

An Insider's Guide to ODS LAYOUT Using SAS® 9.4

Scott Huntley, SAS Institute Inc., Cary, NC

ABSTRACT

Whenever you travel, whether it's to a new destination or your favorite vacation spot, it's nice to have a guide to assist you with planning and setting expectations. The ODS LAYOUT statement became production in SAS® 9.4. For those intrepid programmers who used ODS LAYOUT in an earlier release of SAS® this paper contains tons of information about changes you need to know about. Programmers new to SAS 9.4 (or new to ODS LAYOUT) need to understand the basics. This paper reviews some common issues customers have reported to SAS Technical Support when migrating to the LAYOUT destination in SAS 9.4 and explores the basics for those who are making their first foray into the adventure that is ODS LAYOUT. This paper discusses some tips and tricks to ensure your trip through the ODS LAYOUT statement will be a fun and rewarding one.

INTRODUCTION

When I go on vacation, I spend countless hours researching hotels, attractions, and restaurants I plan on visiting. I rely on those who go before me to provide insight to what I might experience. With this insight I feel confident and comfortable knowing what to expect at my destination. This paper covers several insider tips for using ODS Layout to help you make your ODS Layout programs output a five-star experience

BACKGROUND

This paper focuses on ODS Layout for intermediate programmers. This paper is not a primer or beginner's "how to" paper for programmers who are new to ODS Layout. For more detail on the syntax and specifics on ODS Layout, check out some previous SAS Global Forum papers (O'Connor 2013, Kummer 2014). The topics included in this paper emerged from customer questions posed to SAS Technical Support (TS). I used the TS tracking system and looked for common customer questions and problems encountered by SAS coders who upgraded from previous releases of SAS and started using ODS Layout in production code in SAS 9.4. This paper addresses as many of those common issues as possible. This paper provides helpful tips and tricks to assist users with existing ODS Layout programs make their SAS 9.4 layout work be the best results ever.

To protect the innocent I am using fictitious names for my insider tips.

INSIDER TIP #1 ... DRAW A MAP OF YOUR CONTAINERS

Peter has an excellent tip when starting your SAS Llayout program: use the style attribute "background=" on the ODS LAYOUT and REGION statements to understand where everything is located in your output. This syntax is new in SAS 9.4. Here is sample layout code with a layout container that holds two regions. Notice the "background=" syntax on the LAYOUT and REGION invocations. This enables Peter to have color as a background on his containers. PROC TEMPLATE logic is used to set the available space on the page to light blue.

Look at this code snippet:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options number nodate;
title 'title goes here';
```

```
footnote ' footnote goes here';
ods pdf style=color;
ods layout start height=9.5in width=7in style={background = yellow};

ods region x=1in y=1in height=3in width=5in style={background = lightpink};
PROC REPORT DATA = sashelp.class(obs=5) nowd;run;

ods region x=1.5in y=5in height=4in width=4in style={background = green};
PROC REPORT DATA = sashelp.class(obs=5) NOWD;run;

ods layout end;
ods _all_ close;
```

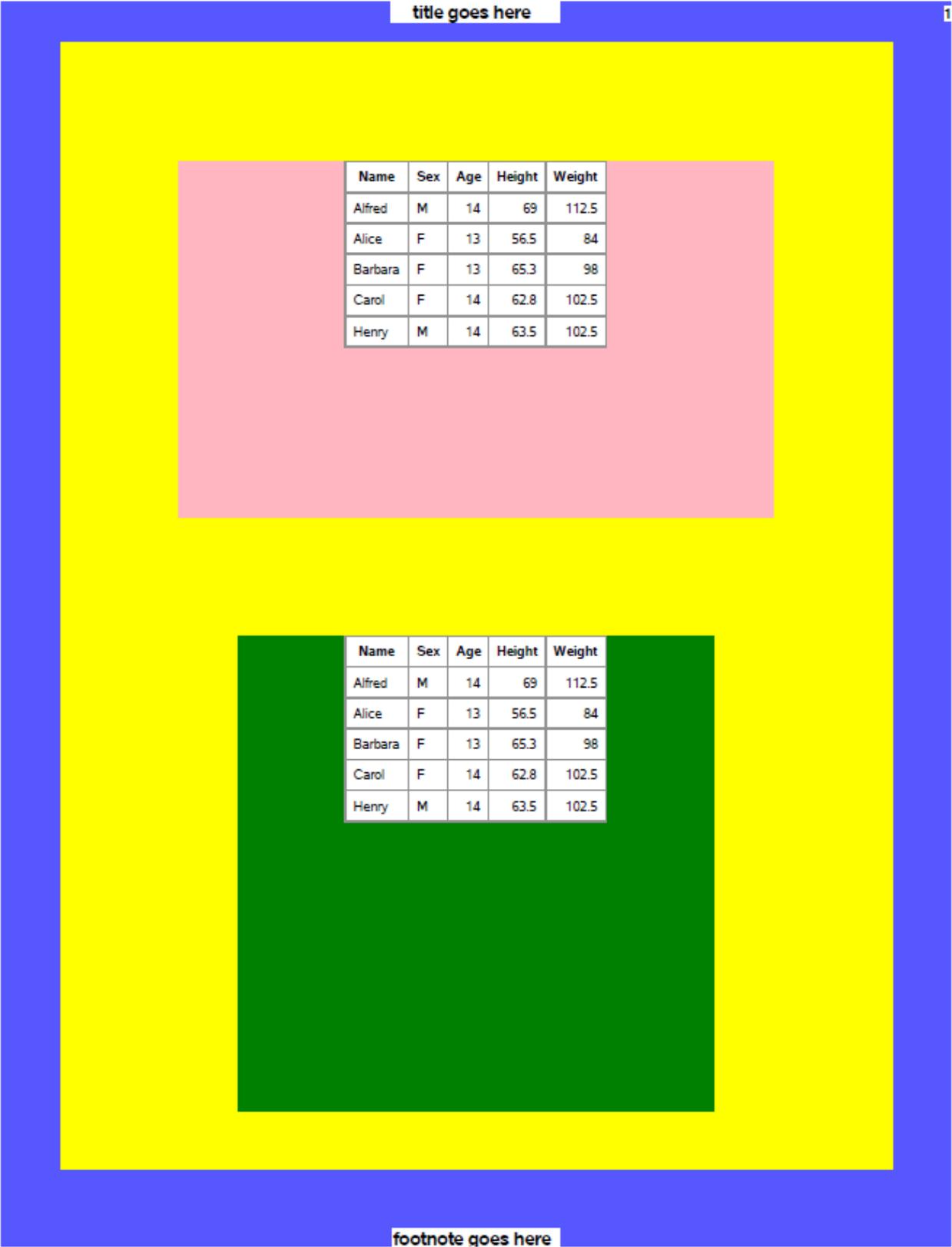


Figure 1. Simple Layout Output

The blue signifies the total available space on the page. The title, footnote, and page number are located in this area. The yellow shows the allotted space for the layout container. The pink and green colors signify the two regions that were defined.

Using this strategy enables Peter to see exactly what space that he has available and where output will be placed on a page.

Peter normally starts out with his output like this and then removes the “background=” attribute when he is satisfied with his Layout results.

INSIDER TIP #2 ... UNDERSTAND HOW YOUR MARGINS AFFECT THE AVAILABLE SPACE YOU HAVE IN YOUR LAYOUT

Mark is a longtime SAS user who recently made a trip to SAS 9.4 ODS LAYOUT. He wanted to create a single page of information and thought using Absolute Layout would do the trick. Absolute Layout is a perfect way to display information that fits exactly on one page. He initially noticed some problems with his width setting in his SAS log:

```
ods layout start width=11in height=8.5in;  
WARNING: WIDTH exceeds available space for PDF destination. This option will  
be ignored.
```

The settings looked valid to Mark. The height and width settings are basically a normal page size using landscape for his orientation. So why is he seeing the warning message?

The first thing Mark needs to check is to determine if the width he specified will fit on his page. However, the message is fairly generic because it doesn't tell him why his sizing doesn't fit and how to correct the problem.

One common mistake when setting up available space in ODS Layout is trying to capture as much of the page as possible. In this example Mark did not take into account his margins that are set on the file, which affect the space available on each page. By default, using the letter papersize, a top, bottom, right, and left margin is set to .25 inch. So the available work area on a standard PDF page set to landscape orientation will be 10.5 inches wide by 8 inches high. Mark can do two things here to fix the warning message. He can either reset his layout size or set the page margins to be smaller. Working with ODS Layout involves a lot of trial and error. It's good to get out a piece of paper and a ruler and measure your layout container and region containers. ODS Layout tries to provide pixel-perfect placement on the output page. Therefore, exact measurement settings are crucial, especially when you want to maximize your entire page. But there are other things we need to consider as well when defining a work area. What about any possible titles or footnotes? Those also need space on the page. Will there be text statements or other output objects between our layout containers?

Mark really wants the maximum space available on a page. In this case he will not have any titles and footnotes. What if Mark tries the OPTIONS statement before his ODS PDF file= statement?

```
options leftmargin=0in rightmargin=0in topmargin=0in bottommargin=0in;
```

That should give him the maximum space, right? Caution, Mark, this is a trick question. Setting the margins to zero will force the PDF destination back to the default values of a quarter of an inch. ODS Printer (PDF) MUST have a value for the margins. If any of the values are set to zero, the destination looks at the Printer Registry defaults and uses the margin setting there. ODS Printer needs margins that are greater than zero. So for Mark, this options statement is perfect:

```
options leftmargin=.001in rightmargin=.001in topmargin=.001in  
bottommargin=.001in;
```

This will give Mark the most space available for his output. His initial height and width setting will work and he will not see a message in his log. By not specifying any width or height on the Absolute Layout container, it will use all available space on the page by default.

INSIDER TIP #3 ... YOUR UNIT OF MEASURE MATTERS

Joan is an intrepid SAS 9.3 ODS LAYOUT coder who spent lots of time coding without documentation. She sends us another tip for regions and margins. She says it's best to use inches instead of percentages when defining a region. For example, the following ODS REGION statement is a valid way to set up a region inside an ODS Layout:

```
ods region x=75pct y=18pct height=40pct width=25pct;
```

However, for more precise results, it is better to use units of measure, like centimeters or inches. To repeat our earlier pixel placement statement: ODS Layout tries to provide pixel-perfect placement on the output page, so using inches (or another unit of measure) enables you to use a ruler on a page and determine exactly where you want the output to be placed.

```
ods region x=5in y=5.5in height=5in width=3in;
```

Using the ODS REGION statement Joan can easily see that the region will be 5 inches from the left margin and 5.5 inches down from the top margin. The region will have a height of 5in by a width of 3in. It's easy for her to measure and visualize.

INSIDER TIP #4 ... PREVENT TABLE ALIGNMENT WOES

Jim has defined two regions with the same table structure in both regions. The first region contains the first five observations from the table while the second region has the next ten observations. What he notices is that his tables are not aligning as he thought they should. His region settings use units of measure and his PROC REPORT code matches. Here is his code:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options nonumber nodate;
title 'reports are different widths';
ods pdf file='quick.pdf' style=color;
ods layout start height=10in width=7in style={background = yellow};

/* 1st region */
ods region x=1.50in y=0in height=4in width=5in
style={background=lightpink};
PROC REPORT DATA = sashelp.cars(obs=5) NOWD;
column make model type origin drivetrain MSRP invoice;
run;

/* 2nd region */
ods region x=1.50 in y=5in height=4in width=5in
style={background=lightgreen};
PROC REPORT DATA = sashelp.cars(firstobs=6 obs=15) NOWD;
column make model type origin drivetrain MSRP invoice;
run;

ods layout end;
ods _all_ close;
```

Here is the output:

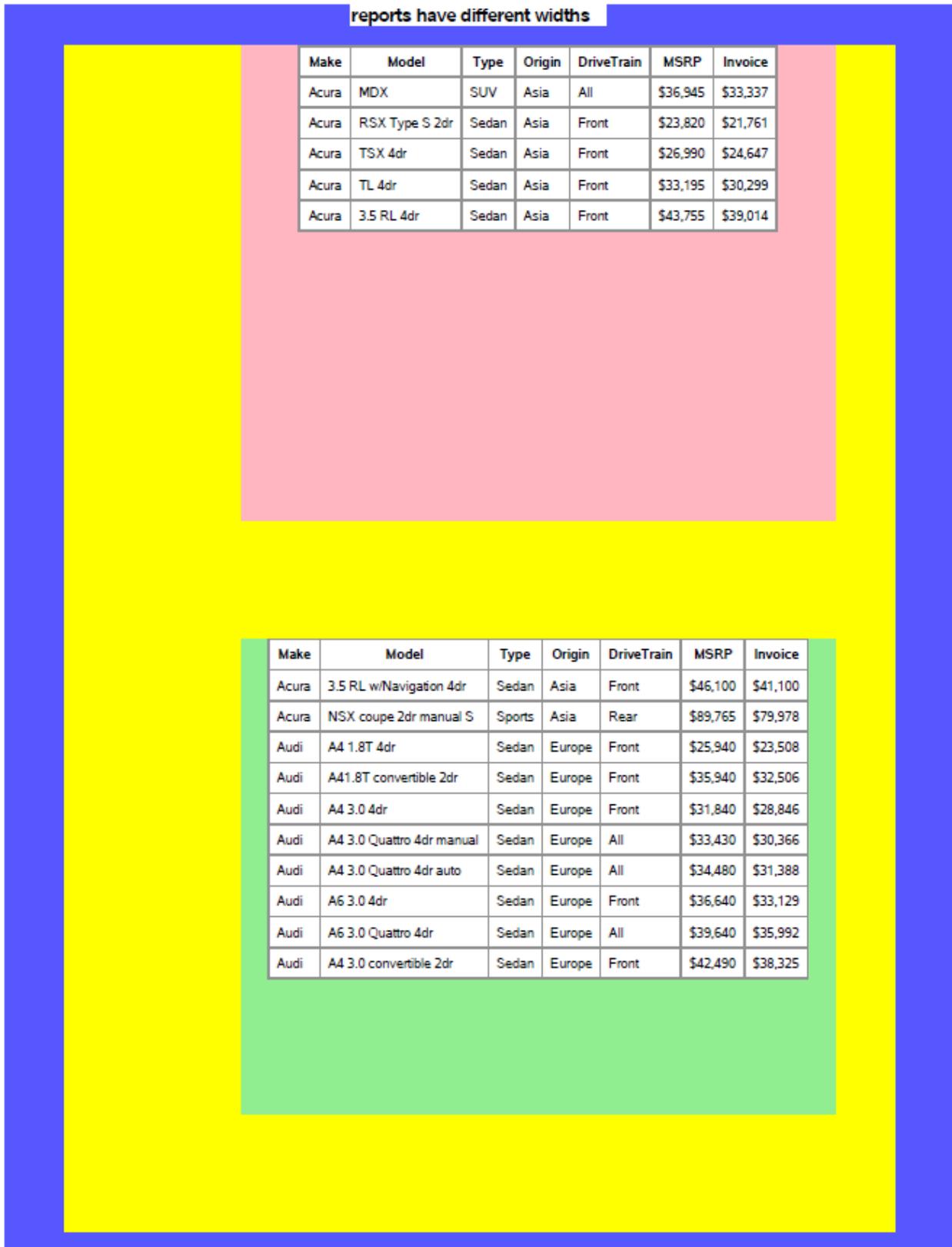


Figure 2. Reports with Different Widths

Jim expects region one and region two to look identical since the regions contain an identical height and width. The only difference is that region one starts at y=0 and region two starts farther down the page at y=5in. So why do the tables inside the regions look different? Why are they not uniform?

This is not a layout issue but a table generation issue. Cell widths within a table are determined by the data inside the cells. In this case the values for the Model variable just happen to have a longer width for those observations in region two than in region one. Therefore, the table width in region two is wider than in region one. How do you fix this? Jim found two simple solutions. First, he found that he could use the WIDTH style attribute in the PROC REPORT statement to set the total width of the table within the region.

```
PROC REPORT DATA = sashelp.cars(obs=5) NOWD style(report)=[width=5in];
```

Alternatively, he found that he could set the style attribute CELLWIDTH on individual columns that might have variable length data.

```
PROC REPORT DATA = sashelp.cars(firstobs=5 obs=10) NOWD;  
column make model type origin drivetrain MSRP invoice;  
define make / STYLE(column)=[cellwidth=0.5in];  
define model / STYLE(column)=[cellwidth=1.5in];  
run;
```

reports have same widths

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice
Acura	MDX	SUV	Asia	All	\$36,945	\$33,337
Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761
Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647
Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299
Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice
Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	\$46,100	\$41,100
Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	\$89,765	\$79,978
Audi	A4 1.8T 4dr	Sedan	Europe	Front	\$25,940	\$23,508
Audi	A4 1.8T convertible 2dr	Sedan	Europe	Front	\$35,940	\$32,506
Audi	A4 3.0 4dr	Sedan	Europe	Front	\$31,840	\$28,846
Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	\$33,430	\$30,366
Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	\$34,480	\$31,388
Audi	A6 3.0 4dr	Sedan	Europe	Front	\$36,640	\$33,129
Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	\$39,640	\$35,992
Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	\$42,490	\$38,325

Figure 3. Reports with Same Widths

By using one of these strategies Jim can make his tables align, and by using the same size in each region for his tables will ensure consistent appearance. A warning from Jim ... make sure your table width or cell widths do not make your tables **wider** than your region or layout container. If your tables are wider than your region or layout container, your table might panel or truncate data. If the region or layout space is too small you will see the following warning message in the log:

```
WARNING: The absolute region was too small to accommodate the output supplied
for the PDF destination. Output will be suppressed.
```

INSIDER TIP #5 ... PAGE BREAKS ARE DIFFERENT IN SAS 9.4

Prior to SAS 9.4 ODS Layout was in pre-production status. As we in R&D developed the software and made it more mature, lots of features might have changed. Things that worked in previous releases might appear different in the production version. Pagination is one thing that caught John off guard as he migrated to SAS 9.4. John had a simple program that had two ODS Layout statements inside one SAS program, as shown in the following code:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;
  title;
  ods pdf style=color;
  options nodate;

  ods layout start width=8in height=10in style={background=yellow};

  ods region y=0in height=4in width=4in style={background=lightpink};
  proc print data=sashelp.class(obs=3);run;

  ods region y=3in height=4in width=4in style={background=lightgreen};
  proc print data=sashelp.class(obs=3);run;
  ods layout end;

  ods layout start style={background=yellow};
  ods region style={background=yellow};
  proc print data=sashelp.class(obs=3);run;
  ods layout end;

  ods _all_ close;
```

As you can see it has a layout with two regions defined inside. Each region has one PROC PRINT inside. Then a new layout starts with another region defined with one PROC PRINT. In SAS 9.3 he got two pages of output. Here is page 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

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

Figure 4. Page 1 of 2 of SAS 9.3 Layout

Here is page 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

Figure 5. Page 2 of 2 of SAS 9.3 Layout

When he ran the same job in 9.4 he only got one page of output. In addition, you see the "background=" attribute on the layout and region containers in SAS 9.4.

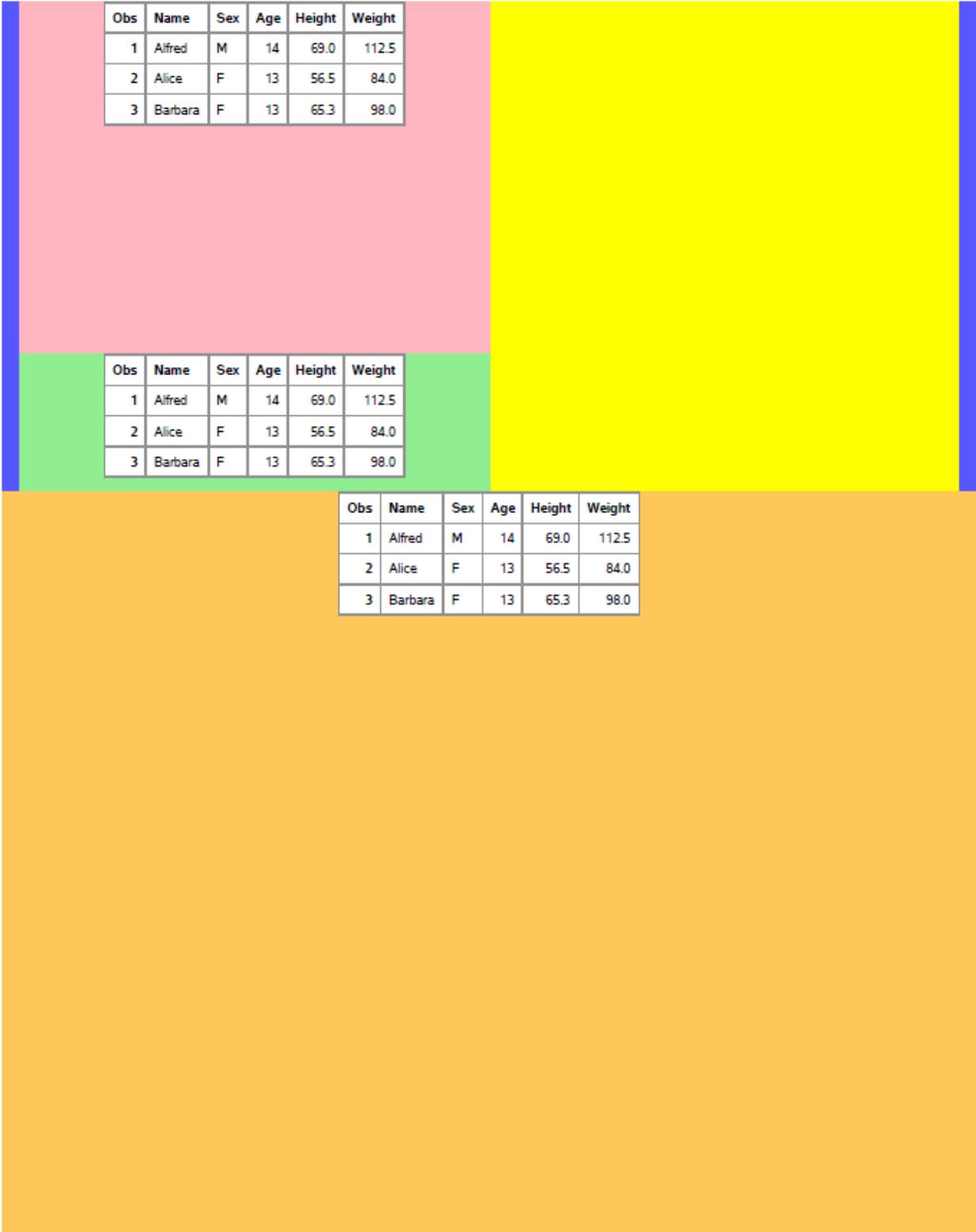


Figure 6. Simple SAS 9.4 Layout

So what changed between SAS 9.3 and SAS 9.4? Within the sample program you see two ODS Layout start and end paired statements. In SAS 9.3 ODS Layout automatically sent the second ODS Layout start statement to another physical page. That is why the third PROC PRINT is on page 2 in SAS 9.3.

During the versions when ODS Layout was in pre-production status, R&D received lots of input from users through Technical Support about what was missing or needed improvement. One thing we heard is that SAS users wanted the ability to have multiple layouts on one page. In order to make this happen we had to give the users the ability to control their own pagination.

John easily added one line of code to his program to replicate his SAS 9.3 output. A quick “ods startpage” statement was the key after his “layout end” and before the start of his 2nd layout.

```
ods layout end;
ods startpage = now;
ods layout start;
```

The statement “ods startpage=now” tells ODS Printer that you want a new page right before the next output. Going forward John tells us that when you use ODS Layout you control everything, from placement of the output on a page to what page you want output to be located.

INSIDER TIP #6 ... DISPENSE WITH STARTPAGE=NEVER

While on the topic of pagination Emily submits a quick and easy tip. She says there is no need to use “ods startpage=never” with ODS Layout. As mentioned above, using ODS Layout gives you complete control over everything, including pagination. “Never” used to be the way to determine whether graphs and tables went to the next page of output in ODS PDF. Using “never” could have some negative side effects at SAS 9.4. Emily had to change some of her SAS 9.3 programs by removing the “startpage=never” snippets when she migrated to SAS 9.4.

INSIDER TIP #7 ... STAY POSITIVE

Sam has an absolute ODS Layout program that created perfect results for him in SAS 9.2. Absolute layout allowed him to put everything on one page as he wanted. Sam used negative numbers in his ODS REGION statement to left-justify his output. He started his layout with a standard 8in x 10in size setting. Then he wanted two regions: one to contain an SGPLOT image and the other to contain a simple REPORT table. By setting the region x coordinate to a negative number in region 2 he could align the start of the table in region 2 with the graph in region 1.

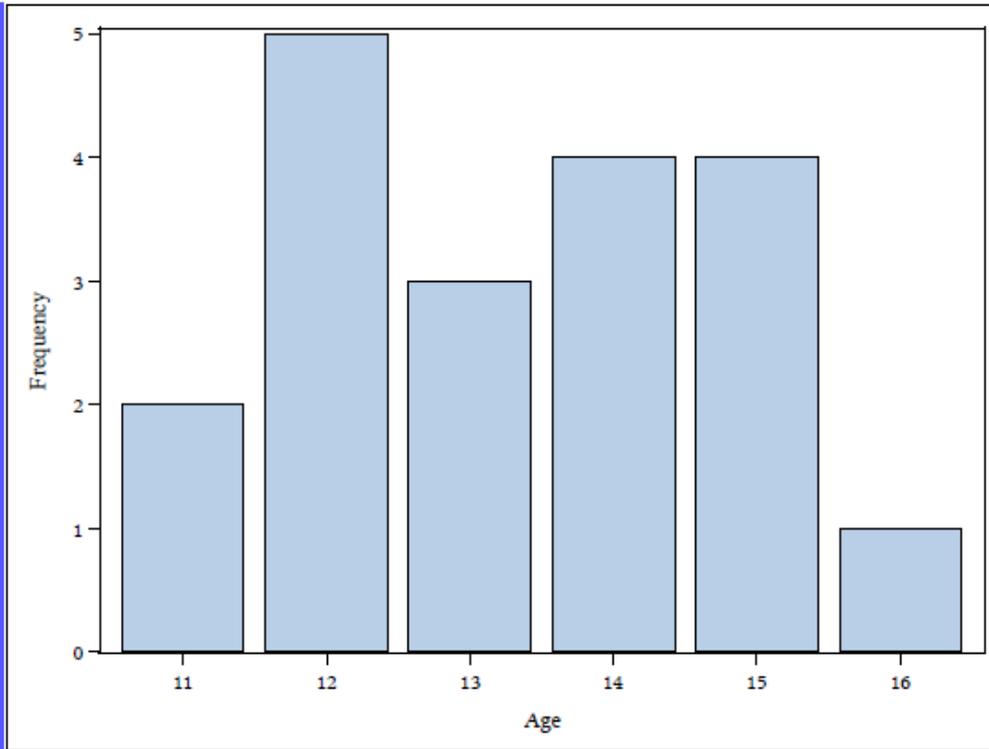
```
proc template;
  define style color;
    parent = Styles.printer;
    style body from body /
      backgroundcolor=light blue;
  end;run;

ods _all_ close;
options leftmargin=0.1in rightmargin=0.1in nodate nonumber;
title;
ods pdf style=color;
ods layout start width=8in height=10in style={background=yellow};
ods region x=0in y=0in height=5in;
proc sgplot data=sashelp.class;
vbar age;
run;

ods region x=-4.0in y=5.25in;
proc Report data=sashelp.class(obs=5) nowd;
run;

ods layout end;
```

```
ods _all_ close;
```



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

Figure 7. SAS 9.3 Using Negative Numbers

Running the same program in SAS 9.4 generates the following log warning:

```
ods region x=-4.0in y=5in;
```

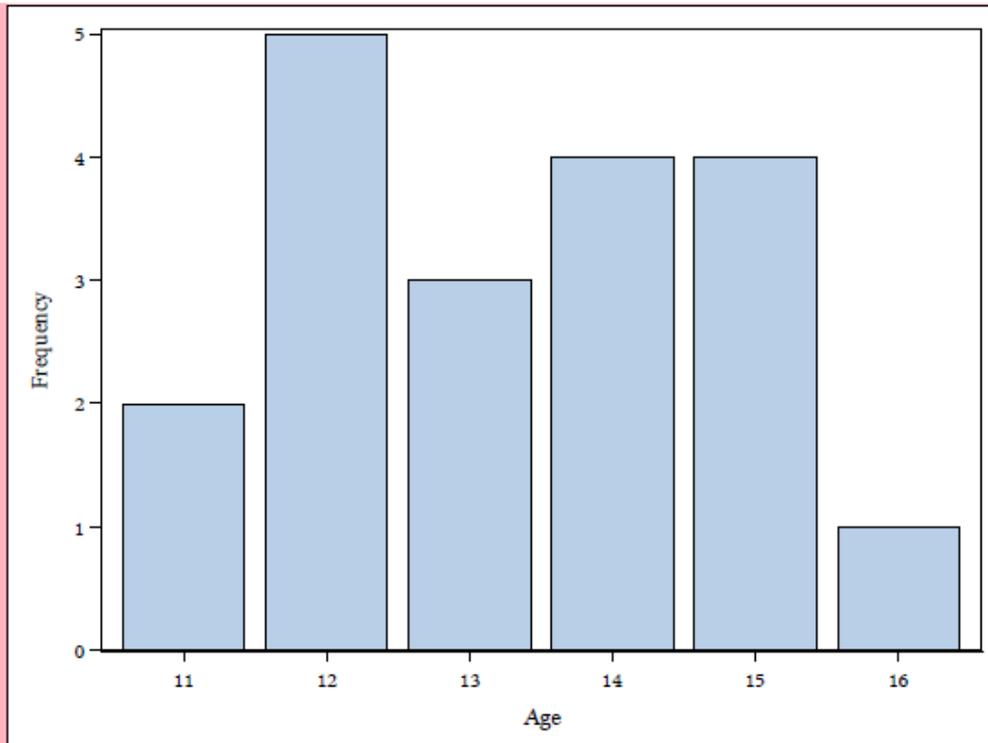
WARNING: Invalid X specified for PDF destination. This option will be ignored.

And his PROC REPORT is centered under his SGPLOT image.

In SAS 9.4 layout became production and part of that process removed or fixed previous problems. The use of negative numbers is no longer needed to align the plot and REPORT table. Sam removes the negative number from the x coordinate on the region statement to eliminate the warning. Second, Sam can get what he wants by making a few changes. He changes the PROC REPORT code to left-justify the table within the region by using the inline style attribute "just=". He then uses a positive x coordinate on his REGION statement to move the region to the right. By moving the region, the table inside also moves, so his results match his SAS 9.2 output.

Here is the code snippet and output results from SAS 9.4:

```
ods region x=0.75in y=5.25in;  
proc Report data=sashelp.class(obs=5) nowd style(report)={just=1};  
run;
```



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

Figure 8. Simple Layout Output Moving Region to Right

Sam can further adjust the x coordinate to move the region to the left or right as needed. Notice how the second region, which has a green background is offset to the left of the first region. This illustrates how in Absolute Layout you have complete control over where your regions are located. The first region starts at x=0 inch while the second region is set to x=0.75inches. This illustration helps reinforce how important the x and y settings are when allocating your region statements in your output.

Another strategy to match the previous version's results is to set the NOCENTER system option. In SAS 9.4, if Sam changes his x coordinate on his 2nd region statement to zero, then both the graph and table will be left-justified inside their respective regions.

INSIDER TIP #8: I KNOW ODS ABSOLUTE LAYOUT IS NOT SUPPORTED IN THE HTML DESTINATION, THEN WHY AM I SEEING: NOTE: WRITING HTML BODY FILE: SASHTML.HTM?

Mary noticed the following new warning message in her SAS 9.4 log file that did not appear in SAS 9.2.

```
NOTE: Writing HTML Body file: sashtml.htm
```

```
WARNING: Absolute layout is not supported for HTML destination. Layout will be ignored.
```

Mary had to think, did I ask for HTML in my SAS program? Starting in SAS 9.3 the HTML destination became the default destination for interactive SAS with the exception of MVS. The message is valid but not applicable to this program. A very simple fix is to user either of these statements before your ODS LAYOUT code:

```
ods html close;
or
ods _all_ close;
```

Both will close the HTML destination and will remove the messages that Mary saw from the log. If Mary is a SAS® Enterprise Guide user she might see this error:

```
ERROR: Absolute layout is supported only for ODS PRINTER. Non-COLUMN/ROW
LAYOUT requests will be ignored for other destinations.
```

In which case the latter statement is helpful, since the default destination in SAS Enterprise Guide is SASReport.

Make sure all unwanted destinations are closed before ODS Layout statements are submitted.

INSIDER TIP #9: HOW DO GRAPHS WORK WITHIN ODS LAYOUT?

Jacques likes the ability to combine graphs and tables on the same page using ODS Layout. His question is: to control the size of my graphs, do I use GOPTIONS with my SAS/GRAPH procedures, or ODS GRAPHICS with my SG procedures, OR do I use the height and width options in the ODS REGION statements? If I use ODS Layout, do I use the region settings or the layout settings? As usual it depends on what your layout looks like. Let's start with Absolute Layout. Using a simple program let's create a vertical bar plot with SGPLOT and see how it is sized:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options nodate nonumber;
title 'simple sgplot';
footnote 'simple footnote';
```

```
ods pdf file='test.pdf' nogtitle nogfoot style=color;
ods layout start width=8in height=9in style={background=yellow};

ods region style={background=lightpink};
proc sgplot data=sashelp.class;
vbar age;run;

ods layout end;
ods _all_ close;
```

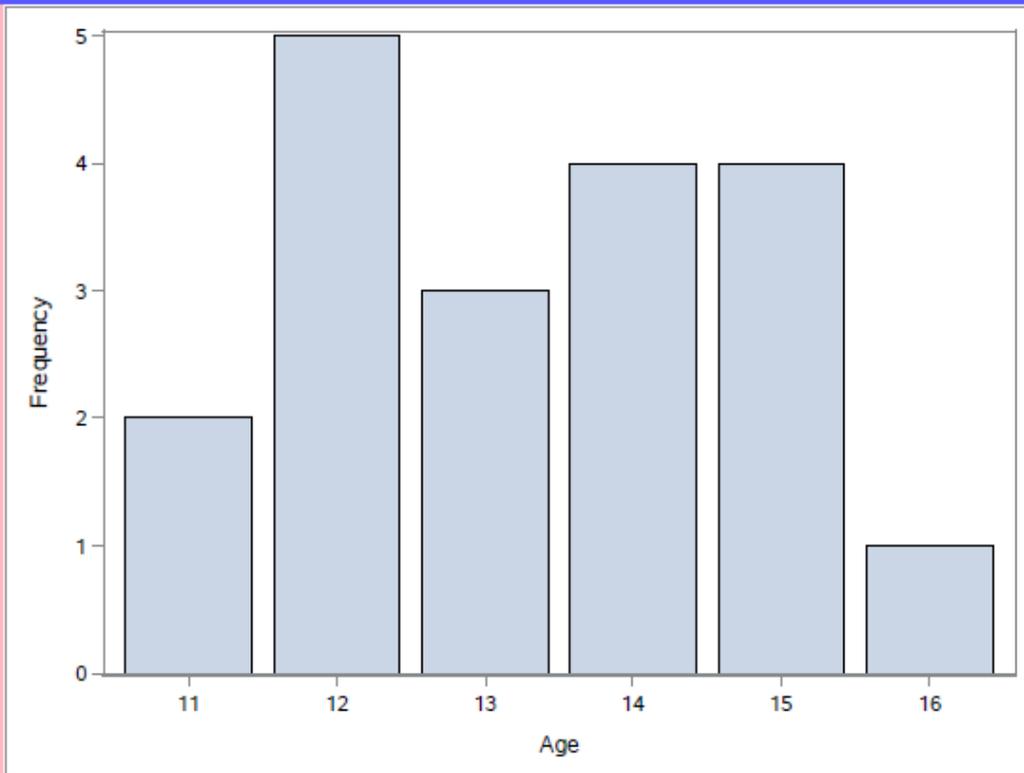


Figure 9. Simple Layout Output with SGPLOT

Notice in the program that the only height and width setting is on the LAYOUT START statement. The LAYOUT size has been specified but not the size of the plot. In this case we are letting SAS determine the size of the plot. Statistical Graphs use a graph template behind the scenes to control sizing. By letting SAS use the defaults, you are not in control, which is what ODS Layout is all about. In order to maximize that control you can do the sizing of output by yourself. Now we will add a height and width to a region and then add another region with another plot in it. Here is the code:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options nodate;
ods _all_ close;
ods graphics / reset=all;
title 'two sgplots';
footnote 'footnote goes here';
ods pdf file='r.pdf' nogtitle nogfootnote style=color;
ods layout start width=8in height=9in style={background=yellow};

ods region height=4in width=4in style={background=lightpink};
proc sgplot data=sashelp.class;
vbar age;
run;

ods region height=3in width=3in style={background=lightgreen};
proc sgplot data=sashelp.class;
vbar sex;
run;

ods layout end;
ods graphics off;
ods _all_ close;
```

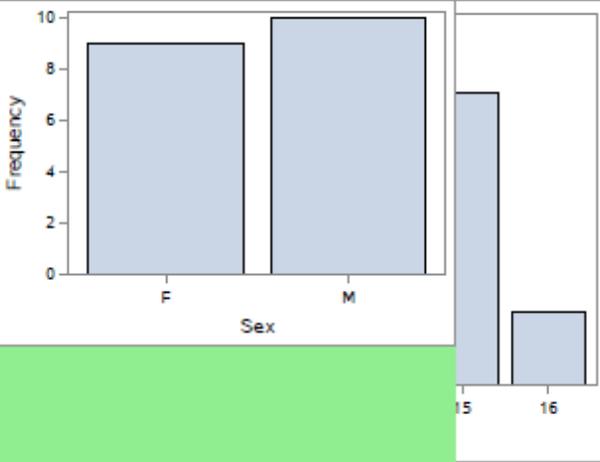


Figure 10. Simple Layout Output with Overlapping Plots

There are a few things that we can learn from this code. Let's start with the overlapping regions. We are in Absolute Layout. Since we did not specify an x or y coordinate with either region, both regions start at the exact same location (x=0, y=0) with every invocation of 'ods region', which is the top-left corner of the layout container. To fix this we can easily add 'y=5in' to our 2nd REGION statement causing the 2nd region to start 5 inches from the top. If you print out this PDF you can measure each region and see that they are exactly 4x4 inches and 3x3 inches. Again, ODS Layout is pixel-perfect placement, so you know the exact size of each region. The biggest lesson we learned is that graphs are sized automatically to fit inside our regions. Jack asks... 'but why aren't the graphs the full size of the region?' In order to fit the graph in the region we had to scale down the height and width. In order to retain the aspect ratio of the graph as we lower one dimension, the other dimension decreases accordingly if needed.

Now let's do another snippet of code with two regions not overlapping and with height and width specified in the ODS GRAPHICS statement. This forces the size of the graphs within the region no matter how big the regions.

```
title '2 sgplots and no overlap';
footnote 'footnote goes here';
ods layout start width=8in height=9in style={background=yellow};

ods graphics / reset=all height=4in width=4in;
ods region style={background=lightpink};
proc sgplot data=sashelp.class;
vbar age;
run;

ods graphics / reset=all height=3in width=3in;
ods region y=5in style={background=lightgreen};
proc sgplot data=sashelp.class;
vbar sex;
run;

ods layout end;
ods graphics off;
```

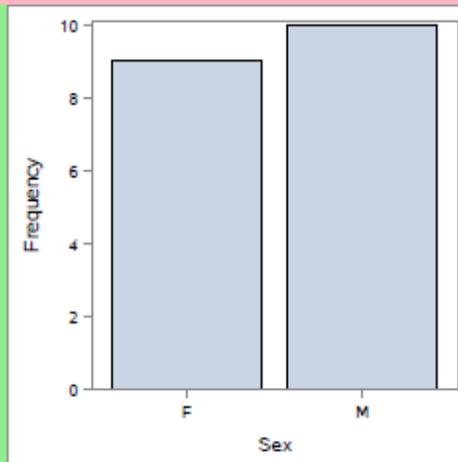
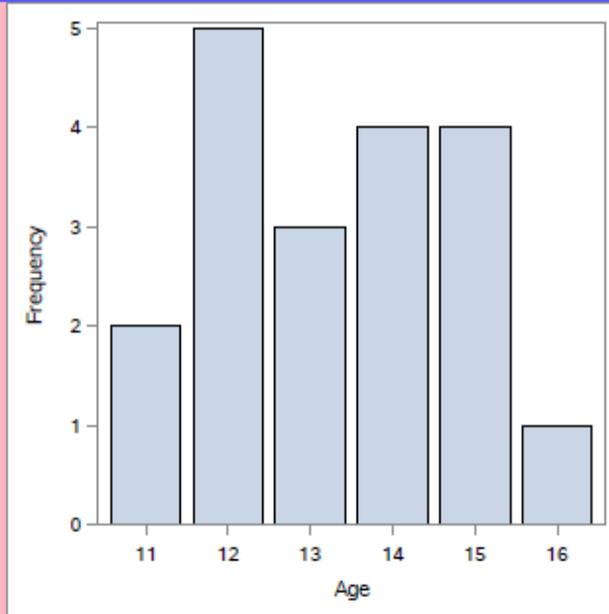


Figure 11. Two Plots with ODS Layout

This is exactly what Jack wanted. Each graph is sized correctly. The plot by age is 4x4 inches in the 1st region. In the 2nd region the plot by sex is 3x3 inches. Notice the 1st region (in pink) is 5 inches in height. Since we asked for the 2nd region to start at the 5-inch mark the 1st region fills up everything until that point. Since our layout container specifically asked for a height of 9 inches, the 2nd region will take up the remaining space on the page. Therefore, the height of the 2nd region is 4 inches. So Jack realizes that it's probably best to size your graph using graphical statements that set the height and width of your output. By doing that you know your graphs will be the exact size you want.

INSIDER TIP #10: TITLE AND FOOTNOTE PROCESSING

Starting in SAS 9.4M3 the ability to put titles and footnotes where you want them will be reality. Currently, when you use titles and footnotes with ODS Layout you only see them at the top and bottom of the page.

Here is an example of using titles and footnotes at SAS 9.4.

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
end;run;

options nodate;
title 'for the whole page';
footnote 'at bottom of the page';
ods listing close;
ods pdf file='tf1.pdf' notoc style=color;
ods layout gridded columns=2 column_widths=(4.5in 3.5in);

/*region 1*/
ods region style={background=lightpink};

proc means data=sashelp.shoes min mean max;
  var sales stores;
run;

/*region 2*/
title2 'title for the region w/report';
footnote2 'footnote for the region w/report';

ods region style={background=lightgreen};
proc print data=sashelp.class(obs=3);run;

proc report data=sashelp.class(obs=5);
title3 'proc report title';
footnote3 'proc report footnote';
run;

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

/*region 3*/
title2 'title for the region w/means';
footnote2 'footnote for the region w/means';
ods region style={background=lightpink};

proc means data=sashelp.shoes min mean max;
  var sales stores;
run;
```

```
ods layout end;  
ods _all_ close;
```

In this example, we wanted a system title and footnote at the top and bottom of the page, titles and footnotes within the 2nd and 3rd regions, and also a title and footnote around the PROC REPORT in the 2nd region. Looking at the output you can see that I did not get what I wanted.

for the whole page

The MEANS Procedure

Variable	Label	Minimum	Mean	Maximum
Sales	Total Sales	325.0000000	85700.17	1298717.00
Stores	Number of Stores	1.0000000	11.6481013	41.0000000

title for the region w/report

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

footnote for the region w/report

proc report title

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

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

proc report footnote

title for the region w/means

The MEANS Procedure

Variable	Label	Minimum	Mean	Maximum
Sales	Total Sales	325.0000000	85700.17	1298717.00
Stores	Number of Stores	1.0000000	11.6481013	41.0000000

footnote for the region w/means

footnote for the region w/means
at bottom of the page

at bottom of the page

Figure 12. Titles and Footnotes Using SAS 9.4

Duplicate footnotes appear at the bottom of the third region. The footnote for the PROC REPORT in the second region is not in the right place. The footnote for the 2nd region is also not in the right place. So what do you do if you want titles and footnotes around your regions but you do not have SAS 9.4M3 yet? The statement 'ODS TEXT=' to the rescue.

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options nodate;
title 'for the whole page';
footnote 'at bottom of the page';
ods listing close;
ods pdf file='tfl.pdf' notoc style=color;
ods layout gridded columns=2 column_widths=(4.5in
3.5in)style={background=yellow};

/*region 1*/
ods region style={background=lightpink};

proc means data=sashelp.shoes min mean max;
  var sales stores;
run;

/*region 2*/
ods region style={background=lightgreen};
ods text = 'title for the region w/report';
proc print data=sashelp.class(obs=3);run;

ods text = 'proc report title';
proc report data=sashelp.class(obs=5);
run;
ods text = 'proc report footnote';

proc print data=sashelp.class(obs=7);run;
ods text = 'footnote for the region w/report';

/*region 3*/
ods region style={background=orange};
ods text = 'title for the region w/means';

proc means data=sashelp.shoes min mean max;
  var sales stores;
run;
ods text = 'footnote for the region w/means';

ods layout end;
ods _all_ close;
```

By using ODS TEXT= you can put text anywhere in the output. You can get the look you want but you have to put the statements where you want the titles and footnotes to appear. There are two major differences that you need to know. The first is that the default style for ODS TEXT= looks differently than the title and footnote style. It's easy to change either by using inline styles or a PROC TEMPLATE to set

the look you want. The second difference is a little more troublesome. Titles and footnotes repeat on every page when you go to a new page. ODS TEXT= will not repeat on page breaks. The output from the above code is only one page, so the problem is not an issue with this example.

for the whole page

The MEANS Procedure

Variable	Label	Minimum	Mean	Maximum
Sales	Total Sales	325.0000000	85700.17	1298717.00
Stores	Number of Stores	1.0000000	11.6481013	41.0000000

title for the region w/report

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

proc report title

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

proc report footnote

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

footnote for the region w/report

title for the region w/means

The MEANS Procedure

Variable	Label	Minimum	Mean	Maximum
Sales	Total Sales	325.0000000	85700.17	1298717.00
Stores	Number of Stores	1.0000000	11.6481013	41.0000000

footnote for the region w/means

at bottom of the page

Figure 13. Titles and Footnotes with Text Statement Using SAS 9.4

Let's look at the first program again running at SAS 9.4M3 and see the difference.



Figure 14. Titles and Footnotes Using SAS 9.4M3

Notice how everything is in the right position. It's clear which titles and footnotes belong to each region. The title and footnote are obvious as well for the PROC REPORT and the overall page. This is exactly what we want with titles and footnotes.

One last thing we need to discuss about titles and footnotes is how they integrate with graph objects. Starting in SAS 9.3 ODS PDF enabled the ability to use the “nogititle” and “nogfootnote” syntax. As mentioned in a past SAS Global Forum paper (Huntley 2012) you can determine if you want the titles and footnotes inside or outside the graph object. The default behavior is to have the titles and footnotes inside the graph object.

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

options nodate nonumber;
ods listing close;
ods pdf file='tf_graph.pdf' notoc gtitle gfootnote style=color;

title 'system title';
footnote 'system footnote';

ods layout gridded columns=2 column_widths=(4.5in 3.5in)
style={background=yellow};
title 'this is region title';
footnote 'this is region footnote';

ods region style={background=lightpink};
proc print data=sashelp.class(obs=5);
title 'this is procedure title: print';
footnote 'this is procedure footnote: print';
run;

goptions hsize=4in vsize=5in;
proc gchart data=sashelp.class;
  hbar age;
title 'this is graph title';
footnote 'this is graph footnote';
run; quit;

ods region style={background=lightgreen};
proc print data=sashelp.class(obs=3);run;

ods layout end;
ods _all_ close;
```

I have “gtitle” and “gfootnote” in the ODS PDF statement. It is not necessary because this is the default behavior but I left it in to show how the toggle behavior works. Here is how the output looks running with SAS 9.4M3.

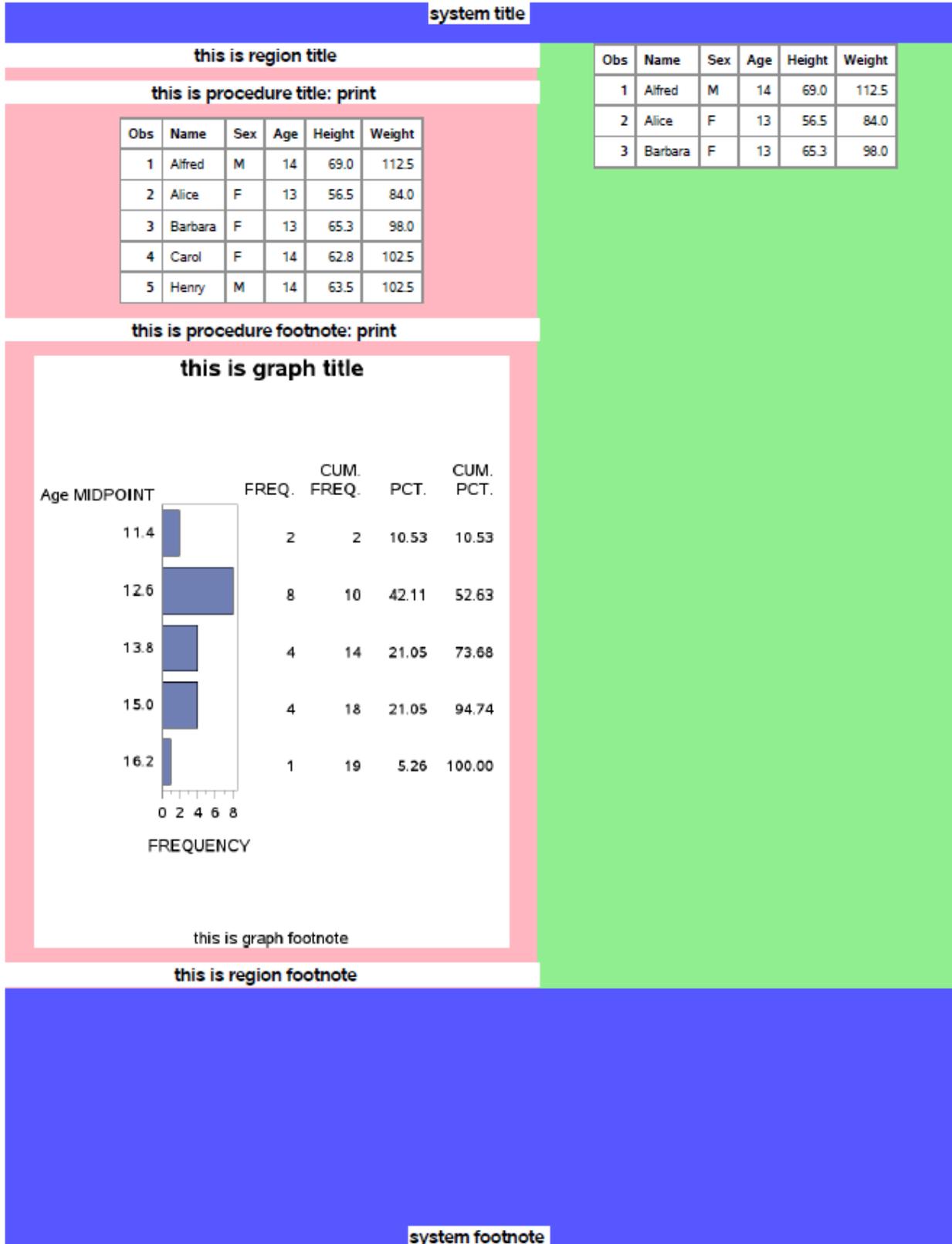


Figure 15. Titles and Footnotes Inside Graph Using SAS 9.4M3

You can see how the graph contains the title and footnote inside the graph because of its placement inside the white graph background. Now if I toggle the ODS PDF syntax like this ...

```
ods pdf file='tf_graph.pdf' notoc nogtitle nogfootnote style=color;
```

You see how the title and footnote look outside of the graph object.

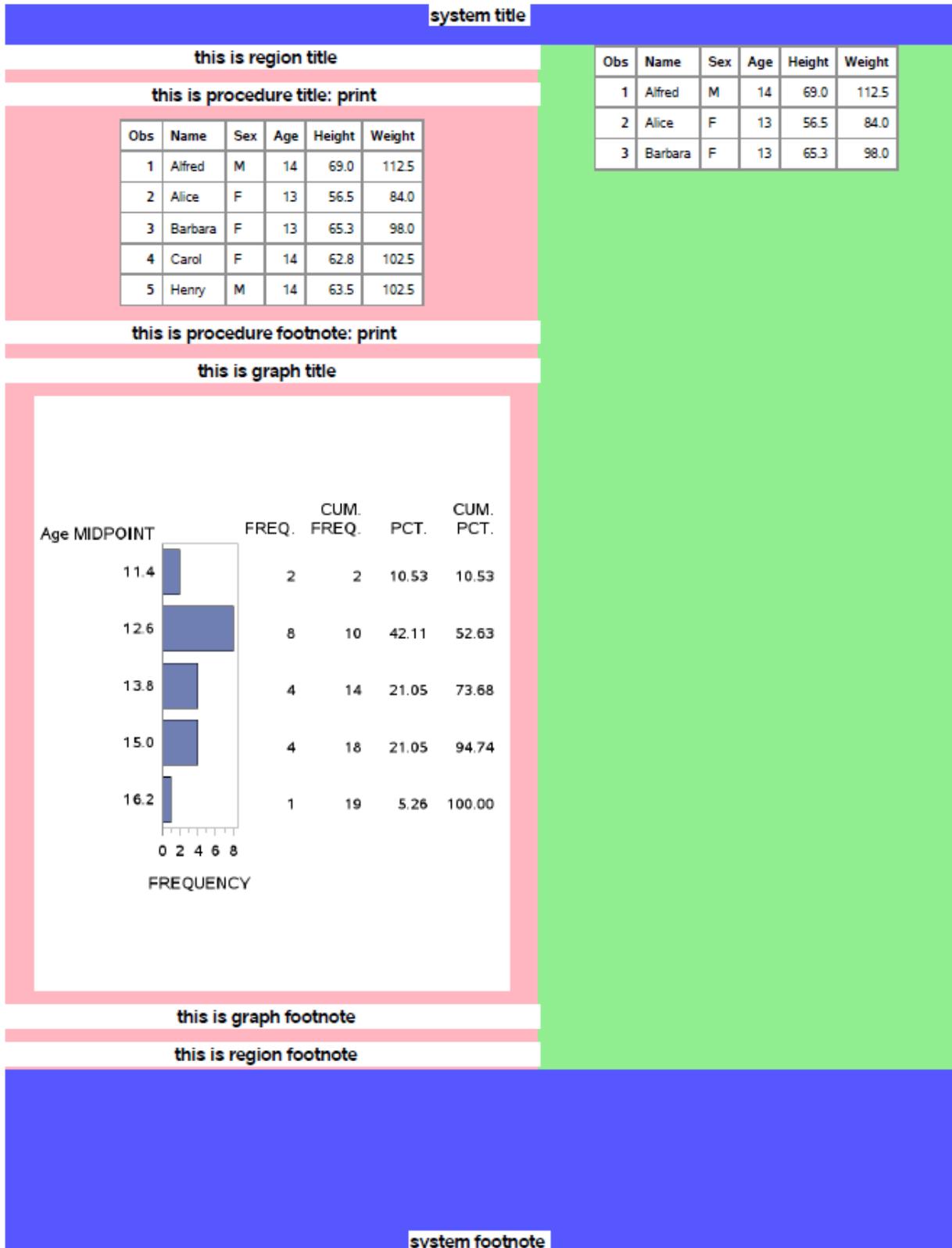


Figure 16. Titles and Footnotes Outside Graph Using SAS 9.4M3

To recap, the functionality of titles and footnotes inside a layout is available in SAS 9.4M3, which is not yet available. Until then you will have to use the ODS TEXT= syntax to put text where you want it located.

INSIDER TIP #11: SOMETIMES YOU DO NOT NEED ODS LAYOUT (AND SOMETIMES YOU DO)!

Louisa offers ... If I just want column output I might not have to use ODS Layout. Louisa has noticed that she just wants her output to have two evenly sized columns, then adding "columns = 2" on her ODS PDF file= statement is the easiest thing to do. By using the syntax "columns = x", ODS PDF will divide up the available space on a page by the number of columns specified. Using the value of "2" shows that ODS PDF will create two 5.25 inch columns using a landscape orientation. Look at Louisa's code snippet:

```
proc template;
  define style color;
    parent = Styles.pearl;
    style body from body /
      backgroundcolor=light blue;
  end;run;

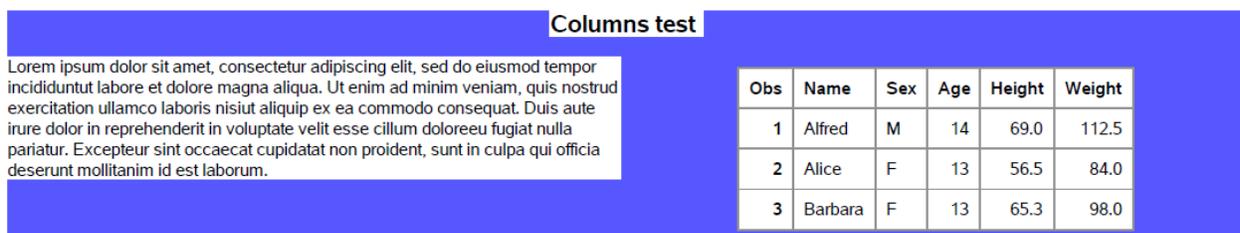
options nodate;
ods pdf file="columns_text.pdf" notoc columns=2 style=color;
title "Columns test";

ods text="Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id
est laborum.";
ods pdf startpage = now;

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

ods pdf close;
```

Louisa did need to set STARTPAGE to NOW to force the PROC PRINT table to go to the next column. When you use columns=2, ODS PDF internally views each column as a separate page of output. Without the STARTPAGE= setting the table would have gone right below the text.



The screenshot shows the output of a SAS ODS PDF statement with columns=2. The output is displayed in a landscape orientation and is split into two columns. The left column contains a block of Lorem Ipsum text. The right column contains a table with 6 columns: Obs, Name, Sex, Age, Height, and Weight. The table has 3 rows of data.

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

Figure 17. Columnar Output with columns=2

Louisa knows if she wants two columns with different widths, she will need to use ODS Layout:

```
options nodate nonumber orientation=landscape;
ods listing close;
```

```
ods escapechar="^";
ods pdf file="gutter_text.pdf" notoc style=color;
title "Gutter test";

ods layout start columns=2 column_gutter=1in;

ods region width=6in;
ods pdf text="Lorem ipsum dolor sit amet, mauris nam curabitur est odio
quis, velit wisi dapibus, id donec tristique risus eget mauris cras, ut
auctor luctus, nihil odio non. Nunc donec ut tristique nihil nulla,
volutpat dui vestibulum est etiam, eget proin sem vestibulum. Amet sodales
interdum blandit semper. Fusce amet velit, sed morbi imperdiet augue
semper donec. Nisl rutrum, vestibulum ac erat, tincidunt bibendum, erat
nam. Etiam integer donec condimentum neque ac, mauris luctus arcu. Non sed
vitae quam dictum nullam. In maecenas, semper in adipiscing tortor iaculis
etiam aut, sem et tellus molestie incididunt lorem, maecenas fusce felis,
primis mauris. Egestas nonummy, nulla urna, ut felis dui gravida. Sit sit
nunc est ultricies, quis sed turpis lacinia ipsum tenetur, praesent sapien,
mattis ac sed nec erat hendrerit ipsum, dignissim id aliquet tortor aenean
ipsum. Mattis per lectus volutpat integer eu, libero gravida et nec lectus,
ut mauris turpis. Consequat dis, in pellentesque.";

ods region width=3in;
proc print data=sashelp.class(obs=3);run;

ods layout end;
ods pdf close;
```

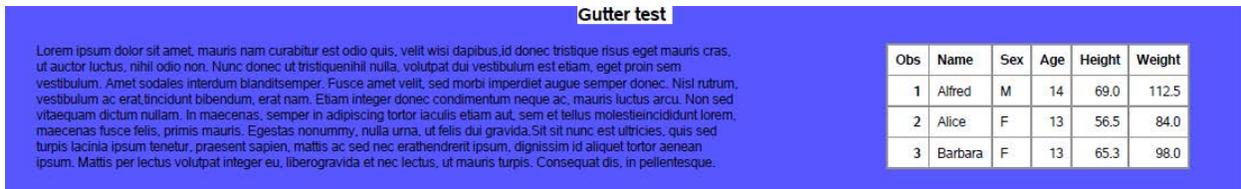


Figure 18. Columnar Output with ODS Layout

You can see Louisa has two columns. One is 6 inches wide while the other is 3 inches wide. Based on Mark's previous tip, she knows she needs to take care to ensure that her layout regions fit on the page. Notice the ODS LAYOUT START statement. It is coded to be two columns. Something else to note is that ODS Layout also has a default space incorporated between the columns. This is referred to as a column gutter and it is set to 30 pixels or about .20 of an inch by default. To change this, use the COLUMN_GUTTER option. Column gutters provide a way to maximize your space and provide good visual spacing between columns.

Comparing the two results you can see that with Layout the first column is wider and can fit more of the text in the space allowed. By using the ODS REGION statement Louisa has more control over how the output fits on the page.

You have to decide beforehand how you want your output to look. How wide do your columns need to be? What orientation should your page be? How much space do you want between columns of output? This will determine whether you use PDF columns= or ODS Layout. It's back to trial and error. You plan out your output. You code it and then test it. You might have to modify some numbers to get the perfect look.

CONCLUSION

So you have SAS 9.4 and you are about to take your first trip to ODS LAYOUT, or you are an intrepid SAS coder who was using ODS LAYOUT in a previous SAS release. With all the resources available to you (SAS online doc, ODS tip sheets, SAS Global Forum papers) this paper will help you narrow your focus and get you on the right path to great ODS Layout output.

REFERENCES

Huntley, Scott. 2012. "A Different Point of View with ODS PDF in SAS® 9.3." *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings12/260-2012.pdf>

Kummer, Daniel. 2014. "Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings14/SAS330-2014.pdf>

O'Conner, Dan. 2013. "Take Home the ODS Crown Jewels: Master the New Production Features of ODS LAYOUT and Report Writing Interface Techniques." Cary, NC: SAS Institute Inc. *Proceedings of the SAS Global Forum 2013 Conference*, Available at <http://support.sas.com/resources/papers/proceedings13/015-2013.pdf>

ACKNOWLEDGMENTS

The author would like to thank Bari Lawhorn, Allison Crutchfield, and Daniel Kummer 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.
100 SAS Campus Drive
Cary, NC 27513
Scott.Huntley@sas.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.