

**Anything Anywhere All At Once:
Output Graphs, Tables, and Text As Any Composite You Wish**

LeRoy Bessler

Bessler Consulting and Research

Visual Data Insights™

Strong Smart Systems™

Le_Roy_Bessler@wi.rr.com

Mequon, Wisconsin, USA

Abstract

SAS® ODS Graphics is unlimited and empowering. You can put anything anywhere on the page for a composite. Divide the display area however you wish, and imbed whatever suits your communication objective: graphs, tables, text, photos, logos, etc. Your output can be a standalone image, or programmatically imbedded in HTML5, Microsoft Word, PowerPoint, Excel, or PDF. You can make an infographic, anything visual.

One of the key design principles in my book “Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization” is to deliver not only a visual for quick and easy interpretation of the data, but also its precise values for unambiguous and correct understanding of the data. It provides a variety of ways to do that in hundreds of examples, but for some situations a composite of visual and table is the only solution. This tutorial based on Chapter 13 in the book gets you started with learning how to build ANY composite you need, not just its example of two visuals and two tables, all with their individual titles, as well as a common title and common footnote for the package.

Creating Composites of Graphs, Tables, Text, etc. with ODS LAYOUT

NOTE: This tutorial is based on some of the content in Chapter 13 of the book Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization. Use of ODS LAYOUT with ODS HTML5, to imbed a composite in a web page, is included in Chapter 14 of the book—that topic is not covered in this paper.

The advantages of composites include the ability to present:

- (a) Alternative graphs for the same data set
- (b) A graph with a companion table of precise values which could not be included inside the graph with data labels, an axis table, or annotation
- (c) Any collection of graphs and tables and text

To assemble a composite that includes one or more tables, any table can be easily deployed using ODS LAYOUT **only if it is first packaged as an image**. You can use the ODS PRINTER destination to create a table as an image file.

Then you can use the combination of ODS LAYOUT and ODS PRINTER to assemble a composite of images, whether graphs or tables as images or any image. The composite is output as a standalone image file on disk.

If using ODS LAYOUT in combination with an ODS destination other than PRINTER, you can imbed the composite Image in a PDF document, an ODS HTML5 web page, in an ODS PowerPoint slide, or in an ODS Word document.

Technical Introduction

There are two versions of ODS LAYOUT. The ODS LAYOUT ABSOLUTE statement is used to achieve total control of the layout—Anything Anywhere. The ODS LAYOUT GRIDDED statement is easier to use, but provides less control. If you want increase your control with GRIDDED layout, the coding becomes more complex. This paper uses ABSOLUTE layout only. My results when using GRIDDED layout were not to my satisfaction.

ODS LAYOUT ABSOLUTE can be used only with ODS PRINTER or ODS PDF.

ODS LAYOUT GRIDDED can used with ODS destinations PRINTER, PDF, POWERPOINT, WORD, and HTML5. However, you can use ODS PRINTER for an absolute layout and imbed that image file in any of the other destinations as a second step.

Besides PROC PRINT, the SAS report writing procedures PROC REPORT and PROC TABULATE can also be used with ODS PRINTER to produce tabular output as an image file.

Many of the SAS procedures that are not explicitly intended as general-purpose reportwriting tools also produce tables of information. For these procedures, there may be multiple tables that are output. They comprise ODS Output Objects. For PROC UNIVARIATE there are five. How to capture and display them as an image is shown in the book's Chapter 13.

NOTE: In this paper you will see DPI=300. It is a holdover from coding for the book. It is not mandatory, but it makes text, lines, and curves smoother.

Before demonstrating the use of ODS LAYOUT, the first step is to create the tables as images with ODS PRINTER and PROC PRINT.

Chapter 13 in the book covers more information and more methods than this paper, as well as more use of macros for that process.

In this tutorial, the focus is on the old workhorse of SAS software for reporting, PROC PRINT, which has been adequate for almost all of my tabular reporting needs.

For graphs, the ODS GRAPHICS statement has the BORDER/NOBORDER option, where BORDER is the default. However, when a table is created as an image, the process cannot provide a border. A second step is required to apply the border. It relies on a macro that is provided in this paper.

Creating a Table As an Image with ODS PRINTER

Student Information With Name Initial J

Name	Age	Height
James	12	57.3
Jane	12	59.8
Janet	15	62.5
Jeffrey	13	62.5
John	12	59.0
Joyce	11	51.3
Judy	14	64.3

Figure 1. Table Created As an Image With Default Formatting

Code used to create Figure 1:

```
options nodate nonumber;
options papersize=(1.65in 2.8in);
ods results off;
ods _all_ close;
ods printer
  file="C:\temp\Fig-1_TableAsImageUsingDefaults.png"
  printer=PNG dpi=300;
title1 height=0.05in " "; /* white space */
title2 "Student Information";
title3 "With Name Initial J";
proc print data=sashelp.class noobs;
where Name=: 'J';
var Name Age Height;
run;
ods printer close;
```

Adding a Border to Any Image

Though the ODS Graphics statement has a default of BORDER to provide a border around the image created with any ODS Graphics procedure, here no such option is available.

The lack of a border is the explanation for the apparent indent of Figure 1 from the margin. Below a macro is used to apply a border to Figure 1. **It can be used with any image, not only with ones created with ODS Graphics or some other graphics software.**

Student Information With Name Initial J		
Name	Age	Height
James	12	57.3
Jane	12	59.8
Janet	15	62.5
Jeffrey	13	62.5
John	12	59.0
Joyce	11	51.3
Judy	14	64.3

Figure 2. A border added to Figure 1.

Code used to create Figure 2:

```
/* If you store the macro below in C:\MyMacros
instead of keeping it in-stream here, use
%include "C:\MyMacros\ApplyBorderToAnyImage.sas"; */

%macro ApplyBorderToAnyImage(in=,out=,height=,width=);
ods results off;
ods _all_ close;
title; footnote;
options center nonumber nodate;
options papersize=(&width &height);
goptions device=PNGT transparency;
ods printer printer=PNG dpi=300 file="&out";
ods layout absolute;
ods region x=0in y=0in width=&width height=&height;
data _null_;
dcl odsout obj();
obj.image(file: "&in");
run;
ods region x=0in y=0in width=&width height=&height;
```

```

proc gslide border;
run; quit;
ods layout end;
ods printer close;
%mend ApplyBorderToAnyImage;

options mprint;
%ApplyBorderToAnyImage(in=C:\temp\Fig-1_TableAsImageUsingDefaults.png,
out=C:\temp\Fig-2_Fig-1_With_Border.png,
width=1.65in,
height=2.8in);

```

Building a Composite with ODS LAYOUT and ODS PRINTER

Every ODS destination except LAYOUT is limited on how you can arrange content in the output entity that it delivers. ODS LAYOUT is the great enabler. With ODS LAYOUT and ODS PRINTER, you can assemble whatever composite you wish and output it as a standalone image. That image can be imbedded in the output for another ODS destination, such as a web page, PDF document, WORD document, POWERPOINT slide, EXCEL worksheet, etc.

Having emphasized the ability to imbed a composite in one's preferred document type, the immediate order of business is to demonstrate how to *build* a composite. The coding can be lengthy, but conceptually it is very straightforward, requiring no abstruse folderol.

Figure 3 is not the most elaborate composite that one could contrive, but it is a good learning tool. It was *my* learning tool.

This example of using ODS LAYOUT for a composite assembles a two-by-two array with a row of two graphs above a row of two tables. The array can consist of any number of rows and columns. It need not consist of recognizable rows or columns of elements. The collection of regions inside the layout need not be an array, matrix, grid, or whatever you might call its configuration. That is clear from the presence of an overall title and overall footnote outside the two-by-two array. The regions can be any rectangular or square shape. Areas of the overall layout can be unallocated and unused. Those spaces are voids. From row to row the number of columns can vary. Content need not be organized in any rows or columns. The total area can be broken up into any arrangement of internal regions. The only limit on region size, region shape, or total number of regions is readability. Any ODS destination allows you to stack content, but arranging the content in arbitrary way is possible only with ODS LAYOUT.

In the book, the result in Figure 3 is created as Figure 13-7, but the tables are created as image files using a macro provided in Chapter 13.

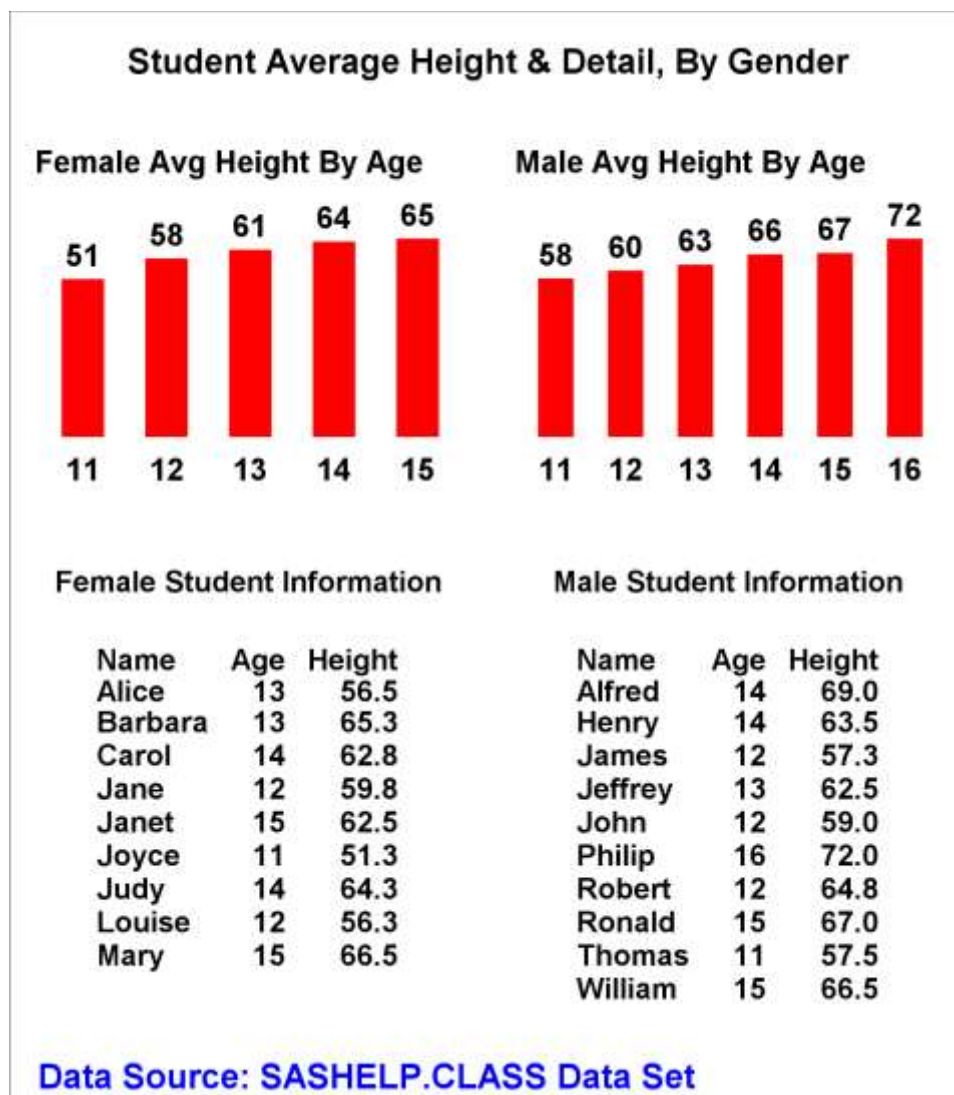


Figure 3. Composite with a Border Consisting of Two Graphs, Two Tables, a Common Title, and a Common Footnote in Six Regions of ODS LAYOUT

Code to create the composite:

```
/* NOTE: Since the image might require tuning
to get to a satisfactory result,
it is convenient to gather controls
that are likely to need change
up here to the top of the code. */

%let GeneralFontSize = 11pt; /* for all text,
except any to which override is applied */
```

```

%let TableCellHeight = 8pt;
    /* used for creating tables as images
       with table rows compacted */

%let WhiteSpace = 0.225in; /* between graphs row
                             and tables row */
/* title and footnote 13pt Arial Bold */
%let TitleFootnoteTextAttrs =
    %str(fontfamily='Arial'
        font_size=13pt font_weight=Bold);

/* center them and make them black */
%let CommonTitleAttrs =
    &TitleFootnoteTextAttrs
    just=center color=black;
%let CommonFootnoteAttrs =
    &TitleFootnoteTextAttrs
    just=left color=blue;

%let CommonTitle =
    %str(Student Average Height & Detail, By Gender);
%let CommonFootnote =
    %str(Data Source: SASHELP.CLASS Data Set);

/* Prep to build tables as images:
   create a temporary ODS style
   that is deleted after the build.
   Using PROC TEMPLATE for that */

proc template;
define style ToBeDeleted;
    parent=styles.PRINTER;
    class Fonts /
        'TitleFont'    =
            ("Arial",&GeneralFontSize,Bold)
        'headingFont' =
            ("Arial",&GeneralFontSize,Bold)
        'docFont'      =
            ("Arial",&GeneralFontSize,Bold)
    ;

```

```

class Header      /
  backgroundcolor = white color = black;
class RowHeader /
  backgroundcolor = white color = black;
class data        /
  backgroundcolor = white color = black;
class Table / rules=none frame=void;
               /* remove the table grid */
class Data        /
  cellheight = &TableCellHeight;
class RowHeader /
  cellheight = &TableCellHeight;
end;
run;

/* The style name created is ToBeDeleted.
   Once created in your SASUSER Template Store,
   any style is permanent unless deleted.
   This style is specific to this experiment.
   I will delete it after done using it. */

/* building tables as images */

/* Defaults are DATE NUMBER CENTER */
options nodate nonumber;

options papersize=(2.11in 2.4in);
      /* table image width & height */

ods results off;
ods _all_ close;

ods printer style=ToBeDeleted dpi=300
      printer=PNG file="C:\temp\RightTable.png";

title1 justify=center color=black
      "Male Student Information";
proc print data=SASHELP.CLASS noobs;
where Sex EQ 'M';
id Name;

```

```

var Age Height;
run;

ods printer style=ToBeDeleted dpi=300
  printer=PNG file="C:\temp\LeftTable.png";

title1 justify=center color=black
  "Female Student Information";
proc print data=SASHELP.CLASS noobs;
where Sex EQ 'F';
id Name;
var Age Height;
run;

title1; /* Prevent TITLE1 being inherited.
  In THIS example, it would cause
  "Female Student Information"
  to also appear at the TOP of the composite. */

/* End the table build as images
  by closing the ODS PRINTER destination
  and deleting the temporary style. */

ods printer close;

/* The style used is specific to this experiment. I am deleting it. */

proc template;
delete ToBeDeleted / store=sasuser.templat;
run;

/* Now build the composite */

/* Defaults are DATE NUMBER CENTER */
options nodate nonumber;
options papersize=("5in" "5.69in");
  /* composite width & height */

ods results off;
ods _all_ close;

```

```

ods printer printer=PNG dpi=300
  /* default style for ODS PRINTER accepted */
  file="C:\temp\Fig-3_Composite.png";

/* ODS PRINTER creates the composite image */
/* ODS LAYOUT places the correct items in the desired places */

ods layout absolute;

/* Location and size of desired placement
   is defined by REGION statements. */

/* Inlay a common title. */

ods region x=0in y=0in
  /* These x & y values are at the upper left
     corner of the "paper".
     They define the location
     of the upper left corner of the region. */
  width=5in height=0.6in;
  /* width is measured right from the left */
  /* height is measured down from the top */

/* Use PROC ODSTEXT for all text parts
   not created inside the images.
   Here, the common title for the composite. */

proc odstext;

/* First, some white space (if needed): */
p ' ' / style={font_size=0.1in};

/* Now the title: */
p "&CommonTitle" / style={&CommonTitleAttrs};

run;

/* White space can provide a finer increment
   than the effect of
   changing Y= on the REGION statement */

```

```

/* setup for BOTH graphs: */
ods graphics on / reset=all
  /* override default SCALE=ON */
  scale=off width=2.5in height=2in noborder;
  /* default is BORDER */

/* define the region for the left graph */
ods region x=0in y=0.6in
/* 0.6 inches down from the upper left corner
  of the composite image */
  width=2.5in height=2in;

ods graphics / reset=index
  /* RESET=INDEX means inherit all prior
    ODS GRAPHICS statement specifications,
    but allow IMAGENAME to be defined here. */
  imagename="LeftGraph";

title justify=left
  font=Arial Bold height=&GeneralFontSize
  'Female Avg Height By Age';
proc sgplot data=sashelp.class noborder;
where Sex='F';
vbar age / response=height stat=mean
  displaybaseline=off
  datalabel
  datalabelattrs=(family=Arial
    size=&GeneralFontSize weight=Bold)
  nooutline
  fillattrs=(color=red)
  barwidth=0.5;
yaxis display=none;
xaxis display=(nolabel noline noticks)
  valueattrs=(family=Arial
    size=&GeneralFontSize weight=Bold);
format age 2. height 2.;
run;

/* same code as LeftGraph except for x=, imagename=, title text, Sex= */

```

```

ods region x=2.5in y=0.6in
    width=2.5in height=2in;
ods graphics / reset=index
    imagename="RightGraph";
title justify=left
    font=Arial Bold height=&GeneralFontSize
    'Male Avg Height By Age';
proc sgplot data=sashelp.class noborder;
where Sex='M';
vbar age / response=height stat=mean
    displaybaseline=off
    datalabel
    datalabelattrs=(family=Arial
        size=&GeneralFontSize weight=Bold)
    nooutline
    fillattrs=(color=red)
    barwidth=0.5;
yaxis display=none;
xaxis display=(nolabel noline noticks)
    valueattrs=(family=Arial
        size=&GeneralFontSize weight=Bold);
format age 2. height 2.;
run;

/* Insert the LeftTable image
    below white space */

ods region
    x=0in y=2.6in /* at the bottom of the region
        allocated to left graph */
    width=2.5in height=2.84in;
/* BIGGER than image dimensions, but harmless.
    (If too small, the image does not appear.) */

proc odstext;
p ' ' / style={font_size=&WhiteSpace};
run;

data _null_;
dcl odsout obj();

```

```

obj.image(file: "C:\temp\LeftTable.png");
run;

/* insert the RightTable image
   below white space */

ods region
  x=2.5in y=2.6in /* at the bottom of the region
                  allocated to right graph */
  width=2.5in height=2.84in;

proc odstext;
p ' ' / style={font_size=&WhiteSpace};
run;

data _null_;
dcl odsout obj();
obj.image(file: "C:\temp\RightTable.png");
run;

/* Inlay a common footnote. */

ods region
  x=0in y=5.44in /* at the bottom of the regions
                  allocated to the graphs */
  width=5in height=0.25in;
/* the bottom of this region is at 5.69in which
   corresponds to the height of the composite */

proc odstext;
p "    &CommonFootnote" /
  style={asis=on &CommonFootnoteAttrs};
/* ASIS=ON preserves
   the three leading blanks for an indent */
run;

ods layout end;
ods printer close; /* Finished */

/* Fig-3_Composite.png, LeftTable.png, and

```

```

    RightTable.png are in C:\temp.
    The table images can be deleted or ignored. */

/* NEXT: Use a macro to apply a border to the composite.
    This border is optional. */

/* About Borders:

    Borders on graphs were suppressed, using
    NOBORDER option on ODS GRAPHICS statement.

    Borders on the tables as images
    COULD BE applied,
    but it requires the same post-processing
    to be used here for the composite. */

/* Replace C:\SharedCode below with whatever is the location
    where you have stored the macro.
    OR paste in the macro source code provided for Figure 2. */
%include "C:\SharedCode\ApplyBorderToAnyImage.sas";

/* Apply a border to the composite */
%ApplyBorderToAnyImage(
in=C:\temp\Fig-3_Composite.png,
out=C:\temp\Fig-3_CompositeWithBorder.png,
width=5in, height=5.69in);

/* Now, Fig-3_Composite.png can be deleted or ignored. */

```

NOTE: The lengthy code above is available via email request.

Capturing ODS Objects from SAS Procedure Output As Images

If any SAS procedure generates a table as part of its function, then you can substitute its invocation for the PROC PRINT procedure step when using ODS PRINTER to output a table as an image. Some SAS procedures generate multiple tables, in which case you can programmatically select the table or tables that you want to present as an image. The first step is to discover the name of the tables as ODS Output Objects to be used on an ODS SELECT statement. See Chapter 13 in the book for the details and an example, or search the SAS documentation for examples of using the ODS TRACE and ODS SELECT statements.

Conclusions

Here you see the process used to assemble a two-by-two package of individually titled graphs and individually titled tables between a common title and common footnote. The alternative arrangements of graphs, tables, any type of images, and text is actually unlimited. This example and code should get you started on your own. See Reference 1 for more examples of ODS Graphics' capability to create an enormous range of types of individual graphs. Numerous other ways to create composites of graphs (without tables) are covered in Chapters 9 and 10.

For Further Reading

Bessler, LeRoy. "Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization". New York, NY, USA: Apress, 2023. Chapter 13 of this book served as source for some of this paper's content. Chapters 9 and 10 provide other ways to create composites of graphs, but without tables.

Bessler, LeRoy. "My Graphic Inventions: You Can Do These, Too", 2023 Michigan SAS Users Group Conference, Livonia, MI, WI, USA. These are some examples from other book chapters.

Author Information

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

Le_Roy_Bessler@wi.rr.com
LeRoy Bessler PhD
Bessler Consulting and Research
Visual Data Insights™
Strong Smart Systems™
Mequon, Wisconsin, USA

LeRoy Bessler, author of the book "Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization", is a consultant, on-site trainer, data analyst, programmer, application developer, data visualization aficionado, and data artist. He is the world's longest serving advocate for and demonstrator of design principles for graphics and best use of color in an ongoing quest for how best to deliver Visual Data Insights™ with SAS software. He has supported SAS servers, SAS software, SAS data, and SAS users, concurrent with his own work as a SAS practitioner. His special interests are data visualization, and software-intelligent application development for reliability, reusability, maintainability, extendability, and flexibility to deliver Strong Smart Systems™.

SAS and 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.
Visual Data Insights and Strong Smart Systems are trademarks of LeRoy Bessler PhD.