

MWSUG 2016 – Paper TT16-SAS

## Recapping Two Winning ODS Layout Talks for SAS® 9.4: ODS Destination for PowerPoint and the ODS PDF Destination

Bari Lawhorn, SAS Institute Inc., Cary, NC

### ABSTRACT

Get your seats now for an ODS layout doubleheader! Bari Lawhorn, SAS Senior Principal Technical Support Analyst, will present highlights from two very popular SAS Global Forum presentations about ODS layout: Jane Eslinger's 2016 paper, "The Dynamic Duo: ODS Layout and the ODS Destination for PowerPoint," and Scott Huntley's 2015 paper, "An Insider's Guide to ODS LAYOUT Using SAS® 9.4." For the lead-off, Bari discusses the winning combination of ODS layout and the ODS destination for PowerPoint, which produces native PowerPoint files from your output. Using ODS destination for PowerPoint together with ODS layout enables you to dynamically place your output on each slide. Through code examples, Bari shows you how to create a custom title slide as well as place the desired number of graphs and tables on each slide. Next up in Bari's talk is the topic of the ODS LAYOUT syntax that has been preproduction in the PDF destination for many SAS versions. She will cover insider tips for how to update your SAS® 9.3 code and how to get the best-looking results when you use ODS LAYOUT and the PDF destination. This presentation is sure to be a hit with ODS fans.

## The Dynamic Duo: ODS Layout and the ODS Destination for PowerPoint

Jane Eslinger, SAS Institute Inc.

### ABSTRACT

Like a good pitcher and catcher in baseball, ODS layout and the ODS destination for PowerPoint are a winning combination in SAS® 9.4. With this dynamic duo, you can go straight from performing data analysis to creating a quality presentation. The ODS destination for PowerPoint produces native PowerPoint files from your output. When you pair it with ODS layout, you are able to dynamically place your output on each slide.

Through code examples this paper shows you how to create a custom title slide, as well as place the desired number of graphs and tables on each slide. Don't be relegated to the sidelines—increase your winning percentage by learning how ODS layout works with the ODS destination for PowerPoint.

### INTRODUCTION

The ODS destination for PowerPoint was first introduced in SAS® 9.4. The destination enables you to create SAS® procedure output in the PresentationML format used by Microsoft PowerPoint 2010 and 2013. As with other ODS destinations, the PowerPoint destination uses table and style templates to render the tables and graphs that you produce using procedures. The syntax for using the ODS destination for PowerPoint is also the same as that for other destinations. You specify the destination and the presentation filename in the ODS statement and close the destination after all desired output has been created:

```
ods powerpoint file ="example.pptx";  
<procedure code>  
ods powerpoint close;
```

The ODS LAYOUT statements have been available since SAS® 9.1. However, they were not fully supported until SAS 9.4. SAS supports two types of layout, but only one is supported for the PowerPoint destination. The type of layout supported for PowerPoint is *gridded layout*. Gridded layout enables you to arrange output in a two-dimensional grid structure. The basic syntax is as follows:

```
ods layout gridded;  
ods region;  
proc print data=sashelp.baseball;  
run;  
ods layout end;
```

The size of the grid cell is dynamically computed to fit the output object. Within the grid structure, there are a number of different ways to arrange objects, such as placing tables next to graphs, text next to tables, or text above both a table and a graph. ODS LAYOUT statements give you the power and flexibility to arrange the output on each slide.

The LAYOUT= option in the ODS POWERPOINT statement applies a predefined layout. ODS LAYOUT statements perform a different function. They write multiple output objects to one slide in a layout that you specify. LAYOUT= is discussed in the last section of this paper.

Both methods can be used within one presentation, but they should not be used on the same slide. For all examples in this paper, the code uses ODS LAYOUT. It does not specify a predefined layout.

This paper is meant to teach the basic techniques of using ODS LAYOUT with the PowerPoint destination. As you read this paper and work through the examples, recognize that the ODS destination for PowerPoint and ODS for other destinations behave in similar though not identical ways. The PowerPoint destination is designed specifically for presentations. Therefore, it is meant for graphs, short

tables, and brief text. The amount of information that can fit on one 8.5x11 page in a PDF file does not necessarily fit on one slide in PowerPoint. The size of the text, tables, and graphs is very large compared to that for all other destinations. Presentations are, by their nature, meant to be viewed on a large screen by an audience.

## THE GAME PLAN—DECIDING WHAT YOU NEED TO DO

Before every game, a pitcher and catcher develop a game plan. They decide how they are going to approach each hitter. You should develop a game plan too. Before you begin programming, you should have a good idea of what you want the presentation to contain and how you want it to look. For example, are you going to have slides that contain only graphs, only tables, or only text, or some combination of all three? Most likely, you will need all three, which means you need to determine how many objects you want to appear on each slide.

As you begin planning, it is important to consider both how the creation of a new slide is triggered in the destination and the size of your output object. These topics are addressed next.

### TRIGGERING A NEW SLIDE

The default behavior of the PowerPoint destination is to output one object per slide, along with the title and footnote. The purpose of ODS LAYOUT is to output multiple objects in a grid structure on each slide. It therefore overrides the PowerPoint destination default behavior.

There are two ways to trigger a new slide when you are using gridded layout. The first method is to fill the entire slide. When the slide is full, PowerPoint is forced to place the next output object, or container, on the next slide.

The second method for triggering a new slide is to use a DATA \_NULL\_ step between layout containers. The DATA \_NULL\_ method should be used when the layout container does not fill a slide but you need the next layout container to start on the next slide. The code should be placed after the ODS LAYOUT END statement:

```
data _null_;  
  dcl odsout obj();  
run;
```

If you have tried using ODS LAYOUT with PowerPoint before, you might have seen that graphs that you expect to see on the second slide were placed at the bottom of the first slide, and the graph objects were very small. This behavior occurs when the layout container does not fill the first slide. The destination tries to fill the slide by adjusting the size of the output objects. The DATA \_NULL\_ step above should remedy this problem.

Also, be aware that the ODSLST, ODSTEXT, and ODSTABLE procedures all have an option in the procedure statement called PAGEBREAK. When PAGEBREAK=YES, the procedure generates a page break before the procedure is run. Note that the option does not help you generate a new slide after the procedure.

### SETTING OUTPUT OBJECT SIZE

In addition, variable labels, axis and marker labels, and BY line strings all affect the size of your output. Graphs and tables might be removed automatically from the presentation if they do not fit within the layout container due to the size of the output object. If the output does not fit in the size of the layout or region, the destination reverts to its default behavior of one output object per slide. Usually, a warning message is written to the log if either one of these issues occurs.

The easiest way to ensure that your graph or table fits into the grid cell is to let the PowerPoint destination size the output. The PowerPoint destination optimizes the size of the output to fit into the size of the space available. However, when placing multiple graphs on a slide, you should specify the size of a graph as you write the procedure code. For the ODS GRAPHICS procedures, the size is specified in the ODS GRAPHICS statement. For SAS/GRAPH® procedures, the size is specified in the GOPTIONS statement.

## UNDERSTANDING DEFAULT BEHAVIOR

When an ODS layout container is started, all of the output is saved until an ODS LAYOUT END statement is reached. After this statement is reached, the destination begins creating the output. You might encounter strange behavior in titles and footnotes because of this accumulation of output. Titles that belong to the last object in the layout might end up with the first object because everything is exported together. Be sure to use null TITLE and FOOTNOTE statements where needed to avoid this issue. Also, the NOGTITLE and NOGFOOTNOTE options should be placed in the ODS POWERPOINT statement as needed to prevent the titles and footnotes from being part of the graph. Remember, anything that is part of the graph contributes to the amount of space required by the object and might prevent the graph from fitting in the grid cell.

Also, settings from the OPTIONS() option in the ODS POWERPOINT statement can be changed only between slides. If a layout container starts at the bottom of slide 1 but continues to slide 2, those option values remain in effect for the second slide. If your second slide contains a specific background image or color that you do not expect, this is a good indication that the first slide was not full and the layout for the second slide actually started on the first slide.

The ODS destination for PowerPoint is a new technology, so you might encounter a few unexpected quirks or bugs. As with most software, it is best to use the latest version of SAS available.

## THE FIRST INNING—CREATING THE TITLE SLIDE

The first inning of a baseball game is very important. It's when a pitcher and catcher start putting their game plan into action and set the tone for the rest of the game. The title slide of a presentation can play the same role. It establishes the theme and look of the presentation and lets the audience know what to expect.

By design, title slides do not have much information on them. Normally, they contain the presentation title, the names of the authors, and perhaps the date and the meeting title. Therefore, a title slide is not likely to require as much work as other slides to put that information exactly where you want it.

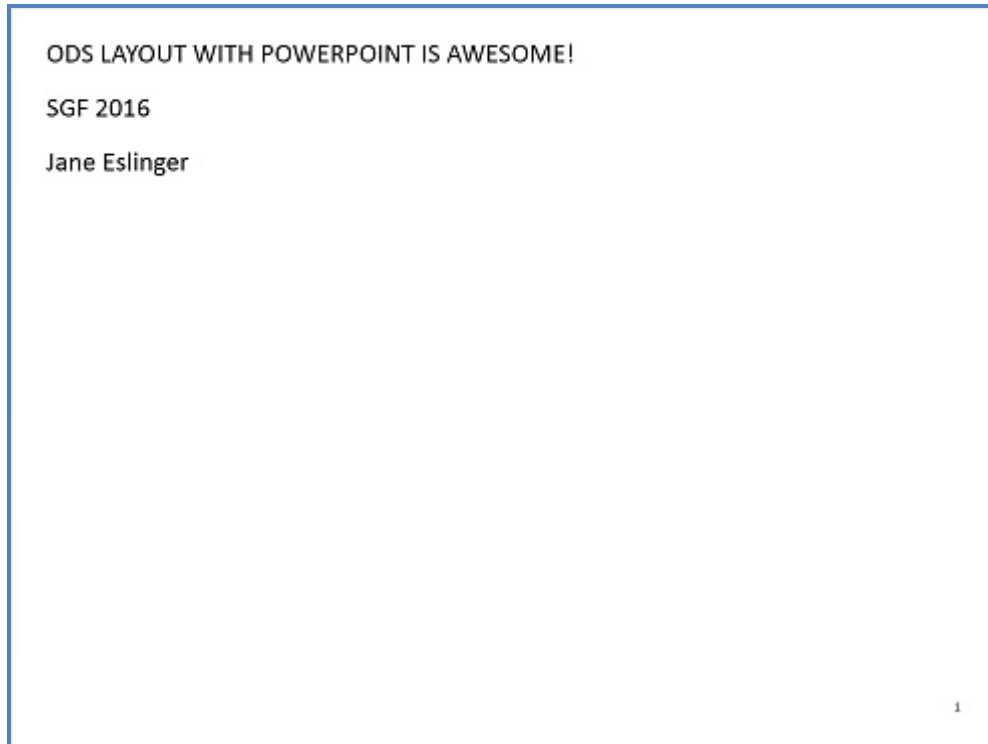
The first step in determining whether you need to adjust the placement of the text on a title slide is to look at what is created by default.

## GENERATING A TITLE SLIDE WITH PROC ODSTEXT

PROC ODSTEXT is very useful for creating a title slide because it outputs text blocks. You can place as many pieces of text within the block as necessary using the P statement, as shown here:

```
title;
footnote;
ods powerpoint file='title1.pptx';
proc odstext;
  p "ODS LAYOUT WITH POWERPOINT IS AWESOME!";
  p "SGF 2016";
  p "Jane Eslinger";
run;
ods powerpoint close;
```

The text from PROC ODSTEXT is placed in the top left corner of the slide by default, as shown below in Display 1.



**Display 1. Slide Created with the P Statement in PROC ODSTEXT**

A slide created using only PROC ODSTEXT is not very appealing. The style templates supplied by SAS for the ODS destination for PowerPoint, Styles.PowerPointLight and Styles.PowerPointDark, contain special style elements for styling titles and subtitles. The style elements change the font size and color and center the text, though it is still placed at the top of the slide. Here is an example. Display 2 shows the improved look.

```
ods powerpoint file='test2.pptx';  
proc odstext;  
  p "ODS LAYOUT WITH POWERPOINT IS AWESOME!" / style=PresentationTitle;  
  p "SGF 2016" / style=PresentationTitle2;  
  p "Jane Eslinger" / style=PresentationTitle2;  
run;  
ods powerpoint close;
```

# ODS LAYOUT WITH POWERPOINT IS AWESOME!

SGF 2016

Jane Eslinger

1

## Display 2. PresentationTitle Style Element Applied to P Statement in PROC ODSTEXT

Display 2 looks better than Display 1, but it might not fulfill your needs. Maybe you need the text moved over and down because of your company logo or a conference logo. Maybe you just want it in a different location. This is a perfect situation in which to use ODS LAYOUT.

## CHANGING TITLE PLACEMENT WITH ODS LAYOUT

ODS LAYOUT statements and their options allow you to move a region container, which holds the text, to another location other than the default. The X= option in the ODS LAYOUT GRIDDED statement specifies the horizontal starting position of the layout. The Y= option controls the vertical starting position of the layout. The default location is  $x=0$   $y=0$ , which is the top left corner of the layout container.

For the title slide, you most likely need only one container that is one row and one column wide, which is the default grid that is generated simply by submitting an ODS LAYOUT GRIDDED statement. The container is filled with the text from the PROC ODSTEXT step.

The X= and Y= options are used in the following example to move the text to the right and down, as illustrated in Display 3. This placement is vital to preventing the text from being written over the top of the background image.

```
ods powerpoint file='title3.pptx'  
  options(backgroundimage="sgf16_background.jpg");  
ods layout gridded x=25pct y=35pct;  
ods region;  
proc odstext;  
  p "ODS LAYOUT WITH POWERPOINT IS AWESOME!" / style=PresentationTitle;  
  p "SGF 2016" / style=PresentationTitle2;  
  p "Jane Eslinger" / style=PresentationTitle2;  
run;  
ods layout end;  
ods powerpoint close;
```



### Display 3. Layout Container Moves Text Down and to the Right Using the X= and Y= Options

The above code used a percentage to place the layout in the desired position. There are a number of different units of measure that could have been used. You must use one of the units of measure when specifying the x and y position. If you do not, the gridded layout is centered by default.

Notice that the ODS LAYOUT GRIDDED statement does not have any additional arguments—because it does not need any. Only the X= and Y= options are necessary to adjust the placement of the layout container, which is one row long and one column wide. The ODS REGION statement does not require options for this example either.

## THE SECOND INNING—OUTPUTTING GRAPHS

The game plan has been established and the tone has been set with the title slide. It is now time to get into the heart of the presentation to show the world the results of your hard work and analysis. You might want to start your presentation by displaying multiple graphs on one slide. The syntax for placing two graphs on one slide has six key steps, as highlighted below:

```

title "Two Graphs Per Slide";
ods powerpoint file='twographs1.pptx' nogtitle nogfootnote; ❶
ods graphics / width=4.5in height=4in; ❷
ods layout gridded columns=2; ❸
ods region; ❹
proc sgplot data=sashelp.baseball;
  where league = "National";
  vbar team / response=nruns group=division;
run;

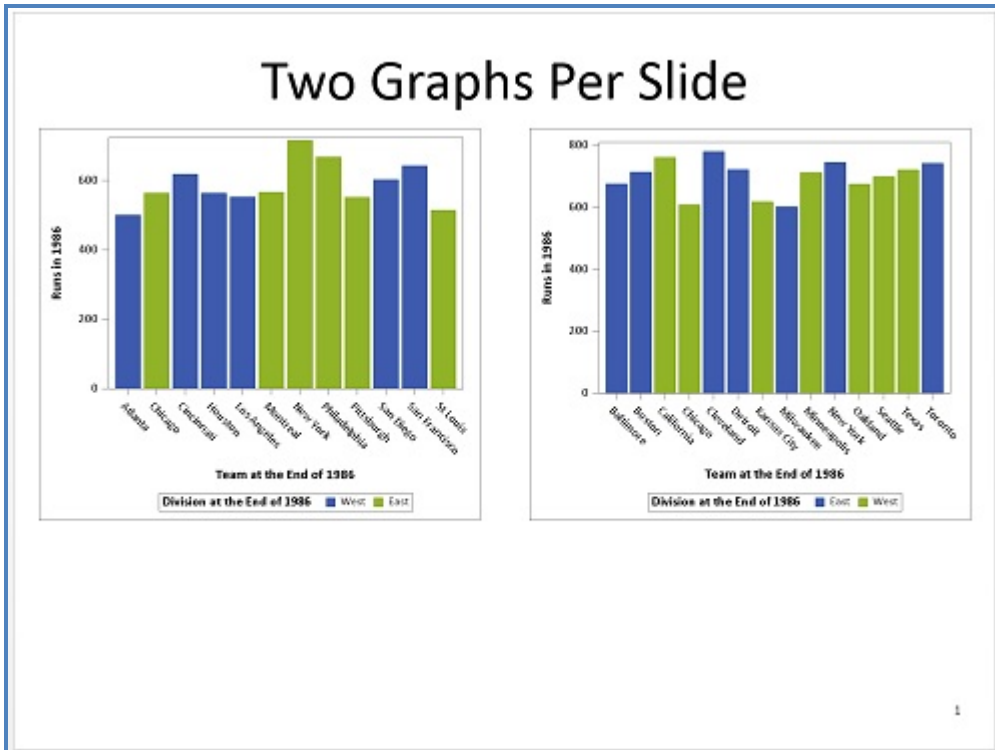
ods region; ❺
proc sgplot data=sashelp.baseball;
  where league = "American";
  vbar team / response=nruns group=division;

```

```
run;
ods layout end; ⑥
ods powerpoint close;
```

- ① The NOGTITLE and NOGFOOTNOTE options are added to the ODS POWERPOINT statement to force any titles and footnotes to be part of the slide instead of part of the graph.
- ② The WIDTH= and HEIGHT= options set the size of both graphs so that they fit within the region that is half the width of the slide. Without these options, PowerPoint reverts to default behavior and puts one graph per slide.
- ③ The ODS LAYOUT GRIDDED statement starts the layout container. The COLUMNS= option is set to 2 so that the output is placed side-by-side.
- ④ The ODS REGION statement begins the region container that is to hold the first graph.
- ⑤ The ODS REGION statement begins the region container that is to hold the second graph.
- ⑥ The ODS LAYOUT END statement ends the gridded layout.

The resulting slide is shown below, in Display 4.



Display 4. Two Graphs on One Slide

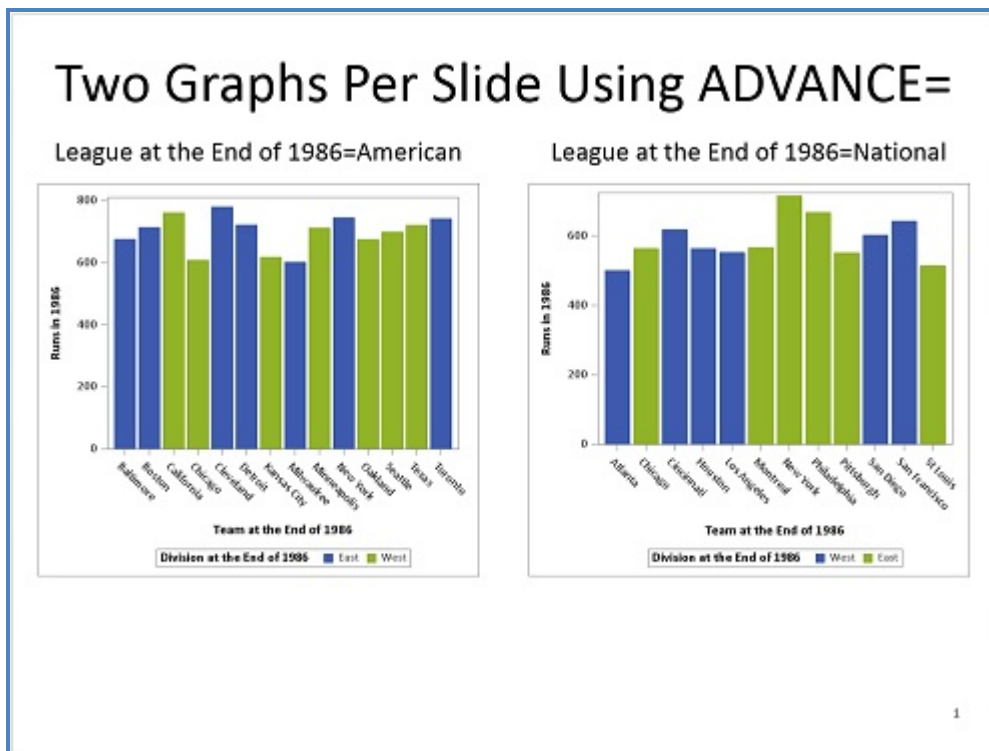
The code shown above used two procedure steps, where each procedure filled one region. It is possible to fill multiple regions using only a single procedure. This is useful when one procedure generates multiple output objects. The ADVANCE= option in the ODS LAYOUT GRIDDED statement specifies how the grid should be populated. It can be explicitly populated or dynamically populated by BY groups, tables, or procedures.



The preceding code can be simplified to use just one SGPLOT procedure with the addition of a BY statement. The ADVANCE= option can be used to put the output in the appropriate regions, as shown here and in Display 5:

```
proc sort data=sashelp.baseball out=baseball;
  by league;
run;

title "Two Graphs Per Slide Using ADVANCE=";
ods powerpoint file='twographs2.pptx' nogtitle nogfootnote;
ods graphics / width=4.5in height=4in;
ods layout gridded columns=2 advance=bygroup;
ods region;
proc sgplot data=baseball;
  by league;
  vbar team / response=nruns group=division;
run;
ods layout end;
ods powerpoint close;
```



Display 5. Two Graphs on One Slide Using ADVANCE=

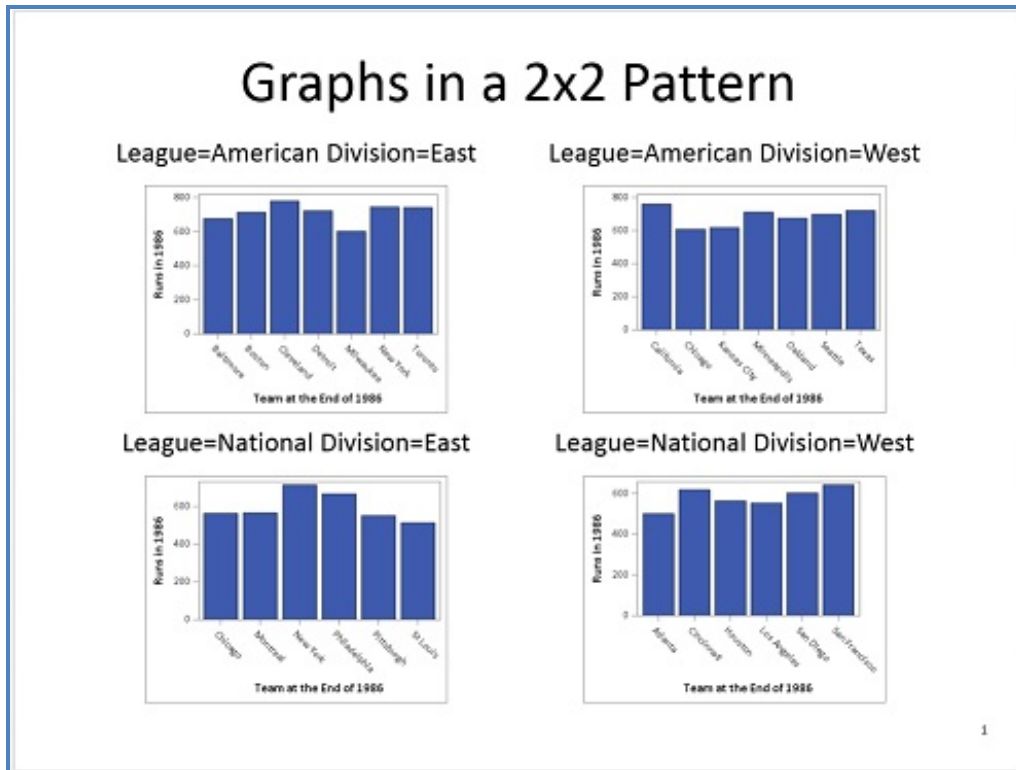
The number of graphs per slide can be increased to four and displayed in a 2x2 pattern. A slide might also hold more than four graphs, but when you start increasing the number of output objects, their size becomes critically important. And remember that titles, BY lines, and axis labels use valuable space.

As in the previous example, this next example uses COLUMNS=2 and ADVANCE=BYGROUP in the ODS LAYOUT GRIDDED statement. The ROWS= option is added and set to 2 so that each slide has two columns and two rows creating a 2x2 pattern. The WIDTH= and HEIGHT= values in the ODS GRAPHICS statement have changed because the graphs need to be much smaller.

The SASHELP.BASEBALL data set contains data for both leagues and both divisions within the league, which is perfect for generating four graphs, as shown in Display 6. Both the LEAGUE and DIVISION variables are in the BY statement.

```
proc sort data=sashelp.baseball out=baseball;
  by league division;
run;

ods powerpoint file='twobytwol.pptx' nogtitle nogfootnote;
title "Graphs in a 2x2 Pattern";
ods graphics on / height=2.25in width=3in;
ods layout gridded columns=2 rows=2 advance=bygroup;
ods region;
proc sgplot data=baseball;
  by league division;
  vbar team / response=nruns;
  label league='League' division='Division';
run;
ods layout end;
ods powerpoint close;
```



**Display 6. Four Graphs on One Slide Using COLUMNS=2 and ROWS=2**

Notice that a LABEL statement is included in the code above. The labels for these variables are both very long. The original full text of the BY line was "League at the End of 1986=American Division at the End of 1986=East." Using this text would have generated the following warning in the log, and only the first row of graphs would have fit on the slide. The labels were shortened to take up less room so that all four graphs fit on the slide together.

```
WARNING: GRIDDED Layout exceeds the available space for POWERPOINT
destination and will be paneled.
```

Note that the example above had **ROWS=2** and **COLUMNS=2**. Specifying both **ROWS=** and **COLUMNS=** values in the ODS LAYOUT GRIDDED statement hardcodes the number of cells in the layout. This technique is acceptable when you know you are outputting only four graphs. However, if the data changes or you know that you need to output more than four graphs, hardcoding the number of cells is likely to cause problems. The following warnings are written to the log if the procedure outputs more graphs than the number of available cells. Also, the first slide will contain four graphs, but all other slides will contain just one graph.

```
WARNING: LAYOUT is full. Implicitly closing LAYOUT.
WARNING: LAYOUT is full. Implicitly closing LAYOUT.
NOTE: There were 322 observations read from the data set WORK.BASEBALL.

103 ods layout end;
WARNING: Missing LAYOUT_START statement. This statement will be ignored.
WARNING: Missing LAYOUT_START statement. This statement will be ignored.
```

The output in Display 6, above, can be generated without hardcoding the number of cells in the layout. The following code demonstrates specifying the region height to place four graphs per slide:

```
proc sort data=sashelp.baseball out=baseball;
  by league division;
run;

ods powerpoint file='twobytwo2.pptx' nogtitle nogfootnote;
title "Graphs in a 2x2 Pattern";
ods graphics on / height=2.25in width=3in; ❶
ods layout gridded columns=2 advance=bygroup; ❷
  ods region height=47%; ❸
  proc sgplot data=baseball;
    by league division;
    vbar team / response=nruns;
    label league='League' division='Division'; ❹
  run;
ods layout end;
ods powerpoint close;
```

- ❶ The height and width of the graph are specified to ensure that the graph fits within the region cell.
- ❷ Only one dimension of the gridded layout is specified, **COLUMNS=2**.
- ❸ The **HEIGHT=** option in the ODS REGION statement specifies that the region should use approximately half of the height of the slide.
- ❹ The labels for the **LEAGUE** and **DIVISION** are shortened, so that the graphs fit in the region.

You might wonder why the height of the regions was specified in the ODS REGION statement instead of in the ODS LAYOUT GRIDDED statement using the **ROW\_HEIGHTS=** option. If **ROW\_HEIGHTS=** is used and set to **47%**, only two graphs are output and a warning is generated for each output object that does not appear in the presentation.

WARNING: The region or layout was too small to accommodate the output supplied for the POWERPOINT destination. Output will be suppressed.

The ROW\_HEIGHTS= option is for the height of each row specified. The ODS LAYOUT GRIDDED statement does not contain the ROWS= option, which means it defaults to a value of `one (1)`. All four graphs cannot fit into a row that is 47% of the available space, so PowerPoint suppresses two of the graphs and issues the warning.

The SAS® 9.4 Output Delivery System documentation describes the calculation of row heights this way:

*If omitted, the ROWS= option defaults to the maximum number of rows needed to populate the regions created in the vertical direction. If there are two columns, then the number of rows is half of the number of regions.*

When ROWS= is not used, it is better to specify the HEIGHT= option in the ODS REGION statement, which sets the height of each region.

## THE MIDDLE INNINGS—SHOWING OFF THE NUMBERS

The pitcher and catcher established a game plan and set it into motion with the title slide and the most important graphs. The middle part of the game or presentation is for backing up your claims. You have to show the most important data values, just like a pitcher proving his stuff can last the entire game.

Data values are often output in tabular form. In general, tables that are placed in a presentation should be wider rather than longer. A long table might span multiple slides, making it difficult for the audience to read. Table output objects present size and spacing challenges just like graphs. You have to pay attention to the number of columns, number of rows, labels, and width of the data values themselves. Often, instead of outputting the data in one large table, like you would in the PDF or HTML destinations, it is best to output the data in multiple smaller tables.

The SASHELP.BASEBALL data set contains two variables, NHITS and NRUNS, which are of great interest and need to be included in the presentation. The count, mean, minimum, and standard deviation are key statistics to display in the table. The goal of this example is to generate the N, MEAN, MIN, and STD statistics for the NHITS and NRUNS variables within League and Division and to have it all fit on one slide.

### PLACE TWO TABLES ON ONE SLIDE

Several different procedures could be used to generate the needed statistics for these two variables. For this paper, the REPORT procedure was chosen over the MEANS or TABULATE procedures because it allows more control over the table structure. This flexibility makes it easier to fit the tables inside the layout container. You must choose which procedure works best for your presentation.

The next example takes the preemptive step of generating a separate table for each variable in an attempt to prevent the table output object from spanning multiple slides. As you will see, it often requires multiple attempts to get the desired output. As shown below, the ODS LAYOUT GRIDDED statement uses `ROWS=2` so that the tables are stacked on top of each other. The results of this code are in Display 7.

```
ods powerpoint file='tables1.pptx' nogtitle nogfootnote;
title 'Data By League and Division';
ods layout gridded rows=2;
ods region;
proc report data=sashelp.baseball;
  column league division nruns,(n mean min std);
  define league / group 'League';
  define division / group 'Division';
run;
```

```

ods region;
proc report data=sashelp.baseball;
  column league division nhits,(n mean min std);
  define league / group 'League';
  define division / group 'Division';
run;
ods layout end;
ods powerpoint close;

```

### Data By League and Division

|          |          | Runs in 1986 |           |     |           |
|----------|----------|--------------|-----------|-----|-----------|
| League   | Division | n            | mean      | min | std       |
| American | East     | 85           | 58.517647 | 15  | 28.099149 |
|          | West     | 90           | 53.188889 | 13  | 23.929623 |
| National | East     | 72           | 49.666667 | 12  | 23.265159 |
|          | West     | 75           | 46.36     | 14  | 23.05378  |

|          |          | Hits in 1986 |           |     |           |
|----------|----------|--------------|-----------|-----|-----------|
| League   | Division | n            | mean      | min | std       |
| American | East     | 85           | 113.32941 | 36  | 48.772403 |
|          | West     | 90           | 102.35556 | 39  | 41.168689 |

---

### Data By League and Division

|          |          | Hits in 1986 |           |     |           |
|----------|----------|--------------|-----------|-----|-----------|
| League   | Division | n            | mean      | min | std       |
| National | East     | 72           | 99.805556 | 31  | 43.805472 |
|          | West     | 75           | 96.84     | 32  | 41.427253 |

**Display 7. Two Tables in a Two-Row Gridded Layout**

Display 7, above, shows that the results of the two PROC REPORT steps and the `ROWS=2` option span multiple slides, even after you put each analysis variable in its own table and carefully choose the procedure. The code placed no height or width restrictions on the regions. Therefore, the destination determined the size of the output objects, and a portion of the second table is placed on a second slide.

You might consider using the `ROW_HEIGHTS=` option in the `ODS LAYOUT GRIDDED` statement. This will result in the output being limited to one slide. However, the two data rows on the second slide in Display 7 will be suppressed, which is not a desired result.

Both tables contain six rows, which does not seem tall, but remember, the PowerPoint destination uses a large font and contains a title. The slide is just not big enough to hold a title and 12 rows of data. In circumstances like this, you have to rethink how to output the data because both tables have to fit on one slide. Otherwise, it throws off the rest of the presentation. Sometimes the answer to getting output objects to fit inside a gridded layout is to change the output objects and not the grid.

The tables from Display 7 can be modified to remove one header row each. The labels for the variables can be placed inside an `ODS TEXT=` statement, which can also serve as a title. The changes are demonstrated in the code below and depicted in Display 8.

```
ods powerpoint file='tables2.pptx' nogtitle nogfootnote;
title; ❶
ods layout gridded rows=2; ❷
ods region;
ods text ='Runs Data By League and Division'; ❸
proc report data=sashelp.baseball;
  column league division nruns,(n mean min std);
  define league / group 'League';
  define division / group 'Division';
  define nruns / ''; ❹
run;

ods region;
ods text='Hits Data By League and Division'; ❺
proc report data=sashelp.baseball;
  column league division nhits,(n mean min std);
  define league / group 'League';
  define division / group 'Division';
  define nhits / ''; ❻
run;
ods layout end;
ods powerpoint close;
```

- ❶ A null TITLE statement prevents a title from being output on the slide.
- ❷ The layout container is defined with two rows.
- ❸ An ODS TEXT= statement is used to output informative text about the table below it. It takes the place of the TITLE statement and provides more room for the table to fit on one slide.
- ❹ The label for the NRUNS variable is set to blank, which removes one of the header rows of the table.
- ❺ A ODS TEXT= statement is used to output informative text about the table below it. The text takes the place of the TITLE statement and provides more room for the table to fit on one slide.
- ❻ The label for the NHITS variable is set to blank, which removes one of the header rows of the table.

| Runs Data By League and Division |          |    |           |     |           |
|----------------------------------|----------|----|-----------|-----|-----------|
| League                           | Division | n  | mean      | min | std       |
| American                         | East     | 85 | 58.517647 | 15  | 28.099149 |
|                                  | West     | 90 | 53.188889 | 13  | 23.929623 |
| National                         | East     | 72 | 49.666667 | 12  | 23.265159 |
|                                  | West     | 75 | 46.36     | 14  | 23.05378  |

| Hits Data By League and Division |          |    |           |     |           |
|----------------------------------|----------|----|-----------|-----|-----------|
| League                           | Division | n  | mean      | min | std       |
| American                         | East     | 85 | 113.32941 | 36  | 48.772403 |
|                                  | West     | 90 | 102.35556 | 39  | 41.168689 |
| National                         | East     | 72 | 99.805556 | 31  | 43.805472 |
|                                  | West     | 75 | 96.84     | 32  | 41.427253 |

**Display 8. Two Tables and ODS TEXT= Placed in a Two-Row Gridded Layout**

Display 8 demonstrates that with a few small changes, all of the statistical values can be shown on one slide for both of the analysis variables. Also, though a title and a header were removed, no information was lost.

### USE MULTIPLE CONTAINERS ON ONE SLIDE

This paper has demonstrated using a layout container to move title text. It has also demonstrated putting multiple graphs and tables on one slide using either the ROWS= or COLUMNS= option in the ODS LAYOUT GRIDDED statement. This final example of ODS LAYOUT demonstrates using multiple layout containers to generate a single slide with multiple output objects.

One reason to use multiple containers for one slide is to have one output object spanning the width of the slide and placed above or below other output objects. The option ROW\_SPAN does not always work as expected in the ODS destination for PowerPoint. Instead of trying to force a spanning row, it is better to create a separate layout container for that output object.

The code in this example combines techniques already seen in previous examples. The text from PROC ODSTEXT is output into a container that is one row and one column wide. A table and a graph are output into a container that is two columns wide. The output from PROC ODSTEXT does not fill the slide, so the output from the next container automatically starts beneath it on the same slide. The size of the table and graph are adjusted to fit in the remaining space on the slide.

```

title 'Multiple Layout Containers One Slide';
ods powerpoint file='tables3.pptx' nogtitle nogfootnote;
ods layout gridded rows=1 columns=1; ❶
ods region;
proc odstext;
  p 'Table Shows Total Runs and Hits for Each League';
  p 'The Graph Contains One Bubble for Each Player. The Size of Each
    Bubble Represents the Magnitude of the RBIs.';
run;
ods layout end;

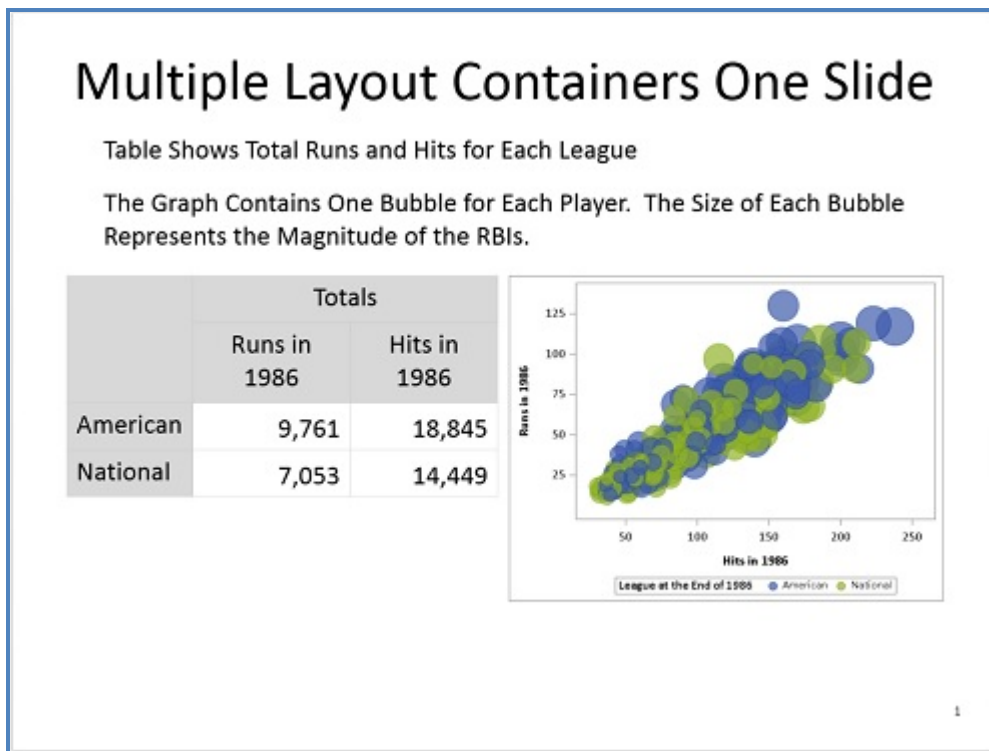
```

```
ods layout gridded columns=2 column_widths=(47% 47%) column_gutter=1pct; ❷
ods region;
proc tabulate data=sashelp.baseball;
  class league;
  var nruns nhits nrbi;
  tables league='', sum='Totals'*(nruns nhits)*f=comma12.;
run;

ods region;
proc sgplot data=sashelp.baseball;
  bubble x=nhits y=nruns size=nrbi/ group=league transparency=.3;
run;
ods layout end;
ods powerpoint close;
```

- ❶ This ODS LAYOUT GRIDDED statement creates the first layout container, which is one row wide and one column tall. The output from PROC ODSTEXT fills the container.
- ❷ This ODS LAYOUT GRIDDED statement creates a second layout container that is two columns wide. Set the column width. The COLUMN\_GUTTER option is used to decrease the space between the two output objects within this container.

Display 9 illustrates the use of multiple layout containers on a single slide.



### Display 9. Three Output Objects in Two Layout Containers

The ODSTEXT, TABULATE, and SGPLOT procedures were used to generate the output in Display 9, above. However, any three procedures could have been used in the layout. The first layout container could be filled with a table while the second layout container is filled with graphs. The important takeaway message from this example is that multiple layout containers can be placed on one slide and are sometimes necessary to get the desired output.



## THE FINAL STRETCH—USING THE LAYOUT= OPTION

As mentioned in the introduction of this paper, the ODS destination for PowerPoint has predefined layouts that are set using the LAYOUT= option in an ODS POWERPOINT statement. This paper focuses on using ODS LAYOUT within the PowerPoint destination, but it is important to be aware of the predefined layouts because they might provide what you need without having to use ODS LAYOUT. Again, both ODS LAYOUT and LAYOUT= can be used within the same presentation, but using them on the same slide is not recommended.

Three possible values for the LAYOUT= option are **TitleSlide**, **TitleAndContent**, and **TwoContent**.

The TitleSlide layout creates only title slides. The titles must be created with the P statement in PROC ODSTEXT and the number of titles is limited to two; only one footnote is allowed. This is the most straightforward way to generate a title slide and is a good first choice unless you need to change the placement of the text as was demonstrated earlier in the paper.

The TitleAndContent layout allows two titles and two footnotes. The number of output objects that fit on the slide is determined by the size of the output objects themselves. For example, two tables with three to five rows each can fit on one slide. The output objects are centered on the slide below the titles. This predefined layout stays in effect until you change it, using an ODS POWERPOINT statement with the LAYOUT= option.

The TwoContent layout places two output objects on the same slide in a two-column structure. This layout allows two titles and two footnotes. The output in a column continues until it is finished, which means it might continue across multiple slides. This predefined layout stays in effect until you change it.

The LAYOUT= option accepts a fourth value, which is **\_NULL\_**. This value has multiple purposes. The primary purpose is to return to the default layout state. This value should be used if you need to change from a predefined layout to the default behavior or need to create slides that use ODS LAYOUT. The second purpose of setting LAYOUT=\_NULL\_ is to trigger a new slide, which it does when placed between two sections using predefined layouts. For example:

```
ods powerpoint layout=twocontent;  
proc...  
proc...  
ods powerpoint layout=_null_;  
ods powerpoint layout=titleandcontent;  
proc...
```

Predefined layouts provide an easy way to generate certain types of output structure. You should consider using them along with ODS LAYOUT when doing so would be beneficial. One advantage of predefined layouts is that they are less likely to generate warnings and suppress output from the final presentation.

## CONCLUSION

The ODS destination for PowerPoint is a wonderful tool for outputting the results of your analysis into a form that you can share with everyone. When you use the capability of ODS LAYOUT within PowerPoint, your presentation is even more impactful because you can strategically place multiple results on the same slide.

This paper demonstrated how you can use ODS LAYOUT to change the position of pieces of text, like the presentation title. ODS LAYOUT is also used to put multiple graphs on one slide or to stack tables on top of each other. Like a good pitcher and catcher, the combination of ODS LAYOUT and PowerPoint are a dynamic duo that give you the advantage that you need to win.

## REFERENCES

Hunter, Tim. 2013. "A First Look at the ODS Destination for PowerPoint." *Proceedings of the SAS Global 2013 Conference*, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings13/041-2013.pdf>.

Huntley, Scott. 2015. "An Insider's Guide to ODS LAYOUT Using SAS® 9.4." *Proceedings of the SAS Global 2015 Conference*, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings15/SAS1836-2015.pdf>.

Hunter, Tim. 2016. "A Second Look at the ODS Destination for PowerPoint." *Proceedings of the SAS Global 2016 Conference*, Cary, NC: SAS Institute Inc.

## ACKNOWLEDGMENTS

The author is immensely grateful to Chevell Parker and Tim Hunter, both of whom contributed to this paper, and to Vivian McGee, who edited the paper.

## RECOMMENDED READING

- SAS® 9.4 Output Delivery System: User's Guide
- SAS® 9.4 Output Delivery System: Advanced Topics

## CONTACT INFORMATION

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

Jane Eslinger  
SAS Institute Inc.  
Cary, NC  
Email: [support@sas.com](mailto:support@sas.com)  
Web: [support.sas.com](http://support.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.

## 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;
```

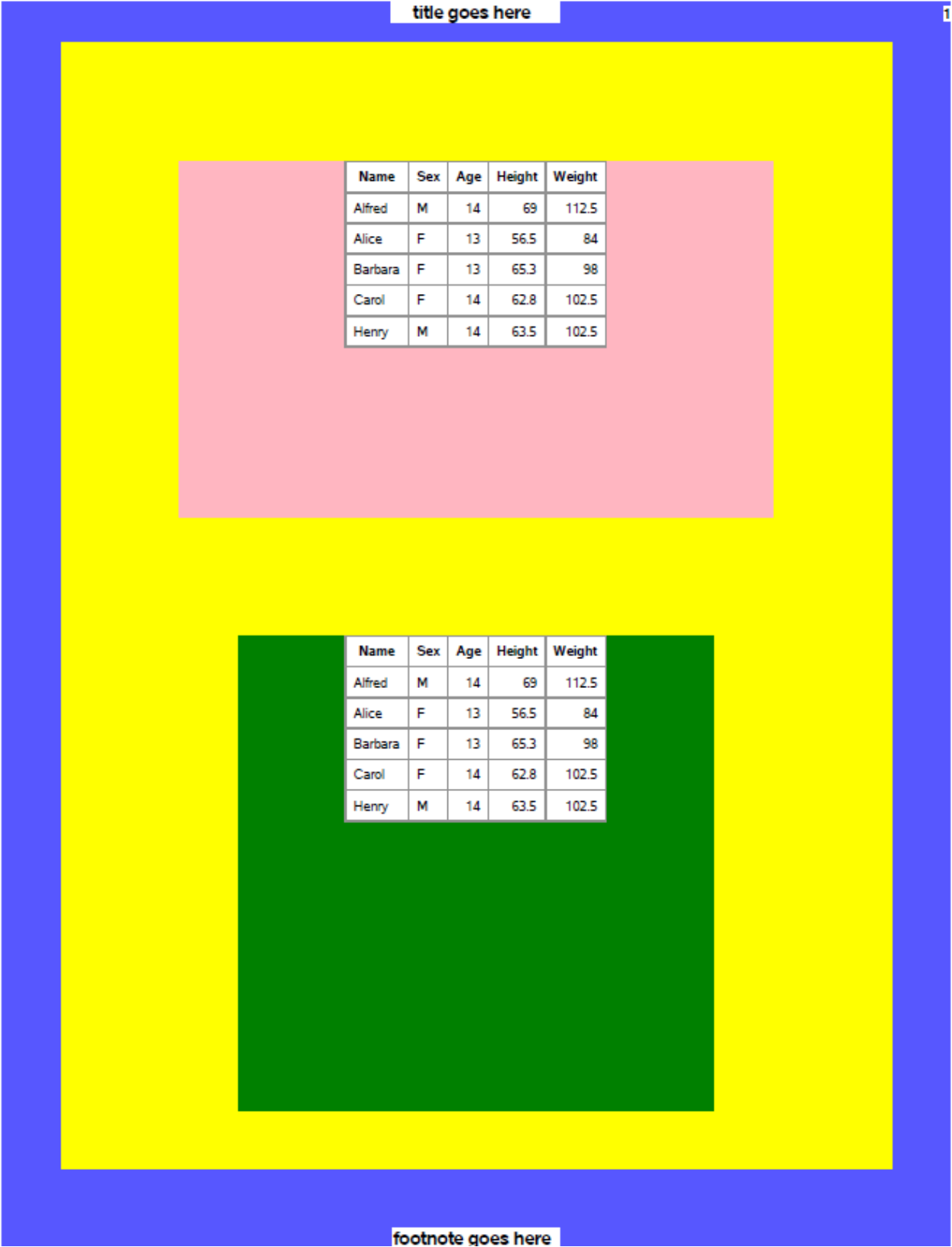


Figure 1. Simple Layout Output

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:

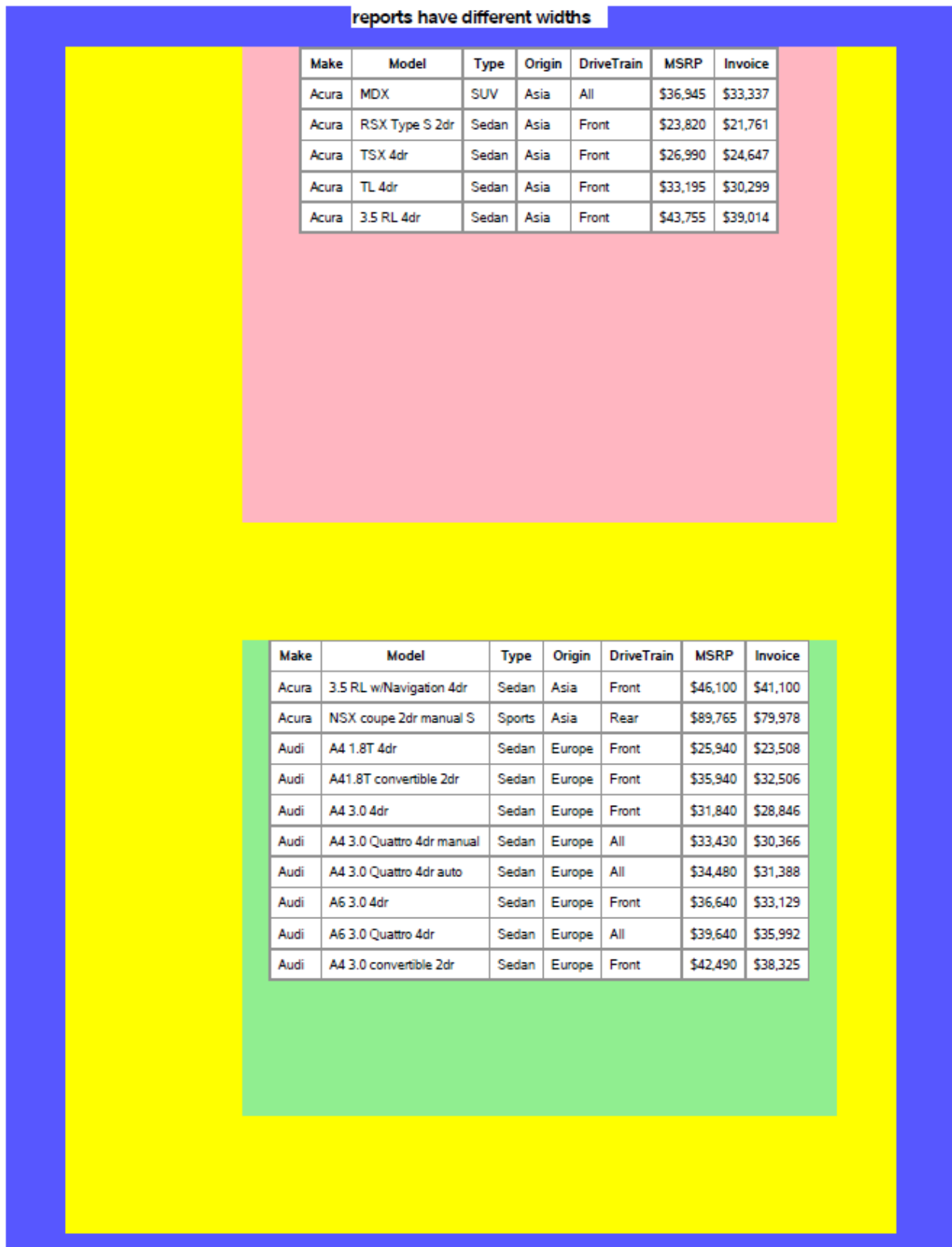


Figure 2. Reports with Different Widths



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   |

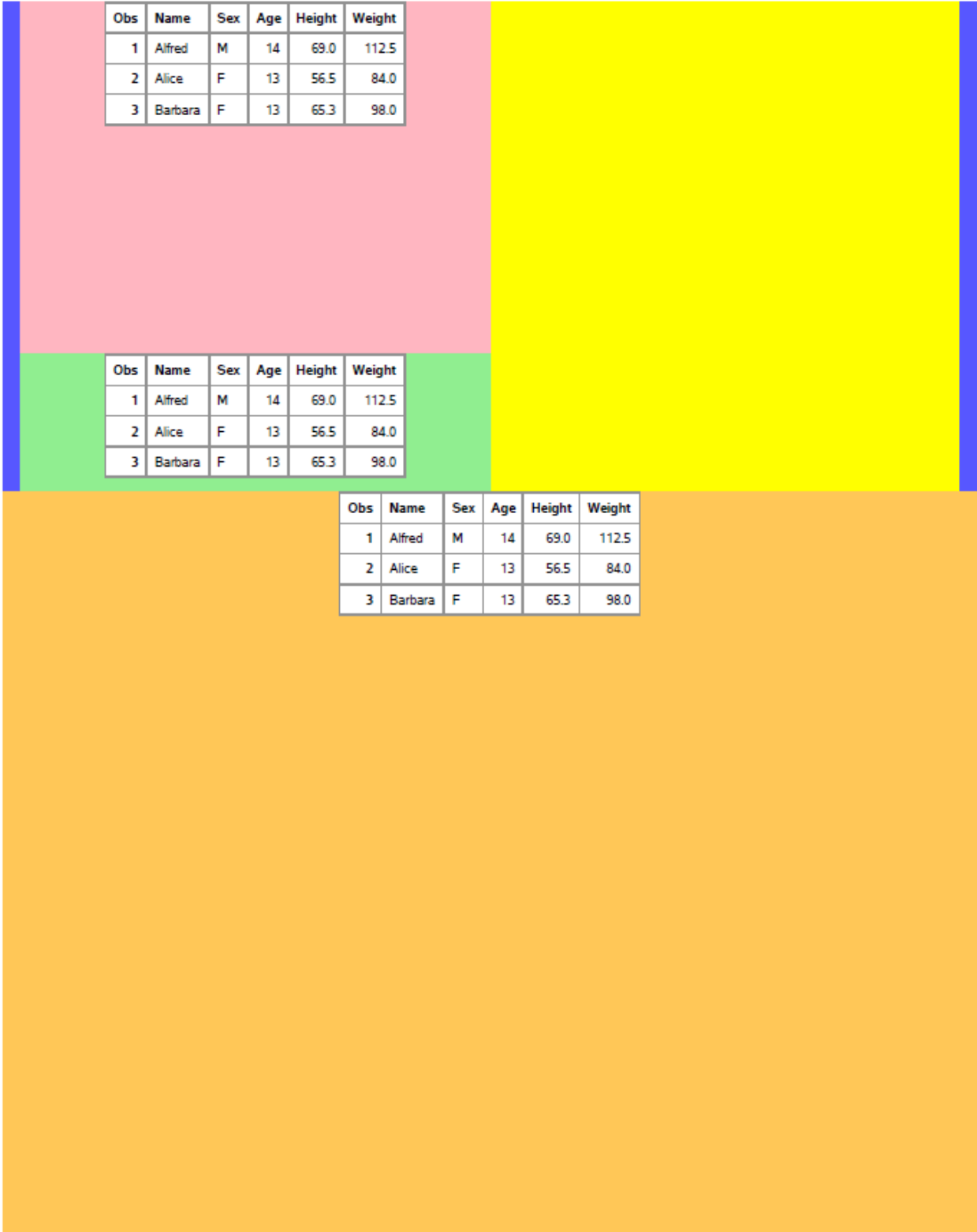
Figure 4. Page 1 of 2 of SAS 9.3 Layout

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 2<sup>nd</sup> 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.

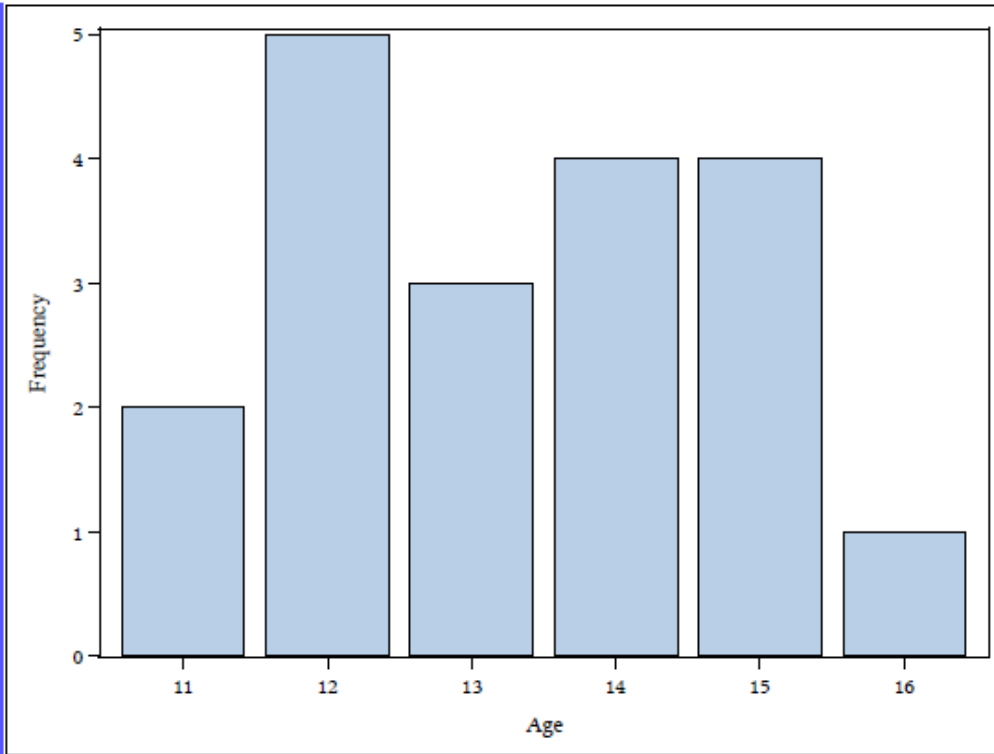
```
proc template;
  define style color;
    parent = Styles.printer;
    style body from body /
      backgroundcolor=light blue;
  end;run;

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;
```

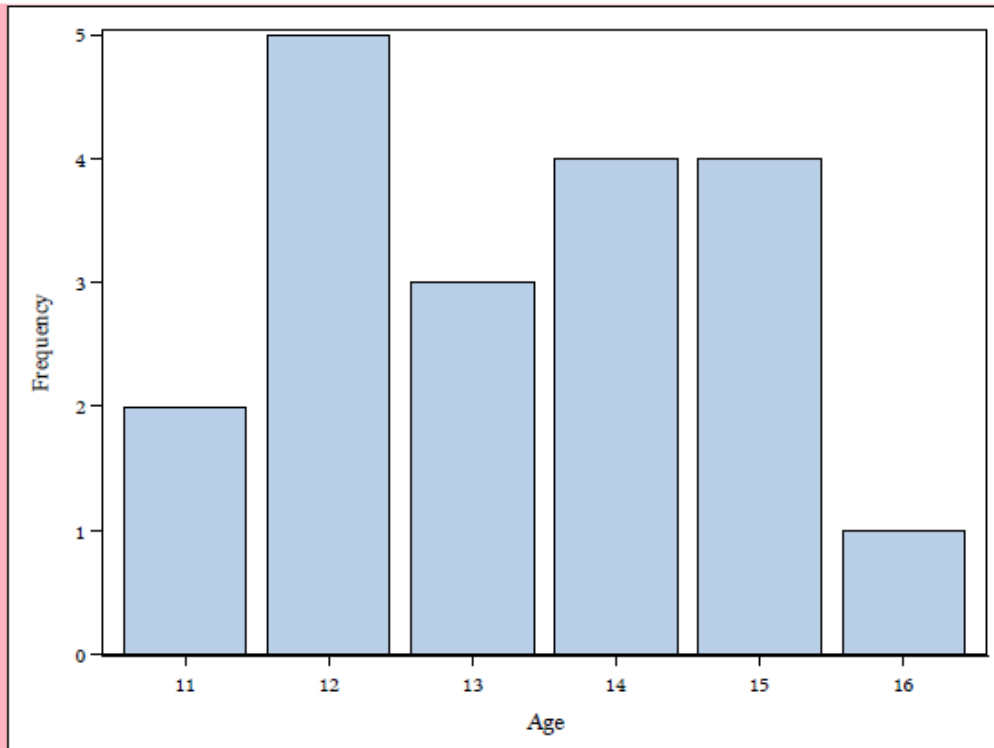
WARNING: Invalid X specified for PDF destination. This option will be ignored.

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 2<sup>nd</sup> 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;
```

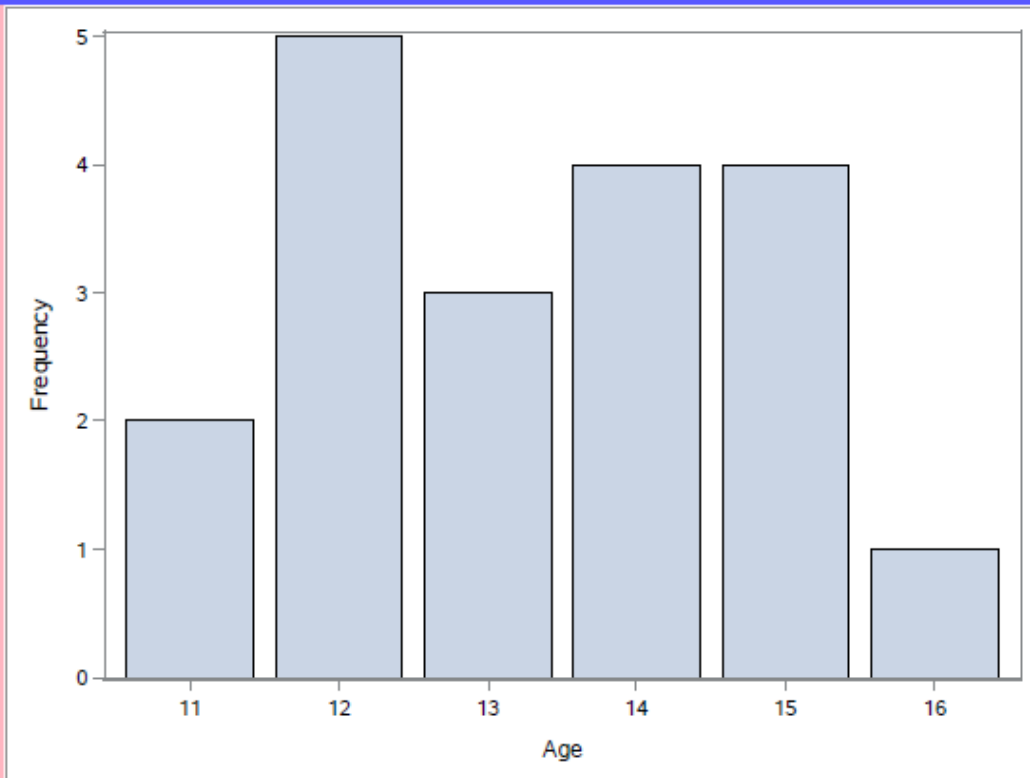


Figure 9. Simple Layout Output with SGPLOT

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;
```

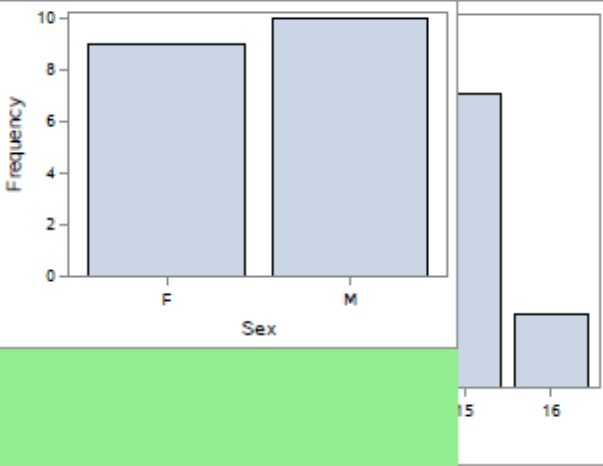


Figure 10. Simple Layout Output with Overlapping Plots

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 2<sup>nd</sup> REGION statement causing the 2<sup>nd</sup> 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.

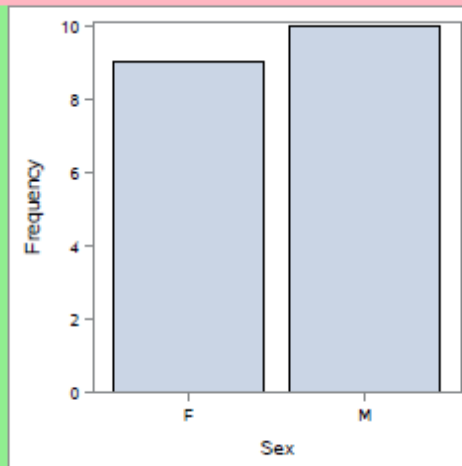
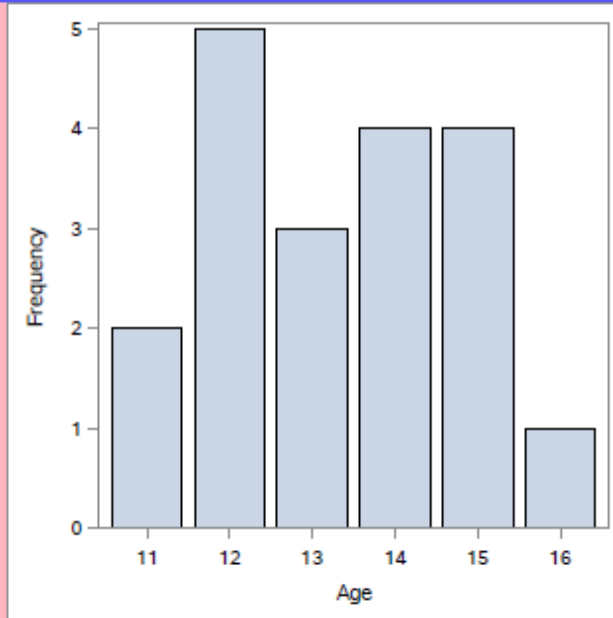
Now let's do another snippet of code with two regions not overlapping and with height and width specified in the ODS GRAPHICS statement. This forces the size of the graphs within the region no matter how big the regions.

```
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;
```



footnote goes here

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 1<sup>st</sup> region. In the 2<sup>nd</sup> region the plot by sex is 3x3 inches. Notice the 1<sup>st</sup> region (in pink) is 5 inches in height. Since we asked for the 2<sup>nd</sup> region to start at the 5-inch mark the 1<sup>st</sup> region fills up everything until that point. Since our layout container specifically asked for a height of 9 inches, the 2<sup>nd</sup> region will take up the remaining space on the page. Therefore, the height of the 2<sup>nd</sup> 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 2<sup>nd</sup> and 3<sup>rd</sup> regions, and also a title and footnote around the PROC REPORT in the 2<sup>nd</sup> 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 2<sup>nd</sup> 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.

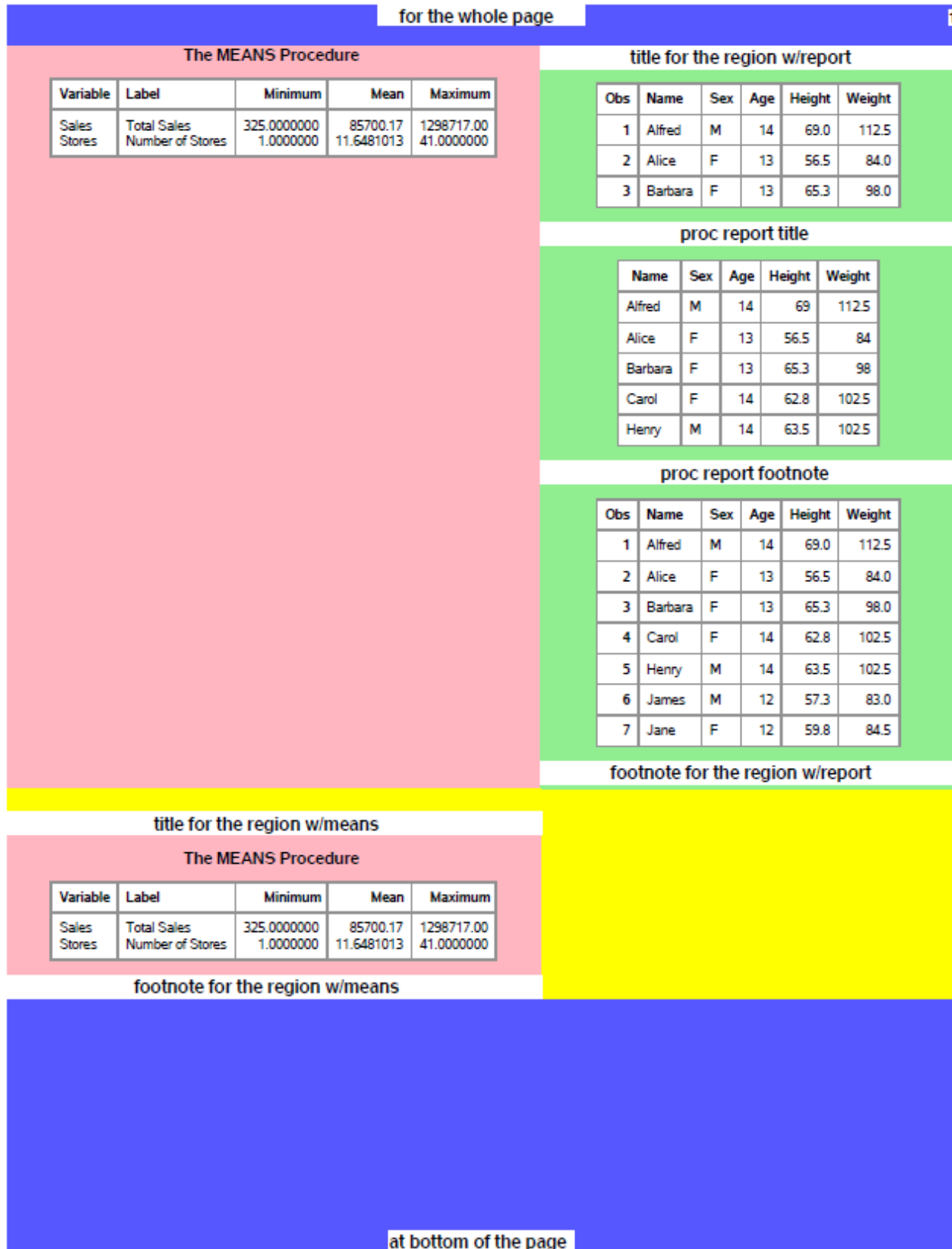


Figure 14. Titles and Footnotes Using SAS 9.4M3

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.

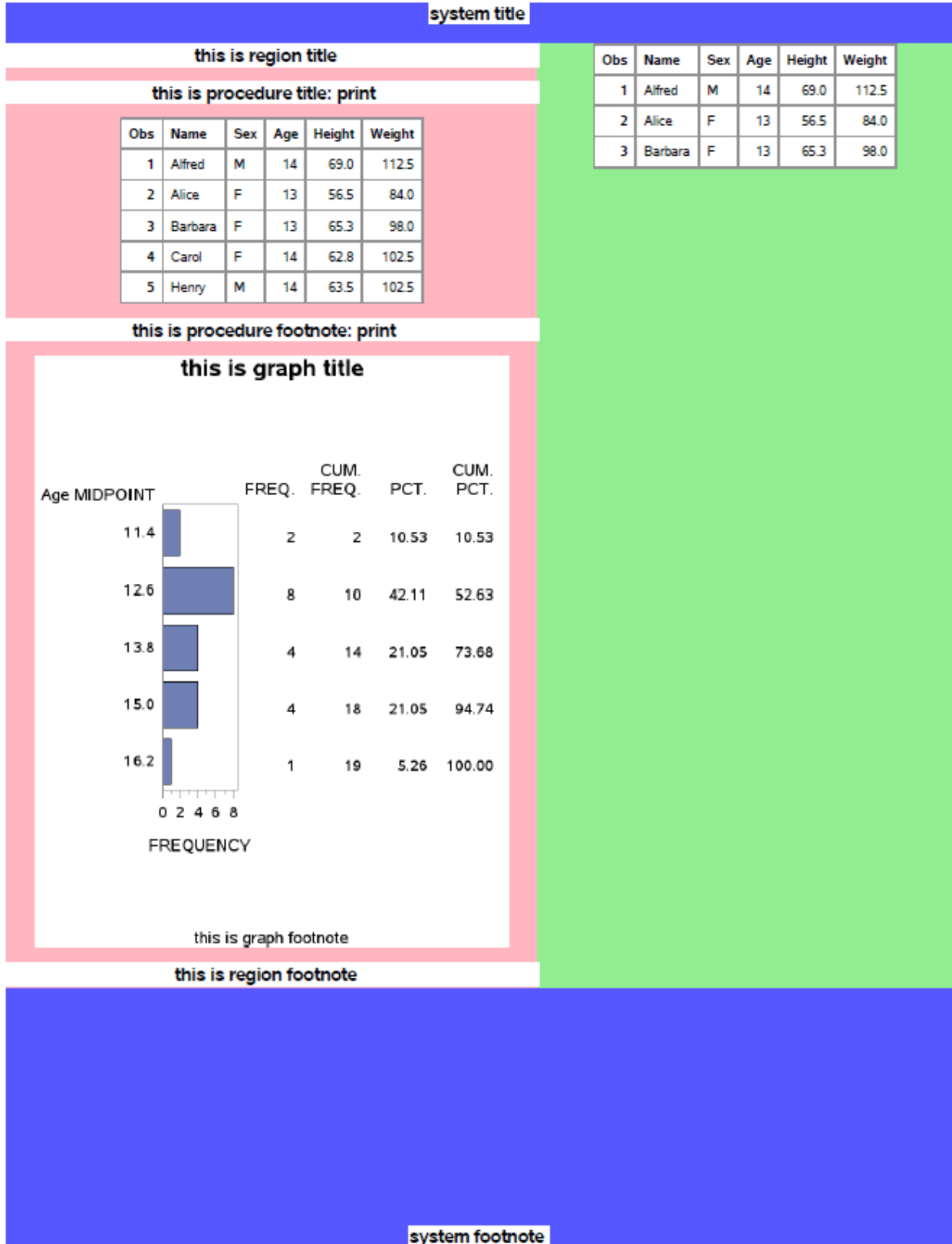


Figure 15. Titles and Footnotes Inside Graph Using SAS 9.4M3



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.

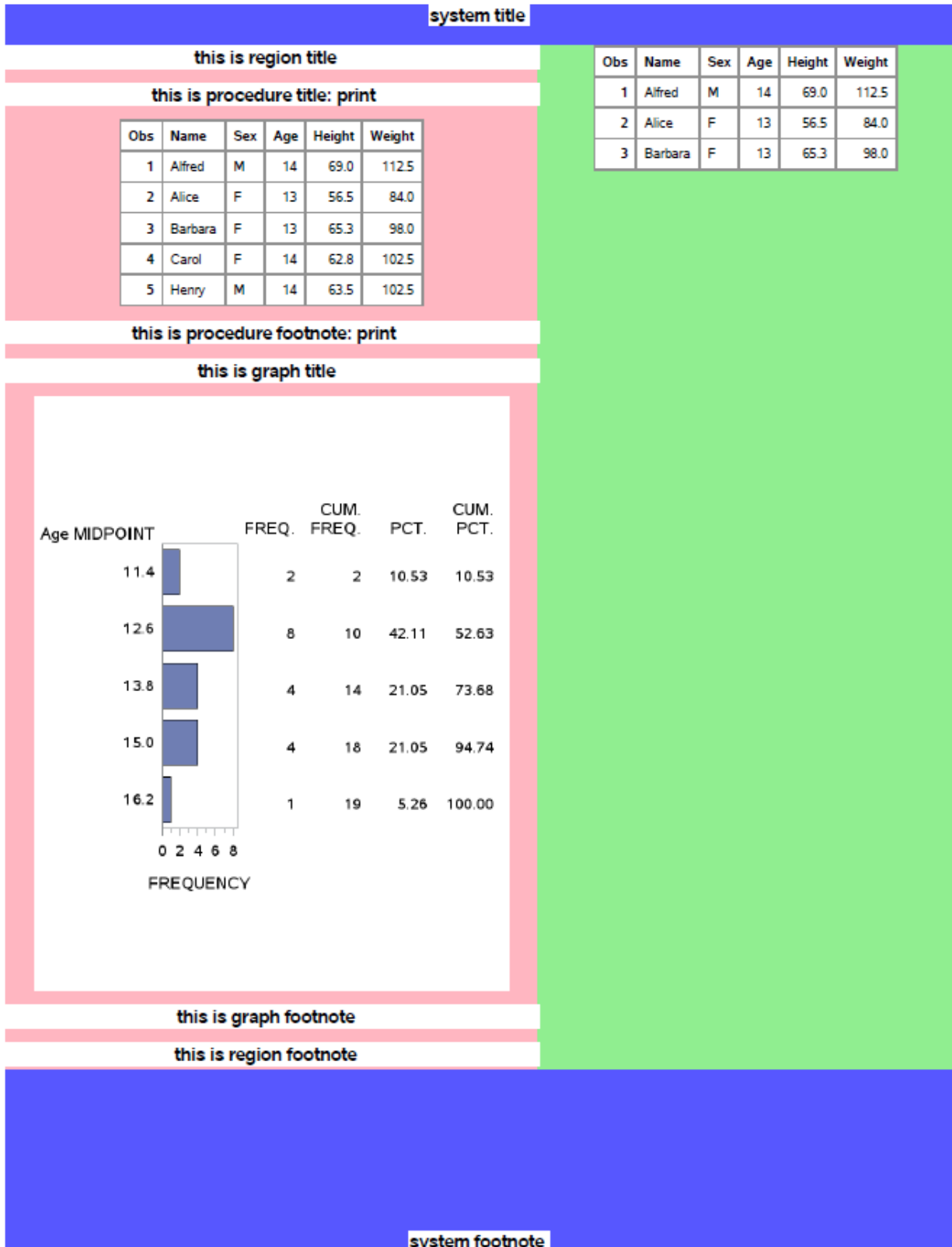


Figure 16. Titles and Footnotes Outside Graph Using SAS 9.4M3

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;

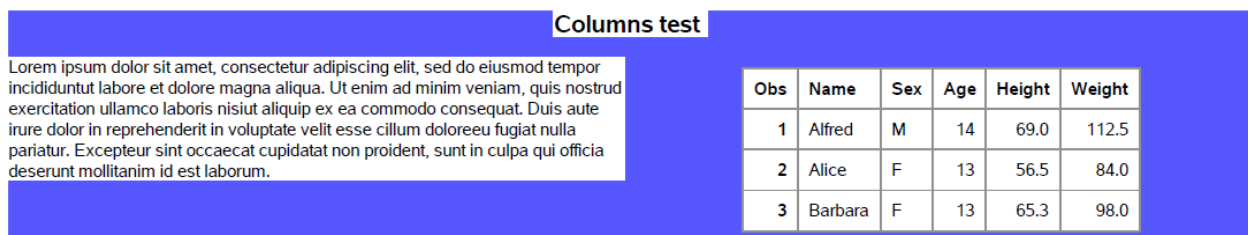
options nodate;
ods pdf file="columns_text.pdf" notoc columns=2 style=color;
title "Columns test";

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 split into two columns. The left column contains a block of Lorem Ipsum text. The right column contains a table with 3 rows and 6 columns: Obs, Name, Sex, Age, Height, and Weight.

| 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;
```

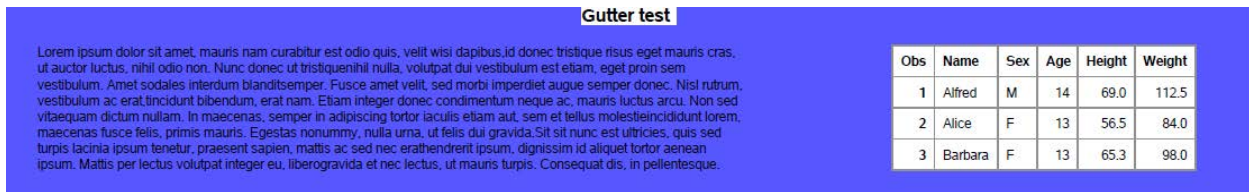
```
ods escapechar="^";
ods pdf file="gutter_text.pdf" notoc style=color;
title "Gutter test";

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.