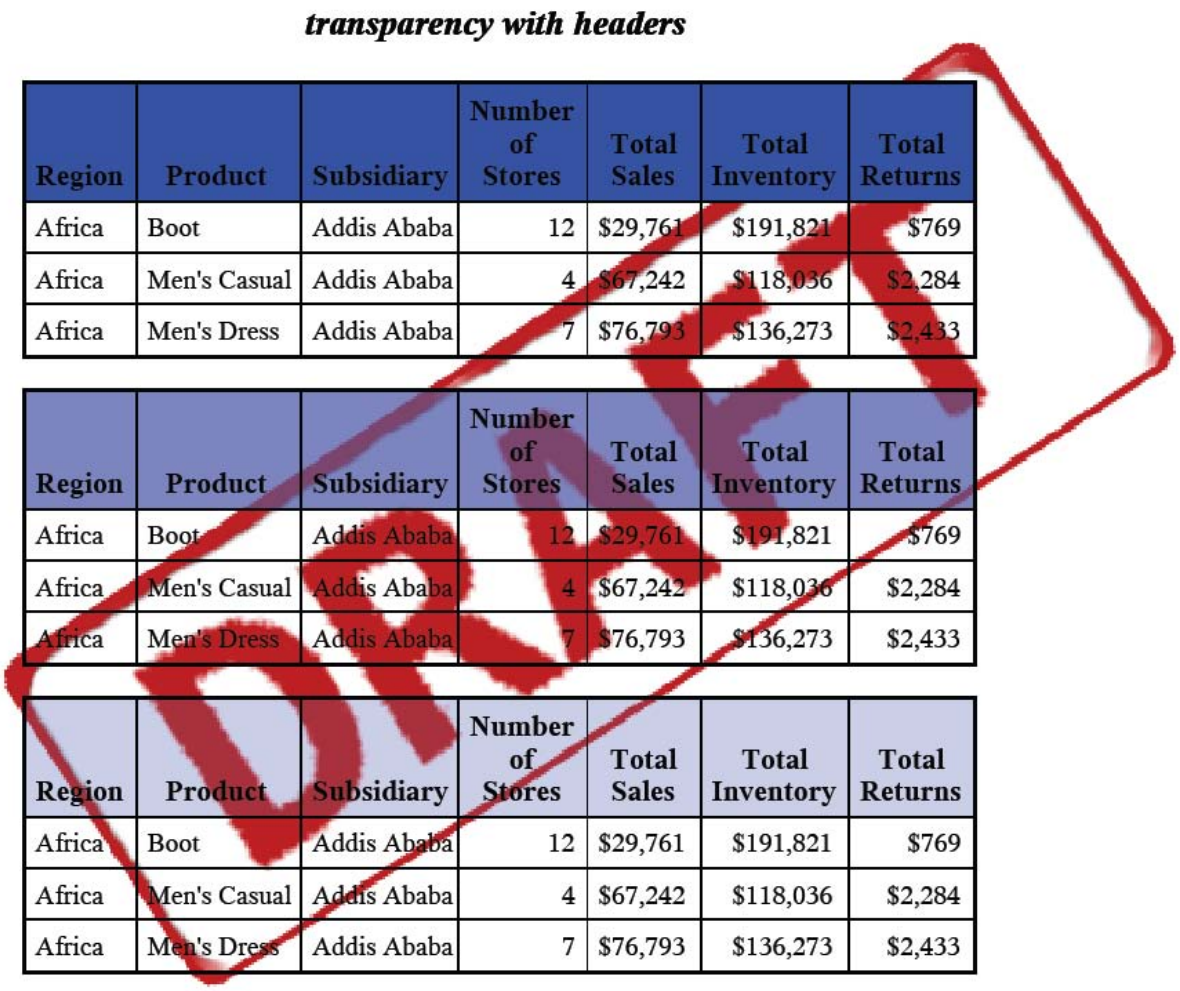
At first glance you see the background image on the page. We have a text statement and a small table. The table is showing the background through data cells. The cells in the table are in a black font so the output is not hidden by the background. Now we will focus on the text string 'RGBA'. Each letter looks lighter going from left to right. This is how transparency works. You will start to see the background image more and more as your font is less opaque. Okay let's walk through the code now. First we see a PROC TEMPLATE step, which is creating a personal style, RGBA, to be used later on. In this style definition we are defining a background image for the body of the output. We also create a small style element called "big", which is setting up some font specifications. Moving down we use the style on our PDF statement so that sets our background image. Then we hit the ODS PDF TEXT statement. This is where the transparency begins. Here is the text statement again:

```
ods pdf text = "^{style big[color=RGBAffffffff]R}
                ^{style big[color=RGBAffffffcc]G}
                ^{style big[color=RGBAffffff99]B}
                ^{style big[color=RGBAffffff66]A}";
```

I'm not going to explain the style function, which has been explained in past papers (Huntley 2006, Huntley 2008). I will focus on the invocation of the "big" style function call. Remember from the PROC TEMPLATE code it was setting up font specifications. It set the font size, family, and weight. Now we are adding color to the font specification. You will notice the first color specifies 'RGBAffffffff'. The first four letters, RGBA, are needed to tell ODS that you are going to set an RGBA color. The next eight letters are hexadecimal numbers specifying the color you want to use. Now we are going to focus on the last two hexadecimal numbers. These two numbers are the extra information used in the RGBA model to indicate transparency. Remember the closer the number is to zero the more transparent the color. When I convert the hexadecimal numbers to decimal numbers you can see how we approach zero:

Hex Number	Decimal Number	Opacity
FF	255	100%
CC	204	80%
99	153	60%
66	102	40%

If I divide the resolved decimal number by 255 (no opacity) we can get what percentage of opacity the color will have. You will notice that it is dropping by 20% on each letter. The last letter has the most transparency and it is allowing the background to show through the most. So we have this great functionality, now how do we use it? We know we can use it on text statements. Let's see an example with a table and its header:



transparency with headers

Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
Africa	Men's Casual	Addis Ababa	4	$67,242	$118,036	$2,284
Africa	Men's Dress	Addis Ababa	7	$76,793	$136,273	$2,433

Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
Africa	Men's Casual	Addis Ababa	4	$67,242	$118,036	$2,284
Africa	Men's Dress	Addis Ababa	7	$76,793	$136,273	$2,433

Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
Africa	Men's Casual	Addis Ababa	4	$67,242	$118,036	$2,284
Africa	Men's Dress	Addis Ababa	7	$76,793	$136,273	$2,433

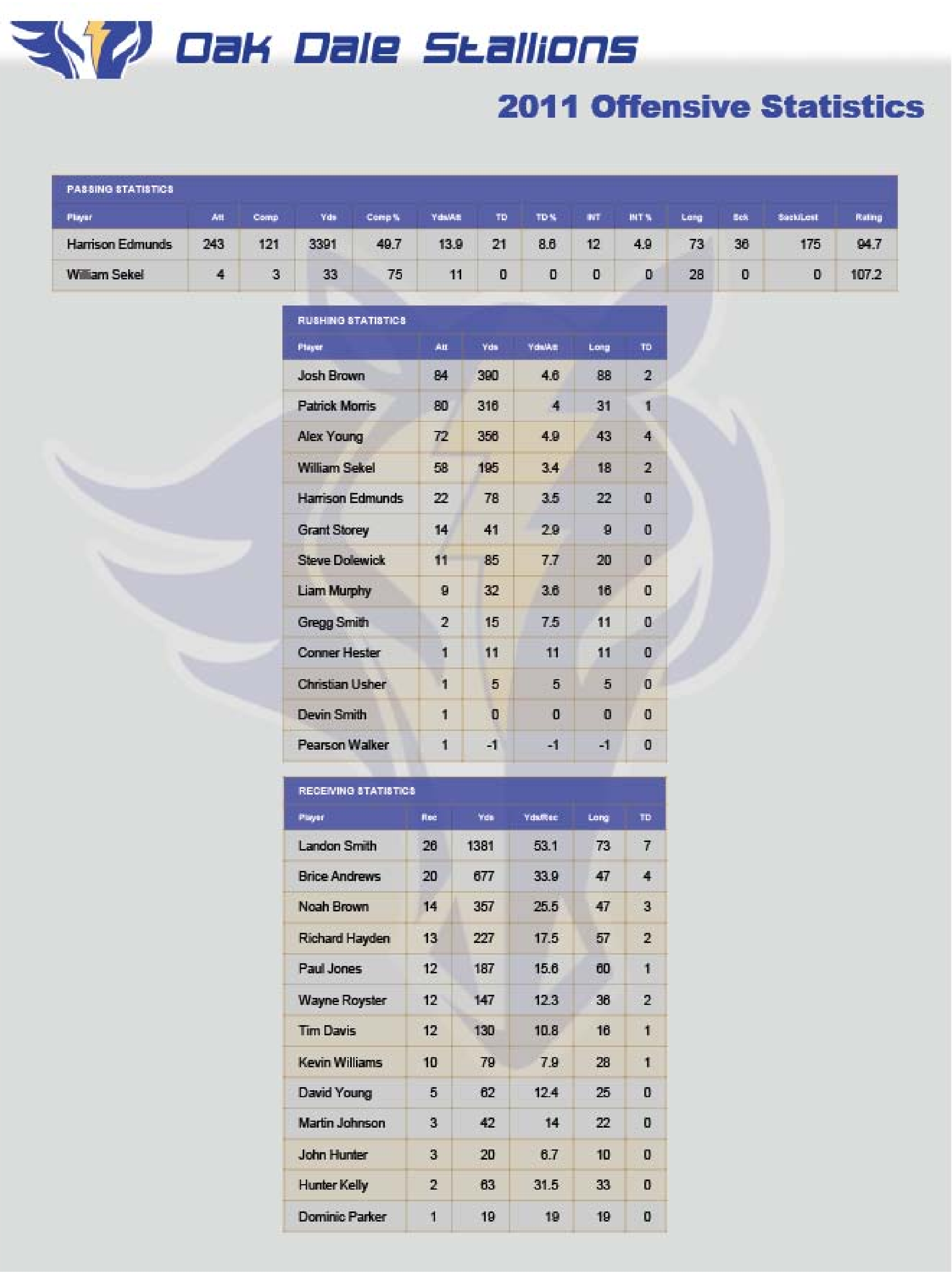
Display 12. Table Headers Using Transparency

```
proc template;
   define style bkgdimage;
   parent=Styles.Printer;
      style Body from Document
         "Controls the Body file." /
              backgroundimage='c:\temp\draft.png';
         end;
              run;

title 'transparency with headers';
ods pdf file='trans1.pdf' startpage=no style=bkgdimage;

/* 100% */
proc report data=sashelp.shoes(obs=3) nowd
```

```
style(header)={background=RGBA0000FFFF};
run;

/* 60% */
proc report data=sashelp.shoes(obs=3) nowd
style(header)={background=RGBA0000FF99};
run;

/* 20% */
proc report data=sashelp.shoes(obs=3) nowd
style(header)={background=RGBA0000FF33};
run;

ods pdf close;
```

We have three reports from the REPORT procedure with their respective headers using RGBA to set the color. The first report will have complete opacity since it is set to 100%, then set we set the opacity to 60% and finally we set the third header to 20%. Notice the first output completely blocks out the red word in the background image. The next two report headers allow more and more of the DRAFT image to show. In the third header you can almost completely see the image without hindrance.

So now let’s see how to work with the data cells inside some tabular output. We need to add them to the transparency mix. Here is another example that fellow ODS developer Kevin Smith helped me put together:



Oak Dale Stallions

2011 Offensive Statistics

PASSING STATISTICS

Player	Att	Comp	Yds	Comp %	Yds/Att	TD	TD %	INT	INT %	Long	Sck	Sack/Lost	Rating
Harrison Edmunds	243	121	3391	49.7	13.9	21	8.6	12	4.9	73	36	175	94.7
William Sekel	4	3	33	75	11	0	0	0	0	28	0	0	107.2

RUSHING STATISTICS

Player	Att	Yds	Yds/Att	Long	TD
Josh Brown	84	390	4.6	88	2
Patrick Morris	80	316	4	31	1
Alex Young	72	356	4.9	43	4
William Sekel	58	195	3.4	18	2
Harrison Edmunds	22	78	3.5	22	0
Grant Storey	14	41	2.9	9	0
Steve Dolewick	11	85	7.7	20	0
Liam Murphy	9	32	3.6	16	0
Gregg Smith	2	15	7.5	11	0
Conner Hester	1	11	11	11	0
Christian Usher	1	5	5	5	0
Devin Smith	1	0	0	0	0
Pearson Walker	1	-1	-1	-1	0

RECEIVING STATISTICS

Player	Rec	Yds	Yds/Rec	Long	TD
Landon Smith	26	1381	53.1	73	7
Brice Andrews	20	677	33.9	47	4
Noah Brown	14	357	25.5	47	3
Richard Hayden	13	227	17.5	57	2
Paul Jones	12	187	15.6	60	1
Wayne Royster	12	147	12.3	36	2
Tim Davis	12	130	10.8	16	1
Kevin Williams	10	79	7.9	28	1
David Young	5	62	12.4	25	0
Martin Johnson	3	42	14	22	0
John Hunter	3	20	6.7	10	0
Hunter Kelly	2	63	31.5	33	0
Dominic Parker	1	19	19	19	0

Display 13. Full Example Using Transparency

Using PROC TEMPLATE and RGBA you can make output for your audience that will really stand out. Notice how the watermark behind the table is shown through the data cells and headers. You will also notice the interlacing colors on the rows of data. Using the right colors you can create a perfect blend with the watermark which makes the whole report look fantastic. There is even transparency on the system titles at the top, so it also blends in well with the overall color scheme.

I've shown in three different examples how to use transparency in the ODS PDF statement. First I used the inline style function for my ODS PDF text statement to show transparency with the text string "RGBA". Then I used REPORT procedure's style function in the second example to show transparency in the headers. Now in this third example we are using PROC TEMPLATE to create this complete report. I will share the entire code needed for this output in the Appendix, but for now let's look at this code snippet to see how I set the transparency in this example:

```
class data, header, rowheader, tablehead, altrow /
    fontfamily = arial
    color = black
    borderwidth = 1px
    borderstyle = solid
    bordercolor = RGBAdcdcdc4c
    backgroundcolor = RGBAb0b0b04c
    paddingtop = 10px
    paddingbottom = 10px
    paddingleft = 18px    paddingright = 18px;
class header, rowheader, tablehead /
    fontsize = 1.1em
    fontweight = bold
    color = white
    backgroundcolor = RGBA3948a0cc
    fontfamily = arial
;
```

In the CLASS statement for the data element you can see we are setting the BORDERCOLOR and BACKGROUNDCOLOR attributes. You will recognize the RGBA color scheme being used. . As we have seen in every example you need the "RGBA" before the color designation to have transparency. . The last two hexadecimal numbers on the BORDERCOLOR and BACKGROUNDCOLOR are (4c), which is 76 in decimal. . The number being so low is why the transparency is so visible. . It takes trial and error with colors and background images to get the right mix but you can use it to produce amazing results. . Let's move to our last topic, which involves digging deeper into your output.

HOTSPOTS AND DRILL-DOWN FUNCTIONALITY WITH PDF OUTPUT

Using links and anchors, the ODS PDF statement has had the ability to link to other PDF files and HTML files for several releases of SAS. The ODS HTML statement has had not only the ability to link to external files from an ODS Table but also the ability to link to external files with traditional SAS/GRAPH procedures. . In SAS 9.3 we have added the capability to link to the SAS/GRAPH output with both the ODS LISTING output and ODS PDF output. . This feature enables you to enhance the graphic output and improve the communication of information in your documents. . We now have the ability to "drill down" into the data behind a graph and to surface more information in our reports

Let’s look at how this feature works with the ODS PDF statement in SAS 9.3 using a simple example.

To reduce the "clutter" on the PDF output, we will disable the page number and date from the output using the OPTIONS statement. . Using the GOPTIONS statement we will disable the border for graphics output.

```
options nodate nonumber;

goptions noborder;
```

Suppose we want to illustrate population differences in some of the southeastern states and link a map as well as the state’s official Web site to our document. . We will start with a subset of population data from the 2010 Census about the states North Carolina, South Carolina, Virginia, Georgia, and Florida (Source: US Census Bureau 2010 Census data).

The data set mydata will be used as the input data for the PROC GCHART and maps for the individual states. . First we will build the state and population elements in a data step.

```
data mydata;

   format population comma12.0;

   input st $ 1-2 population;

   year=2010;

   state=stfips(st);

   name=stnamel(st);

datalines;

VA  8001024

NC  9535483

SC  4625364

FL 18801310

;

run;
```

How do we add a "hotspot" or a link to the PDF output? For SAS/GRAPH output created with ODS LISTING or ODS PDF we use an HTML variable within the data set to locate and name the destination of a hotspot. . We are illustrating the process here using ODS PDF, but you can generate the output as well with ODS LISTING using DEVICE= PDF or DEVICE=SVG (Scalable Vector Graphics).

Let’s say we want to display the population data above using a PROC GCHART and then drill down to a state map. . We would need to create the GMAP output for each of the states. Within each map we would need to have an HTML variable with the information for the state Web site. Then we would need to use PROC GCHART and add the ability to drill down to the state maps. . The HTML data for the PROC GCHART would have the HREF field to the specific state map. . When we are done we will have a chart of the state populations with links to the states and to the individual state Web site.

We will need to extend our data set *mydata* to include the HTML variables. . In this simple example we create two variables, one defining the PDF file we created of the selected state (statepdf), and the other to direct us to the state

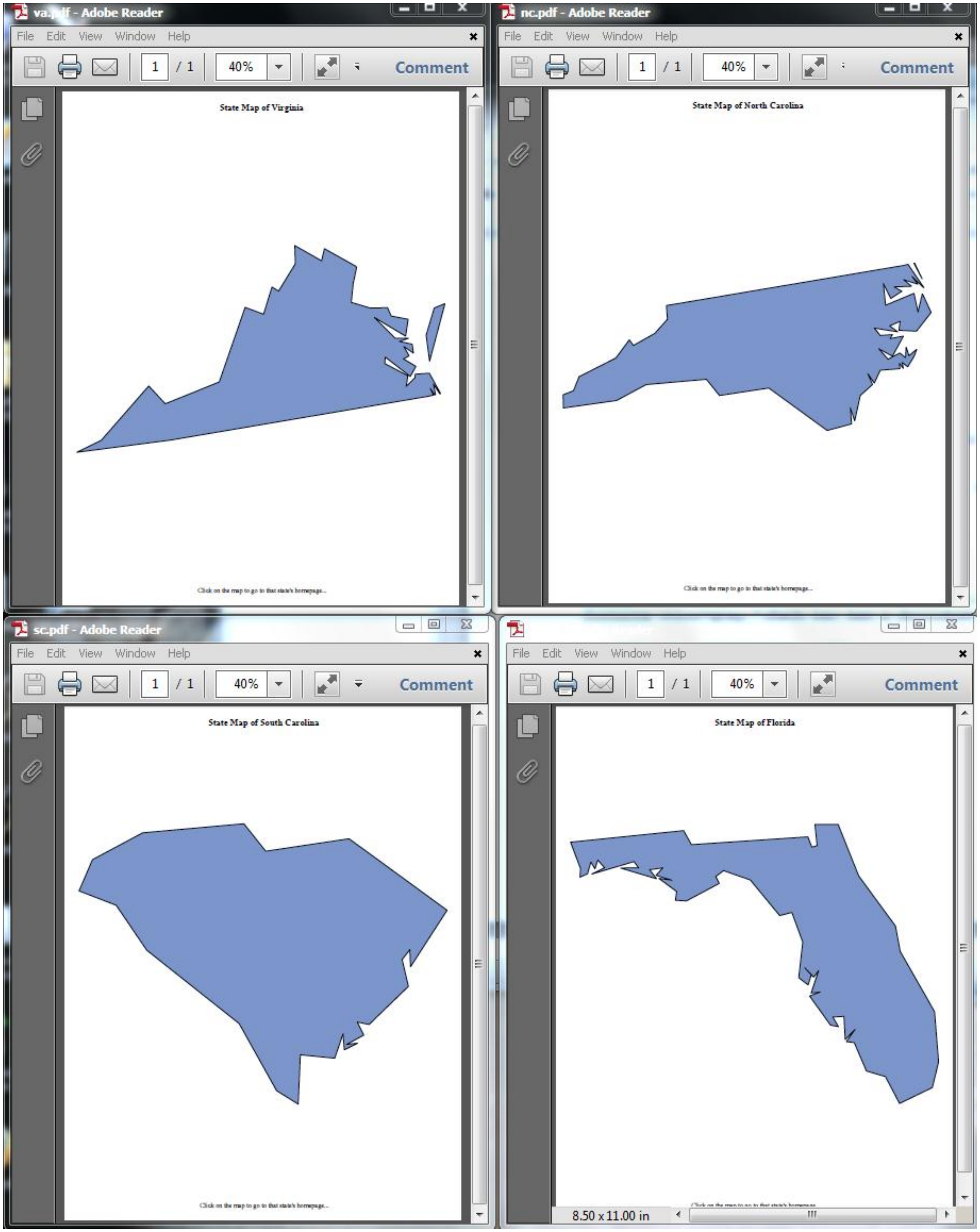
Web site (stateweb).

```
/* Add the HTML variables to the data */
data mydata;
   set mydata;

   /* This is the drill-down info for the bar chart */
   length statepdf $128;
      statepdf='href="' || trim(left(lowcase(st)))||'.pdf"';

   /* This is the drill-down info for the maps */
   length stateweb $128;
      stateweb='href="http://www.'||compress(lowcase(name),' ')||'.gov"';
run;
```

Now we are ready to create the maps for each of the states. . Since we are creating multiple states maps from the same data and with similar characteristics, we will create a macro to generate the state maps. . The macro creates each of the state maps with a unique name and uses the HTML variable in the data set to reference the state's Web site

```
%macro genstate(state, name);
filename out "&state..pdf";

title "State Map of &name";
footnote "Click on the map to go to that state's homepage...";

ods pdf notoc file=out;  /* NOTOC disables the table of contents */

proc gmap data=mydata map=maps.us
          (where=(state = stfips("&state")))
          imagemap=imap2;
id state;
choro state / levels=1 coutline=black nolegend
              html=stateweb des=""
              name="&state" ;
run;
quit;
ods pdf close;
%mend; /* genstate */
```

```
%genstate(fl, Florida);
%genstate(nc, North Carolina);
%genstate(sc, South Carolina);
%genstate(va, Virginia);
```

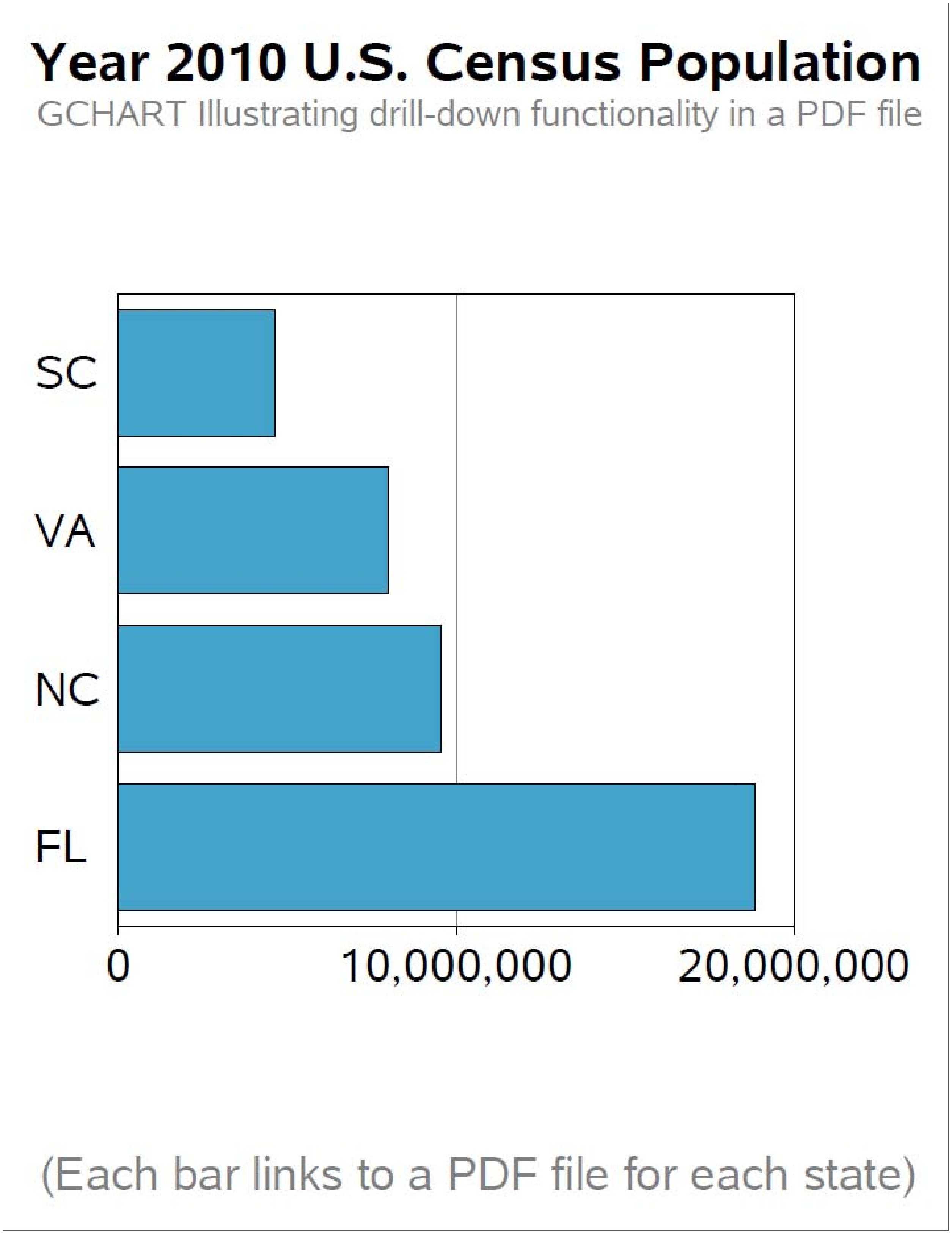

va.pdf - Adobe Reader
File Edit View Window Help
1 / 1
40%
Comment
State Map of Virginia
Click on the map to go to that state's homepage...
nc.pdf - Adobe Reader
File Edit View Window Help
1 / 1
40%
Comment
State Map of North Carolina
Click on the map to go to that state's homepage...
sc.pdf - Adobe Reader
File Edit View Window Help
1 / 1
40%
Comment
State Map of South Carolina
Click on the map to go to that state's homepage...
File Edit View Window Help
1 / 1
40%
Comment
State Map of Florida
8.50 x 11.00 in

Display 14. Four Southeastern United States Maps

Once we have the state maps we are ready to use PROC GCHART to add the population at a horizontal reference and the ability to drill to the state map.

```
goptions gunit=pct htitle=6 ftitle="arial/bo" htext=4 ftext="arial" noborder;

axis1 label=none;
axis2 label=none minor=none offset=(0,0);
pattern v=solid color=cx43a2ca;

title "Year 2010 U.S. Census Population";
title2 color=gray "GCHART Illustrating drill-down functionality in a PDF file";
footnote color=gray "(Each bar links to a PDF file for each state)";

ods pdf notoc file='gchart.pdf';

proc gchart data=mydata imagemap=imap;
hbar st / discrete
 type=sum sumvar=population
 ascending
 nostats
 maxis=axis1
 raxis=axis2
 autoref cref=gray
 clipref
 coutline=black
 html=statepdf
  ;
run;
quit;
ods pdf close;
```

Now that we have generated the output, let’s see how it looks. . This is the bar chart that is created.



Year 2010 U.S. Census Population

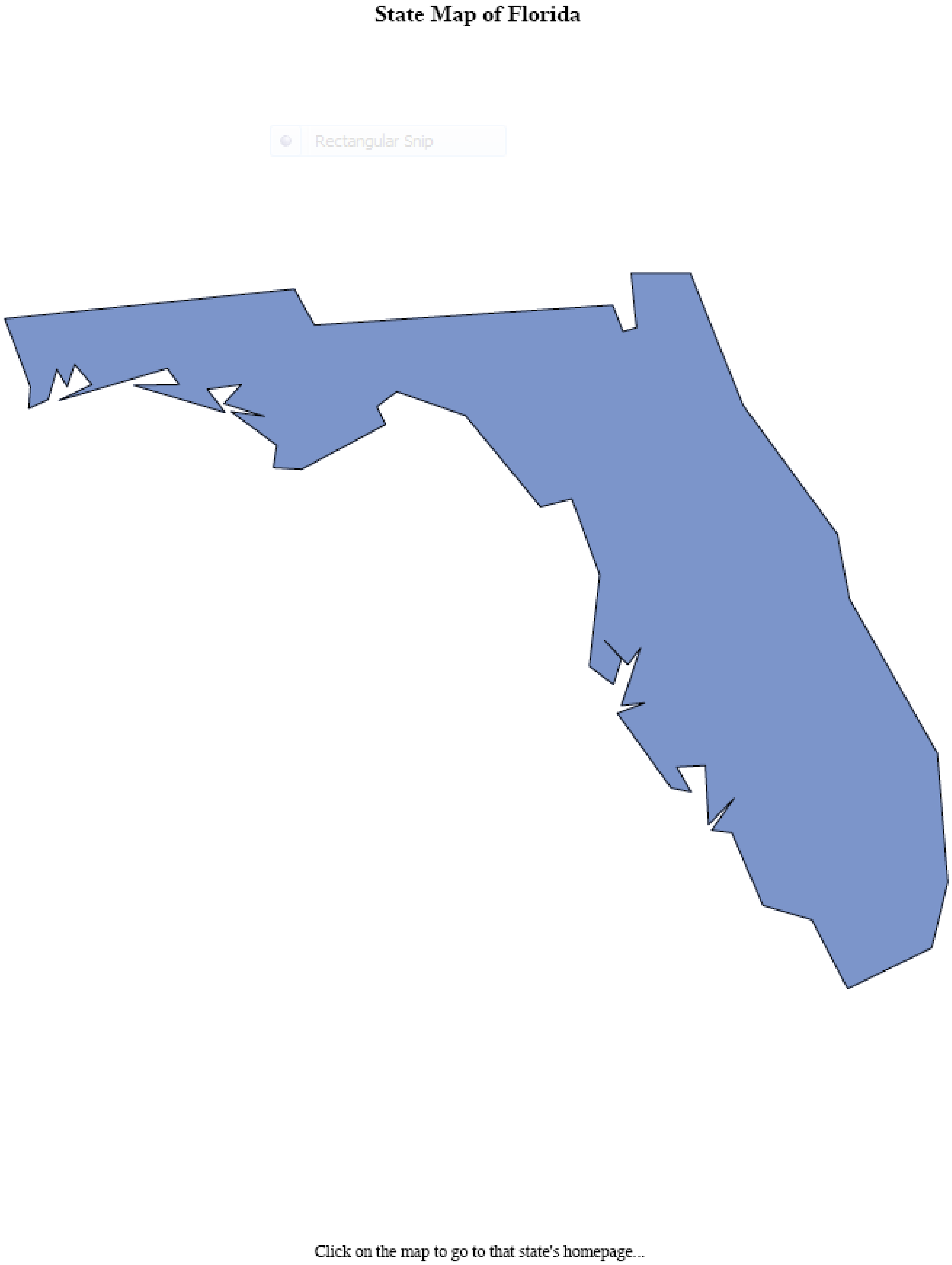
GCHART Illustrating drill-down functionality in a PDF file

(Each bar links to a PDF file for each state)

Display 15. . PROC GCHART with Population Data for Selected Southeastern States

To drill down to the state maps place the cursor over the state you want to view and click. (The destination name is displayed when you hover over the hotspot. We do not show the hover text in this paper.) Each of the states listed reroutes the symbolic name constructed in the html variable to the state Web site.

We will use the state of Florida for our example. Selecting the Florida population bar will generate the map of Florida.

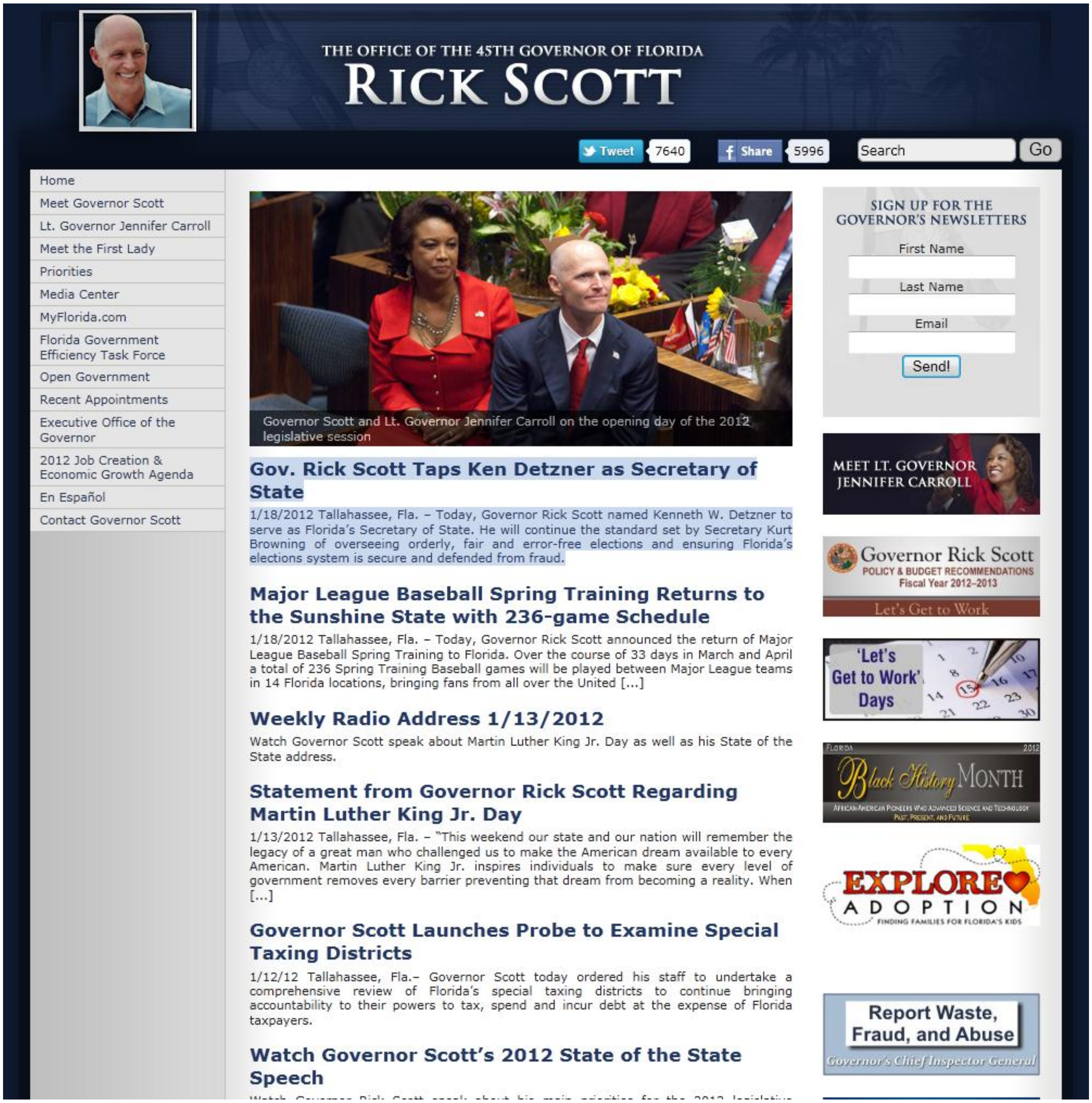


State Map of Florida

Click on the map to go to that state's homepage...

Display 16. Map of Florida

Clicking on the state map of Florida generates a link to the state Web page.


THE OFFICE OF THE 45TH GOVERNOR OF FLORIDA
RICK SCOTT
Tweet
7640
Share
5996
Search
Go
Home
Meet Governor Scott
Lt. Governor Jennifer Carroll
Meet the First Lady
Priorities
Media Center
MyFlorida.com
Florida Government Efficiency Task Force
Open Government
Recent Appointments
Executive Office of the Governor
2012 Job Creation & Economic Growth Agenda
En Español
Contact Governor Scott
SIGN UP FOR THE GOVERNOR'S NEWSLETTERS
First Name
Last Name
Email
Send!
MEET LT. GOVERNOR JENNIFER CARROLL
Governor Rick Scott
POLICY & BUDGET RECOMMENDATIONS
Fiscal Year 2012-2013
Let's Get to Work
'Let's Get to Work' Days
FLORIDA 2012
Black History MONTH
EXPLORE ADOPTION
FINDING FAMILIES FOR FLORIDA'S KIDS
Report Waste, Fraud, and Abuse
Governor's Chief Inspector General

Governor Scott and Lt. Governor Jennifer Carroll on the opening day of the 2012 legislative session

Gov. Rick Scott Taps Ken Detzner as Secretary of State

1/18/2012 Tallahassee, Fla. – Today, Governor Rick Scott named Kenneth W. Detzner to serve as Florida's Secretary of State. He will continue the standard set by Secretary Kurt Browning of overseeing orderly, fair and error-free elections and ensuring Florida's elections system is secure and defended from fraud.

Major League Baseball Spring Training Returns to the Sunshine State with 236-game Schedule

1/18/2012 Tallahassee, Fla. – Today, Governor Rick Scott announced the return of Major League Baseball Spring Training to Florida. Over the course of 33 days in March and April a total of 236 Spring Training Baseball games will be played between Major League teams in 14 Florida locations, bringing fans from all over the United […]

Weekly Radio Address 1/13/2012

Watch Governor Scott speak about Martin Luther King Jr. Day as well as his State of the State address.

Statement from Governor Rick Scott Regarding Martin Luther King Jr. Day

1/13/2012 Tallahassee, Fla. – "This weekend our state and our nation will remember the legacy of a great man who challenged us to make the American dream available to every American. Martin Luther King Jr. inspires individuals to make sure every level of government removes every barrier preventing that dream from becoming a reality. When […]

Governor Scott Launches Probe to Examine Special Taxing Districts

1/12/12 Tallahassee, Fla.– Governor Scott today ordered his staff to undertake a comprehensive review of Florida's special taxing districts to continue bringing accountability to their powers to tax, spend and incur debt at the expense of Florida taxpayers.

Watch Governor Scott's 2012 State of the State Speech

Display 17. Web Site for the State of Florida

With the ODS PDF statement we now have the capability of extending the traditional SAS/GRAPH output to include links to other PDF files as well as HTML sites. We can reference those links for any shaped object, including maps.

CONCLUSION

SAS 9.3 is full of new features and enhancements to make your output better. We have shown you some new points of view using the ODS PDF statement in SAS 9.3. We discovered we can change the orientation of our output within the same PDF file on each page. We can now control the graph title and place it inside or outside of our graph output. Using transparency you can add some new flair to your output. Drill-down functionality gives you more flexibility in your output as you can link SAS/GRAPH output to other PDFs or HTML websites. Used individually or together you will have new ways to show your audience different views of output using the ODS PDF statement. Try the code snippets within the paper and all the code listed in the appendix. Play with them in SAS 9.3 and see the results. Replace our data with your data and change the way your output looks. Show your audience all the new points of view that you can create with SAS 9.3

REFERENCES

Huntley, Scott. 2006. "Let the ODS PRINTER Statement Take Your Output into the Twenty-First Century." *Proceedings of the SAS Users Group International 31 Conference*. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/sugi31/227-31.pdf.

Huntley, Scott. 2008. "How to Add a Little Spice to Your PDF Output." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/forum2008/307-2008.pdf

ACKNOWLEDGMENTS

The authors would like to thank Kevin Smith, David Kelley, and Bari Lawhorn for their contributions to this paper.

CONTACT INFORMATION

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

Scott Huntley
SAS Institute Inc.
Building S, SAS Campus Drive
Cary, NC 27513
E-mail: Scott.Huntley@sas.com

Woody Middleton
SAS Institute Inc.
Building S, SAS Campus Drive
Cary, NC 27513
E-mail: Woody.Middleton@sas.com

APPENDIX

Code to Create Report with Transparent Headers (Display 6):

```
options nodate;

proc sort data=sashelp.stocks out=stocks; by date; run;


data splitIBM;

   length split $3;

   set stocks(where=(stock="IBM"));

   label volume='(millions)';

   volume=volume/1000000;

   if date < '01MAY97'd then do;

     open = open / 4;

        close = close/4;

        high=high / 4;

        low=low/4;

   end;
   else if date ='01MAY97'd then do;
     open = open / 4;
        close = close/2;    /*Anomaly in data    */
        high=high / 4;
        low=low/2;          /*Anomaly in data    */
        split="2/1";
        freq=1;
   end;
   else if date <'03MAY99'd then do;
     open = open / 2;
        close = close/2;
        high=high / 2;
        low=low/2;
   end;
   else if date ='03MAY99'd then do;
     open = open / 2;
        high=high / 2;
        split="2/1";
        freq=1;
   end;
   run;

data moveavg(drop=move25index move50index);
   set splitIBM;
   retain closehigh prevvolume;
   array move25[25] _temporary_;
   array move50[50] _temporary_;
   array vmove25[25] _temporary_;
   format close high low avg25 avg50 vavg dollar8.0;
   move25index=mod(_n_,25);
   move50index=mod(_n_,50);
   vmove25index=mod(_n_,25);
   move25index =ifn(move25index,move25index,25);
   move50index =ifn(move50index,move50index,50);
   vmove25index =ifn(vmove25index,vmove25index,25);
   move25[move25index]=close;
   move50[move50index]=close;
   vmove25[vmove25index]=volume;
   if _n_ ge 25 then do;
      avg25=mean(of move25[*]);
```

```
         vavg=mean(of vmove25[*]);
         bolupper= avg25 + (2*std(of move25[*]));
         bollower= avg25 + (-2*std(of move25[*]));
   end;
   if _n_ ge 50 then avg50=mean(of move50[*]);
   if _n_ gt 1 then do;
      if close gt closehigh then do;
           cindex = 1;
              closehigh = close;
         end;
      else do;
           cindex = 2;
              closehigh=closehigh;
         end;

      if volume gt prevvolume then
           vindex = 1;
      else
           vindex = 2;

      prevvolume=volume;
 end;
 else do;
    closehigh=close;
       prevvolume=volume;
       cindex=1;
       vindex=1;
 end;
run;

ods graphics / reset imagename="HighLow" imagefmt=png;
ods pdf file='nv_highlow.pdf' notoc dpi=600;

Title1 "Stock Plot";
proc sgplot data=moveavg (where=(date>='01jan2000'd));
   yaxis grid label="Stock Value";
   band x=date upper=bolupper lower=bollower /
            transparency=0.5 legendlabel="Bollinger Bands(25,2)" name="boll";
   vector x=date y=high / yorigin=low xorigin=date noarrowheads;
   scatter x=date y=close / markerattrs=(symbol=plus size=3);
   series x=date y=avg25 / lineattrs=GraphFit legendlabel="SMA(25)" name="d25";
   series x=date y=avg50 / lineattrs=GraphFit2(pattern=solid) legendlabel="SMA(50)"
name="d50";
   keylegend "boll" "d25" "d50" / across=4 noborder position=TopRight location=inside;
run;
ods pdf close;
```

Code to Create Report with Transparent Background (Display 13):

```
options nodate nonumber;
data passing;
input player $ 18. att comp yds comppct ydsatt td tdpct int intpct long sack sacklost
rating;
label player='Player'
      att='Att'
      comp='Comp'
      yds='Yds'
      comppct='Comp %'
      ydsatt='Yds/Att'
      td='TD'
      tdpct='TD %'
      int='INT'
      intpct='INT %'
      long='Long'
      sack='Sck'
      sacklost='Sack/Lost'
      rating='Rating';
datalines;
Harrison Edmunds   243 121 3391 49.7 13.9 21 8.6 12 4.9 73 36 175 94.7
William Sekel         4 3 33 75.0 11.0 0 0.0 0 0.0 28 0 0 107.2
;

data rushing;
input player $ 18. att yds ydsatt long td;
label player='Player'
      att='Att'
      yds='Yds'
      ydsatt='Yds/Att'
      long='Long'
      td='TD';
datalines;
Josh Brown                 84 390 4.6 88 2
```

```
Patrick Morris                 80 316 4.0 31 1
Alex Young                     72 356 4.9 43 4
William Sekel                  58 195 3.4 18 2
Harrison Edmunds               22 78 3.5 22 0
Grant Storey                   14 41 2.9 9 0
Steve Dolewick                 11 85 7.7 20 0
Liam Murphy                    9 32 3.6 16 0
Gregg Smith                    2 15 7.5 11 0
Conner Hester                  1 11 11.0 11 0
Christian Usher                1 5 5.0 5 0
Devin Smith                    1 0 0.0 0 0
Pearson Walker                 1 -1 -1.0 -1 0
;

data receiving;
input player $ 18. rec yds ydsrec long td;
label player='Player'
      rec='Rec'
      yds='Yds'
      ydsrec='Yds/Rec'
      long='Long'
      td='TD';
datalines;
Landon Smith                   26 1381 53.1 73 7
Brice Andrews                  20 677 33.9 47 4
Noah Brown                     14 357 25.5 47 3
Richard Hayden                 13 227 17.5 57 2
Paul Jones                     12 187 15.6 60 1
Wayne Royster                  12 147 12.3 36 2
Tim Davis                      12 130 10.8 16 1
Kevin Williams                 10 79 7.9 28 1
David Young                    5 62 12.4 25 0
Martin Johnson                 3 42 14.0 22 0
John Hunter                    3 20 6.7 10 0
Hunter Kelly                   2 63 31.5 33 0
Dominic Parker                 1 19 19.0 19 0
;

proc template;
define style stallions;
```

```
class body /
    fontfamily = arial
    backgroundcolor = #e0e0e0
    color = #000000
    backgroundimage = 'c:\sgf2012\stallions.png'
    backgroundcolor = #dadada
    backgroundrepeat = no_repeat
;
class table /
    bordercollapse = collapse
    borderspacing = 0
    borderwidth = 1px
    borderstyle = solid
    bordercolor = RGBA9b793130
;
class data, header, rowheader, tablehead, altrow /
    fontfamily = arial
    color = black
    borderwidth = 1px
    borderstyle = solid
    bordercolor = RGBAdcdcdc4c
    backgroundcolor = RGBAb0b0b04c
    paddingtop = 10px
    paddingbottom = 10px
    paddingleft = 18px    paddingright = 18px;
class header, rowheader, tablehead /
    fontsize = 1.1em
    fontweight = bold
    color = white
    backgroundcolor = RGBA3948a0cc
    fontfamily = arial
;
class tablehead /
    fontsize = 1.3em
    textalign = left
;
class systemtitle /
    preimage = 'c:\sgf2012\stallions-banner.png'
;
class systemtitle2 /
```

```
    preimage = _undef_
    fontfamily = 'arial black'
    fontweight = bold
    fontsize = 4em
    textalign = right
    paddingright = 1em
    color = RGBA3948a0ff
;
class altrow /
    backgroundcolor = RGBA9b793121
;
end;
run;

ods pdf file='stallions.pdf' style=stallions startpage=never;
title ' ';
title2 '2011 Offensive Statistics';

proc odstable data=passing;
define header head; text 'PASSING STATISTICS'; style=tablehead; end;
cellstyle mod(int((_row_-1)/2), 2) as altrow;
run;

title;

proc odstable data=rushing;
define header head; text 'RUSHING STATISTICS'; style=tablehead; end;
cellstyle mod(int((_row_-1)/2), 2) as altrow;
run;

proc odstable data=receiving;
define header head; text 'RECEIVING STATISTICS'; style=tablehead; end;
cellstyle mod(int((_row_-1)/2), 2) as altrow;
run;

ods _all_ close;
```

Code to Create Drill-Down Examples (Displays 14–17):

```
/*
  - Disable the page number and date as well as the border for graphics procedures -
```

```
*/
 options nodate nonumber;
 goptions noborder dev=sasprtc;

/* - Basic data set containing the state data - */
data mydata;
   format population comma12.0;
   input st $ 1-2 population;
   year=2010;
   state=stfips(st);
   name=stnamel(st);
datalines;
VA  8001024
NC  9535483
SC  4625364
FL 18801310
;
run;

/* Add the HTML variables to the data */
data mydata; set mydata;

/* This is the drilldown for the bar chart */
length statepdf $256;
statepdf='href="' || trim(left(lowcase(st)))||'.pdf"';

/* This is the drilldown for the maps */
length stateweb $256;
stateweb='href="http://www.'||compress(lowcase(name),' ')||'.gov"';
run;

/* - MACRO to generate the State Maps - */
%macro genstate(state, name);
filename out "&state..pdf";

title "State Map of &name";
footnote "Click on the map to go to that state's homepage...";

ods pdf notoc file=out;
```

```
proc gmap data=mydata map=maps.us
          (where=(state = stfips("&state")))
          imagemap=imap2;
id state;
choro state / levels=1 coutline=black nolegend
              html=stateweb des=""
              name="&state" ;
run;
quit;
ods pdf close;
%mend; /* genstate */
/* - Generate each of the states for the test case - */
%genstate(fl, Florida);
%genstate(nc, North Carolina);
%genstate(sc, South Carolina);
%genstate(va, Virginia);

/*- PROC GCHART Code to drill down to the state maps - */
goptions gunit=pct htitle=6 ftitle="arial/bo" htext=4 ftext="arial";

axis1 label=none;
axis2 label=none minor=none offset=(0,0);
pattern v=solid color=cx43a2ca;

title "Year 2010 U.S. Census Population";
title2 color=gray "GCHART Illustrating PDF Drilldown";
footnote color=gray "(Each bar has a drilldown to a PDF file for each state)";

ods pdf notoc file='gchart.pdf';

proc gchart data=mydata imagemap=imap;
hbar st / discrete
 type=sum sumvar=population
 ascending
 nostats
 maxis=axis1
 raxis=axis2
 autoref cref=gray
 clipref
 coutline=black
 html=statepdf
  ;
run;
quit;
ods pdf close;
```