SAS® Graphs for BlackBerry, iPhone, PowerPoint, Word, Web/HTML, PDF, or RTF

LeRoy Bessler, Assurant Health, Milwaukee, Wisconsin, USA, bessler@execpc.com

Abstract

This paper provides tips, techniques, and tools needed for ad hoc graphs for presentations, or production graphs for management reporting or a performance tracking "dashboard": how to select the best device driver and the right options, depending on the purpose; how to design, create, and deliver graphs to a BlackBerry, iPhone, or any other small screen email device; how to create graphs direct-to-disk for subsequent manual insert to Microsoft PowerPoint or Microsoft Word; how to "right-size" graphs for those destinations; how to package and "right-size" graphs for web/HTML, PDF, and RTF; why and how to use ALT text for web graphs; and how to create graphs directly in a set of PowerPoint slides. The ideas, insights, and practical experience can help you to best exploit the versatile power of SAS/GRAPH and ODS.

Introduction

The Best Choice must be a communication-effective choice and a reliable choice.

See Appendices 1 and 2 for how to create and deliver readable graphs for a BlackBerry and how to create a graph directly in a PowerPoint slide.

Sometimes finding the best choice is a matter of finding the right choice.

It can be challenging when, without any warning in the SAS log, a device driver other than what you specified is used, or when the unattractive and can-be-hard-to-read SIMULATE font ("Is that 0 or D?") is substituted for what you specified. The SAS/GRAPH manual is over 1,600 pages, and the ODS manual is over 700 pages. Add to that information at support.sas.com in SAS Usage Notes, Technical Notes, Communities, and Samples, at sascommunity.org, at SAS-L, and elsewhere. Rather than suggest that you wade through it all, my modest goal here is to give you help likely to work as a reliable model or starting point, so that you can efficiently exploit that huge knowledge base along the shortest route to a solution that meets your needs exactly.

This paper:

(a) includes in Appendix 1 and 2 stand-alone special-topic papers on SAS graphs for BlackBerry, iPhone, or other small screen devices and on creating SAS graphs directly in PowerPoint slides;

(b) shows how to customize a basic version of the three most popular types of presentation graphs to implement the principles of communication effectiveness presented in Reference 1;

(c) provides a recommendation of SAS/GRAPH device drivers for different situations;

(d) demonstrates how to prepare graphs as Graphic Stream Files (GSF) on disk for insert into Microsoft PowerPoint or Microsoft Word and how to customize them after insert, if desired;

(e) provides examples for ODS destinations RTF, PDF, and HTML, with the HTML discussion limited to how to optimally size a graph for a web page (the HTML destination and web design are covered extensively in Reference 1); and

(f) focuses on "Accessibility"—how to reach the widest audience with graphic data presentation.

Improving a Pie Chart (Text is more readable for both charts if not reduced to half page width.)

Figure 1: Default Pie Chart

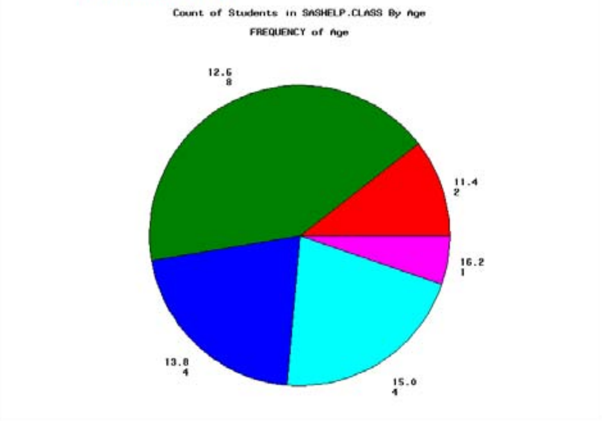
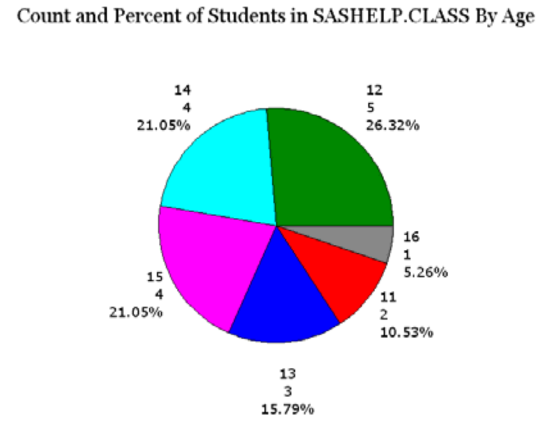


Figure 2: Custom Pie Chart



Here is the code used to create the Custom Pie Chart:

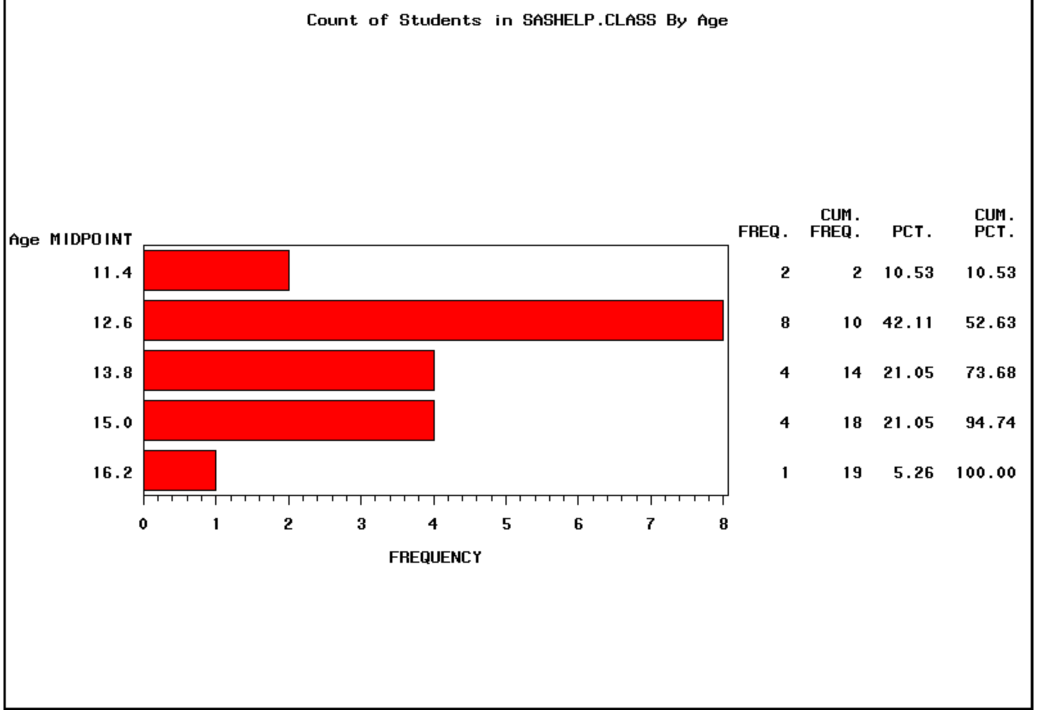
```
goptions border; /* border around image, but adding Word border in the paper */
goptions ftext='Verdana'; /* font  for text for which no font=   is assigned */
goptions htext=3.75 PCT;  /* height of text for which no height= is assigned */
title1 height=1 PCT ' ';          /* white space at top    of graph */
footnote1 angle=0 height=1 PCT ' ';   /* white space at bottom of graph */
footnote2 angle=+90 height=1 PCT ' '; /* white space at right  of graph */
footnote3 angle=-90 height=1 PCT ' '; /* white space at left   of graph */
title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Figure 2: ' color=CX0000FF 'Custom Pie Chart'
  justify=LEFT height=2.5 PCT " "; /* white space before the next TITLE line */
title3 font='Georgia' height=5 PCT color=CX000000
  'Count and Percent of Students in SASHELP.CLASS By Age';

/* in HEIGHT= above, and in OFFSET= later, PCT means Percent of total graph height */

proc gchart data=sashelp.class;
* For a pie chart of total weight of students in each Age group,
  use SUMVAR=WEIGHT after the / in the PIE statement. *;

pie age /* create pie slices for AGE */
  /
  discrete   /* use the discrete values of AGE,
                not midpoints of default subranges  */
  noheading  /* suppress the default heading
                (in this case, "FREQUENCY of Age") */
  descending /* order the slices by decreasing size */
  percent=outside;
/* Display Percent of Total Frequency for each slice.
   VALUE and SLICE (Slice Name) are defaults.
   Other options are NONE, INSIDE, and ARROW.
   "ARROW" connects display outside to the slice. */
/* To turn off pie slice color, include a PATTERN statement like this:
   PATTERN1 VALUE=PEMPTY REPEAT=99;
   99 can be any number greater than or equal to the number of slices.
   For how to control pie slice color,
   please see Reference 2 or the SAS/GRAPH manual. */

run; quit;
```

Here is the code to create the Default Pie Chart:

```
/* three title statements */
proc gchart data=sashelp.class;
pie age; run; quit;
```

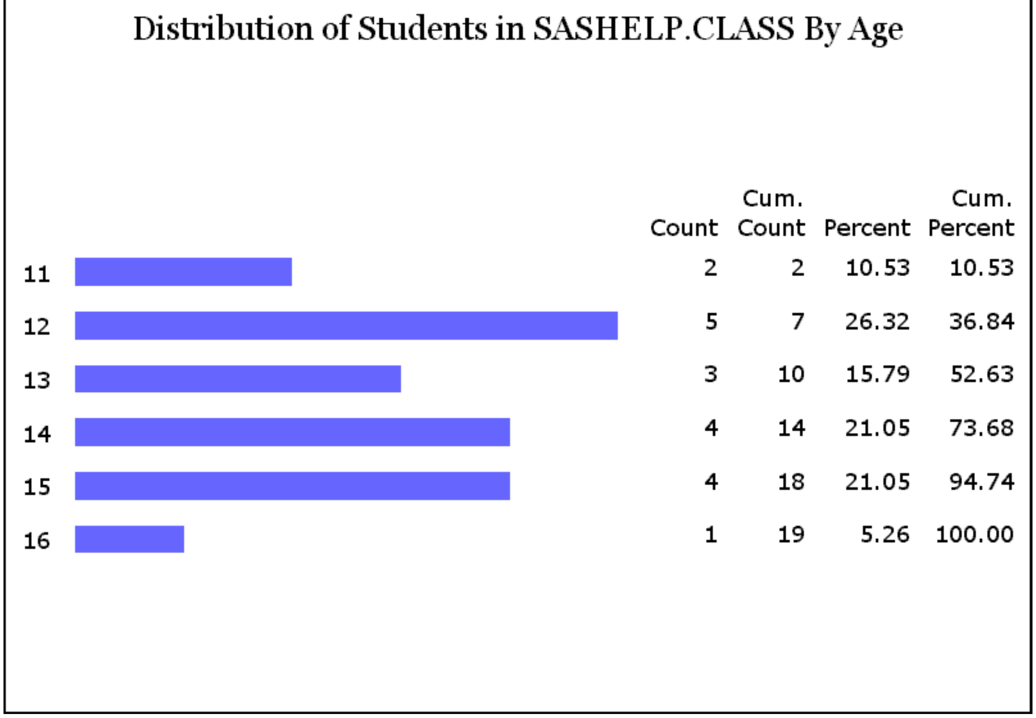
Improving a Horizontal Bar Chart

Figure 3: Default Bar Chart



Age MIDPOINT	FREQ.	CUM. FREQ.	PCT.	CUM. PCT.
11.4	2	2	10.53	10.53
12.6	8	10	42.11	52.63
13.8	4	14	21.05	73.68
15.0	4	18	21.05	94.74
16.2	1	19	5.26	100.00

Figure 4: Custom Bar Chart



	Count	Cum. Count	Percent	Cum. Percent
11	2	2	10.53	10.53
12	5	7	26.32	36.84
13	3	10	15.79	52.63
14	4	14	21.05	73.68
15	4	18	21.05	94.74
16	1	19	5.26	100.00

Customization replaces default subrange midpoints for Age with the actual discrete Age values, eliminates the unneeded response axis entirely, removes the midpoints axis line and chart frame, suppresses the midpoints axis label which is already identified in the graph title, takes control of the column label text, and uses a lighter bar color that is less image-overwhelming.

Here is the code used to create the Custom Bar Chart:

```
goptions border; /* border around image, but adding Word border in the paper */
goptions ftext='Verdana'; /* font  for text for which no font=   is assigned */
goptions htext=3.75 PCT;  /* height of text for which no height= is assigned */

title1 height=1 PCT ' ';              /* white space at top    of graph */
footnote1 angle=0 height=1 PCT ' ';   /* white space at bottom of graph */
footnote2 angle=+90 height=1 PCT ' '; /* white space at right  of graph */
footnote3 angle=-90 height=1 PCT ' '; /* white space at left   of graph */

title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Figure 4: ' color=CX0000FF 'Custom Bar Chart'
  justify=LEFT height=2.5 PCT " "; /* white space below TITLE2 */

title3 font='Georgia' height=5 PCT color=CX000000
  'Distribution of Students in SASHELP.CLASS By Age';

pattern1
  value=solid     /* solid fill for the bars   */
  color=CX6666FF; /* lighter browser-safe blue */

axis1
  label=none /* suppress the axis label */
  major=none /* suppress tick marks for every value along the axis */
  minor=none /* suppress tick marks between any major tick marks */
  style=0    /* suppress the axis line */
  offset=(0,7 PCT);
  /* OFFSET for the top of the axis prevents Count and Percent labels from
     overlapping the values listed below them for the top bar in the charts. */

axis2 label=none major=none minor=none style=0
  value=none /* suppress values along the axis */
  offset=(2 PCT, 2 PCT); /* create white space at each end of the axis */

proc gchart data=sashelp.class;

hbar age /
  discrete /* use the discrete values of AGE,
              not midpoints of default subranges                             */
  freq /* Frequency of the discrete values of AGE is the RESPONSE variable   */
       /* list the values for each AGE in a column to the right of the bars  */
  freqlabel='Count' /* custom label for Frequency column                     */
  cfreq  /* provide a Cumulative Frequency column  to the right of the bars  */
  cfreqlabel='Cum. Count' /* custom label for Cumulative Frequency column    */
  percent /* list the Percent of total frequency for each AGE in a column    */
  percentlabel='Percent' /* custom label for Percent column                  */
  cpercent  /* provide a Cumulative Percent column  to the right of the bars */
  cpercentlabel='Cum. Percent' /* custom label for Cumulative Percent column */
  maxis=axis1 /* AXIS1 statement defines the axis for the MIDPOINT variable  */
  raxis=axis2 /* AXIS2 statement defines the axis for the RESPONSE variable  */
  width=1.6   /* make bar width close to the height of the text              */
  space=1.6;  /* make space between bars equal to the bar width              */

run; quit;
```

Here is the code to create the Default Bar Chart:

```
/* three title statements */
proc gchart data=sashelp.class;
hbar age;
run; quit;
```

Improving a Trend Plot (Please see Reference 1 for the author's preferred alternatives.)

Figure 5: Default Plot

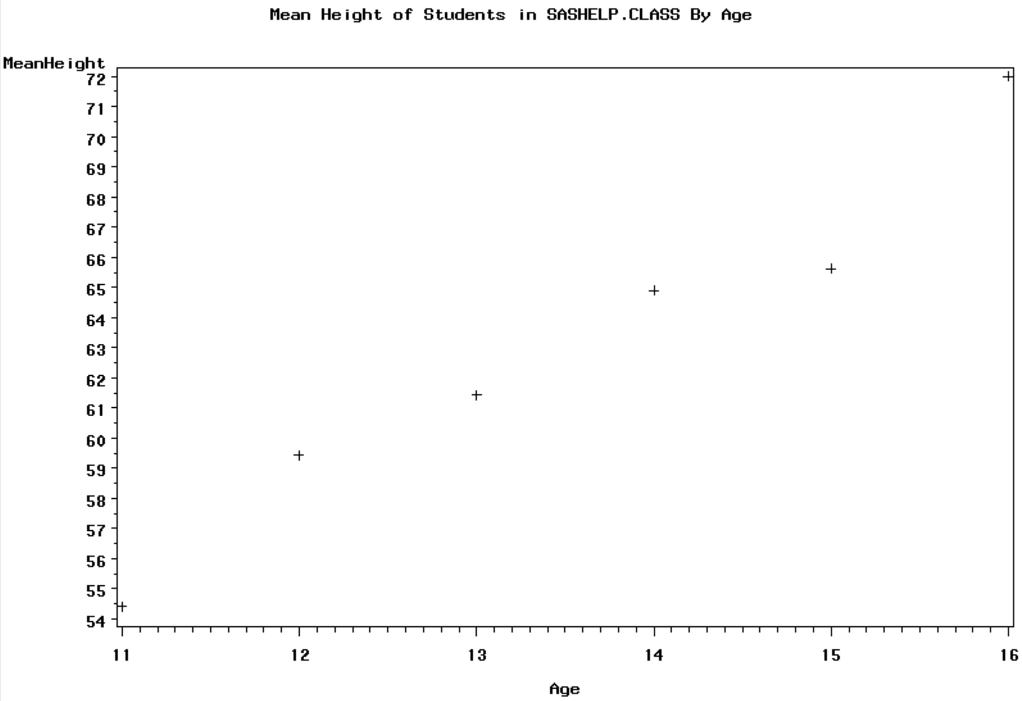
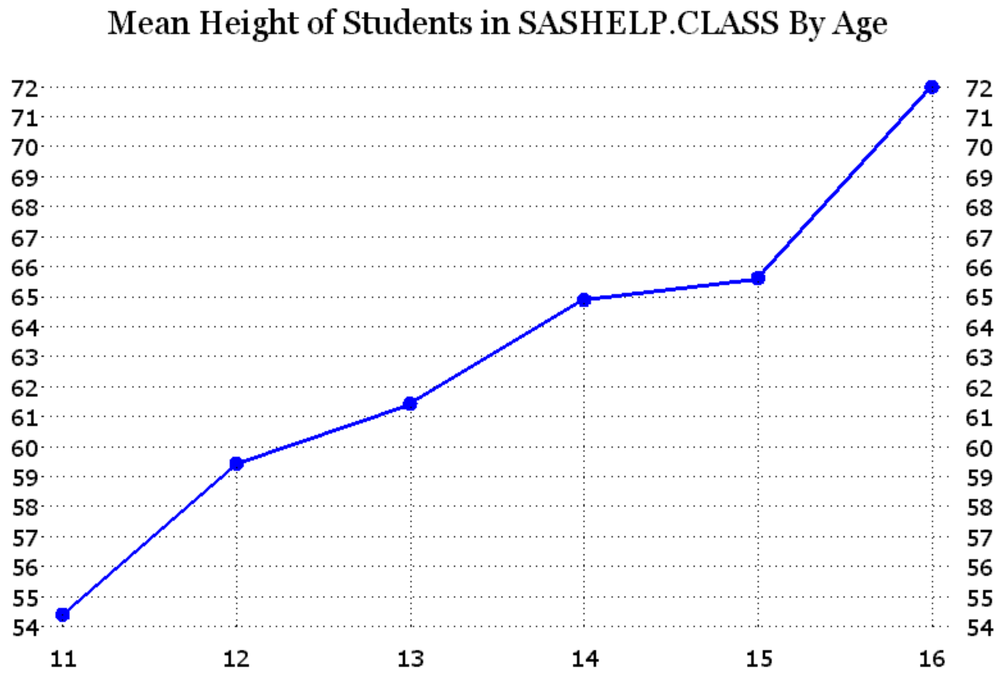


Figure 6: Custom Plot



In the graph above, with so few x-values, there really is no justification to provide the needle lines from the plot points to the tick-mark values. In a more realistic situation, however, they would help. To eliminate reliance on this very busy design in order to assure that the viewer can estimate the y-values, Reference 1 offers better solutions.

Here is the code used to create the Custom Plot:

```
proc means data=sashelp.class mean noprint nway;
class age;
var height;
output out=MeanHeightByAge mean=MeanHeight;
run;

/* include the same GOPTIONS statements
   and
   same TITLE1 & FOOTNOTE statements
   as for the Bar Chart */

title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Figure 6: ' color=CX0000FF 'Custom Plot'
  justify=LEFT height=2.5 PCT " ";

title3 font='Georgia' height=5 PCT color=CX000000
'Mean Height of Students in SASHELP.CLASS By Age';

symbol1
  value=dot /* other possibilities include POINT, NONE, etc. */
  height=1.5
  interpol=join /* connect the dots */
  line=1 /* solid line */
  width=2 /* thicken it */
  color=CX0000FF; /* browser-safe blue */

symbol2
  value=none /* suppress the marker */
  interpol=needle /* line from the axis to the marker */
  line=33 /* finest dotted line */
  color=CX000000; /* browser-safe black */

axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 offset=(2 PCT, 2 PCT);

/* For this data, there was no need to use the ORDER= and VALUE=
   axis tick mark value controls available for the AXIS statement. */

proc gplot data=MeanHeightByAge;

* PLOT  statement defines a plot using the Left-Hand-Side vertical axis *;
plot MeanHeight*age=1 /* use SYMBOL1 statement for this plot */
  /
  vaxis=axis1 /* AXIS1 statement defines the vertical   axis          */
  haxis=axis2 /* AXIS2 statement defines the horizontal axis          */
  autovref /* a reference line at every vertical axis tick mark value */
  lvref=33; /* linetype 33 for vertical axis reference lines          */
  /* (You can use cvref= to customize color of reference lines.)      */

* PLOT2 statement defines a plot using the Right-Hand-Side vertical axis *;
plot2 MeanHeight*age=2 /* use SYMBOL2 statement for this plot */
  / vaxis=axis1;

/* For the author’s preferred alternatives to this Custom Plot,
   please see Reference 1. */

run; quit;
```

Here is the code to create the Default Plot:

```
/* three title statements */
proc gplot data=MeanHeightByAge;
plot MeanHeight*age;
run; quit;
```

SAS/GRAPH Device Drivers

SASPRTC is the vendor’s current recommended driver for PDF graphs, replacing PDFC in that distinction. For gray-shade (monochrome black-and-white) graphs, SASPRTG (SASPRTM) is the equivalent of SASPRTC. I also had success with the GIF driver for PDF graphs.

PNG is the driver with which I had success for RTF graphs.

GIF is my recommended driver for HTML (web) graphs and for graphs that will be exported to Microsoft PowerPoint or Microsoft Word. I used the GIF driver for Figures 1-6. For me, the GIF driver has performed reliably. Except for photographs, GIF is the most widely used image file type on the web. All web browsers and PCs work with GIF.

GIF files are small. They will download quickly if staged for the web. The GIF driver can optionally produce transparent images, which means that the graph elements and the text are opaque, but web page background can show through. The related GIFANIM driver can be used to create animated graphs, which work as expected if imbedded in a web page or a PowerPoint slide (See Reference 3 for an example of web animation.)

I have used SAS/GRAPH since 1981. During more than a quarter of a century, I have had to work with many different types of output, display devices, printers, computer types (mainframes, mid-range / UNIX, and PCs), operating systems, and versions/releases of SAS/GRAPH. I have been forced to deal with numerous different device drivers, including even third-party (i.e., non-SAS-provided) device drivers. The GIF device driver has been the most dependable.

Among the drivers **not** shown in this paper’s examples are SASEMF, JPEG, Java, ActiveX, JAVA Image, and ActiveX Image, all of which are SAS Enterprise Guide Graph Results choices.

The SASEMF (and EMF) driver did not perform to my satisfaction. To explain why would be too long and complicated. You may have better luck, but I cannot recommend them.

The JPEG image format was developed by the Joint Photographic Experts Group. I do not know why there is a SAS/GRAPH JPEG driver. Graphs are not like continuous-tone photographs.

The Java & ActiveX drivers impose special requirements on the PC that will view their interactive output. If you have a business case to use these drivers, you must ensure that any PC used to display your graphs meets the requirements. How do you do that? Also, you will find that some features in SAS/GRAPH are not supported by the Java and ActiveX drivers. (Which features are unsupported is documented. For SAS 8, see Technical Notes TS-601 and TS-602 at support.sas.com. For SAS 9, see the SAS/GRAPH manual.) I prefer device drivers that support all SAS/GRAPH functions and features and whose output will display correctly on any PC.

However, if you look into the matter of Accessibility (discussed in a later section), you may conclude that ActiveX is an appropriate and important device driver, despite its requirements for the user’s PC/laptop and its graphic function limitations.

The Java Image and ActiveX Image drivers generate non-interactive PNG files with Java and ActiveX technology. Unfortunately, the font support in these drivers is not equal to that of the SAS/GRAPH PNG device driver. I can think of no reason to use them.

Creating Graphic Stream Files on Disk for Insert to Microsoft PowerPoint or Microsoft Word

To route a graphic image file directly to disk, rather than use a SAS/GRAPH Export function from the OUTPUT window, is very simple. (Figures 1-6 were created as Graphic Stream Files and then were inserted in this paper.) Before the graph creation code, use these statements:

```
ODS LISTING; /* Even though you may not be creating any tabular report (i.e., LISTING)
output, you must turn on the LISTING destination in order to create GSF output. */
ODS NORESULTS; /* Shut off the OUTPUT window. */
GOPTIONS GSFNAME=AnyName GSFMODE=REPLACE;
FILENAME AnyName "PathToTheFile\AnyFileName.FileExt";
```

FileExt is the three-character file extension appropriate to the graphic device driver that you are going to use (e.g., for DEVICE=GIF, use GIF as the file extension).

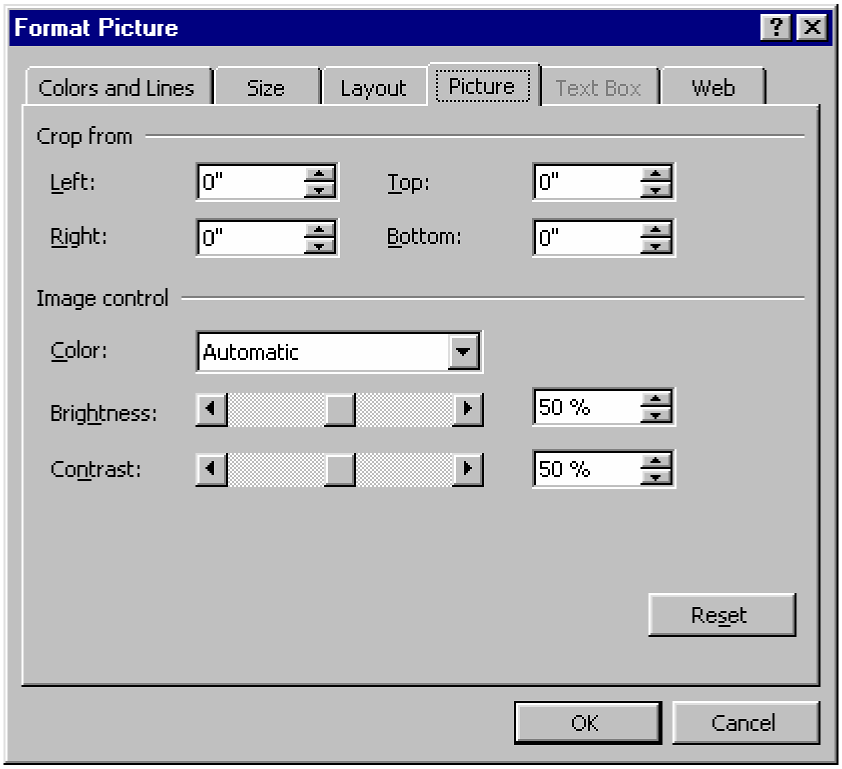
The PathToTheFile could be: `C:\AnyFolderName`

for which case the SAS log would contain a message like:

```
NOTE: 65 RECORDS WRITTEN TO C:\AnyFolderName\AnyFileName.GIF
```

Working with SAS/GRAPH Output in Microsoft Word

First, place your cursor at the desired location in the document. To import SAS/GRAPH output, use Insert > Picture > From File. When you have the file in your Word document (where it will have been automatically shrunk to fit (but preserving its aspect ratio) within the margins) you can right-click on the image to bring up this panel:



Format Picture

Crop from

Left: 0"
Top: 0"
Right: 0"
Bottom: 0"

Image control

Color: Automatic
Brightness: 50 %
Contrast: 50 %

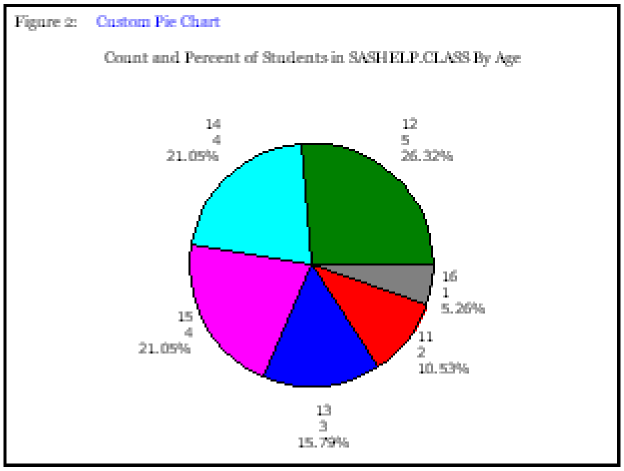
The Picture tab comes up in front, and you could use it to crop (trim the edges of) the image. The Layout tab is used if you wish to wrap text around an image that does not spread to the margins.

You can use the Size tab to shrink the image, presumably with “Lock aspect ratio” checked. An image may still be readable after half-sizing. At 3.2 inches, an available page display width of 6.5 inches will leave room to insert an inter-image gap with the keyboard Space bar between a pair of side-by-side images as done in this paper below.

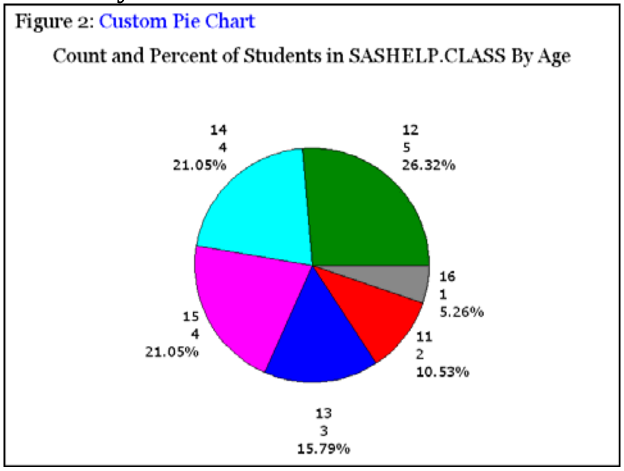
After right-clicking on the image, you can select Borders and Shading, and use the Borders tab to apply a Box around the image. Adding this extra border to an image with a SAS/GRAPH border has, I think, a pleasing visual effect and has been applied to all figures in this paper.

This paper (before conversion to PDF) was created with Word, with available page width of 6.5 inches. Below are seven GIF images. Four of them compare images manually reduced in Word with the same images created at a little less than half of the page width (3.2 inches) by specifying “GOPTIONS XPIXELS=307 YPIXELS=230”. The next two compare Georgia and Verdana fonts after manual image reduction. The last image is simply auto-fit by Word to page width.

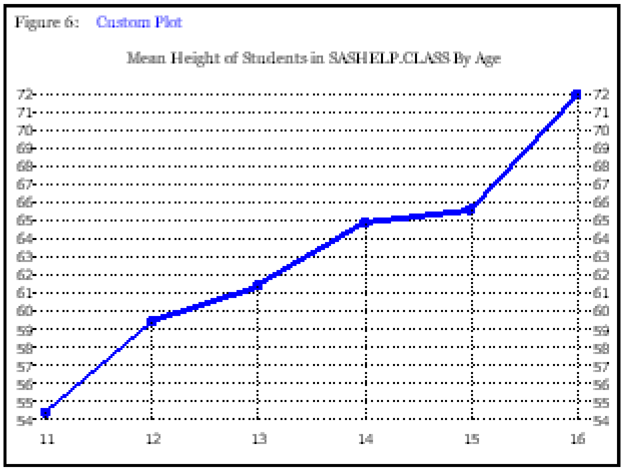
Created at 3.2 inches wide.



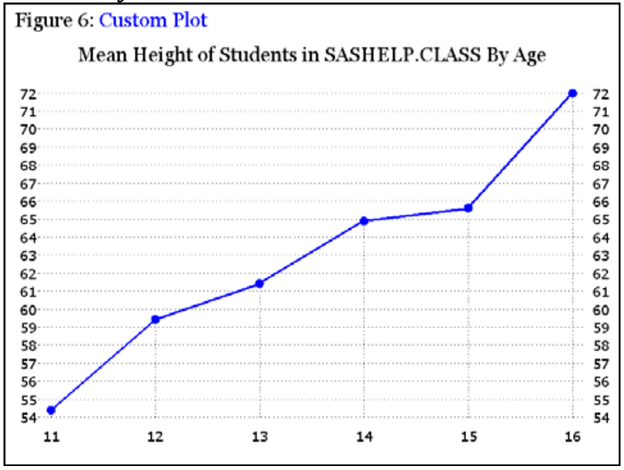
Manually resized to 3.2 inches wide.



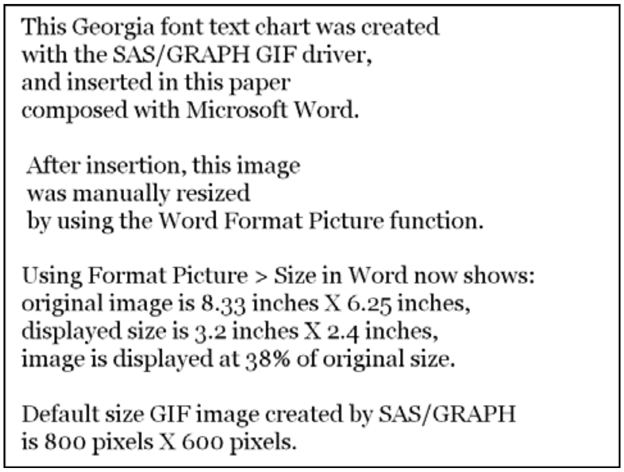
Created at 3.2 inches wide.



Manually resized to 3.2 inches wide.



Above, manually resized graphic elements look better than those created at half-page width. The figures below focus on the effect of resizing message-critical content: text in a graphic image.

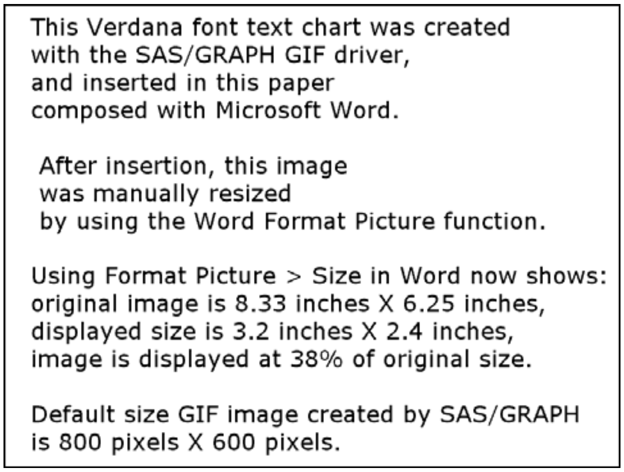


This Georgia font text chart was created
with the SAS/GRAPH GIF driver,
and inserted in this paper
composed with Microsoft Word.

After insertion, this image
was manually resized
by using the Word Format Picture function.

Using Format Picture > Size in Word now shows:
original image is 8.33 inches X 6.25 inches,
displayed size is 3.2 inches X 2.4 inches,
image is displayed at 38% of original size.

Default size GIF image created by SAS/GRAPH
is 800 pixels X 600 pixels.

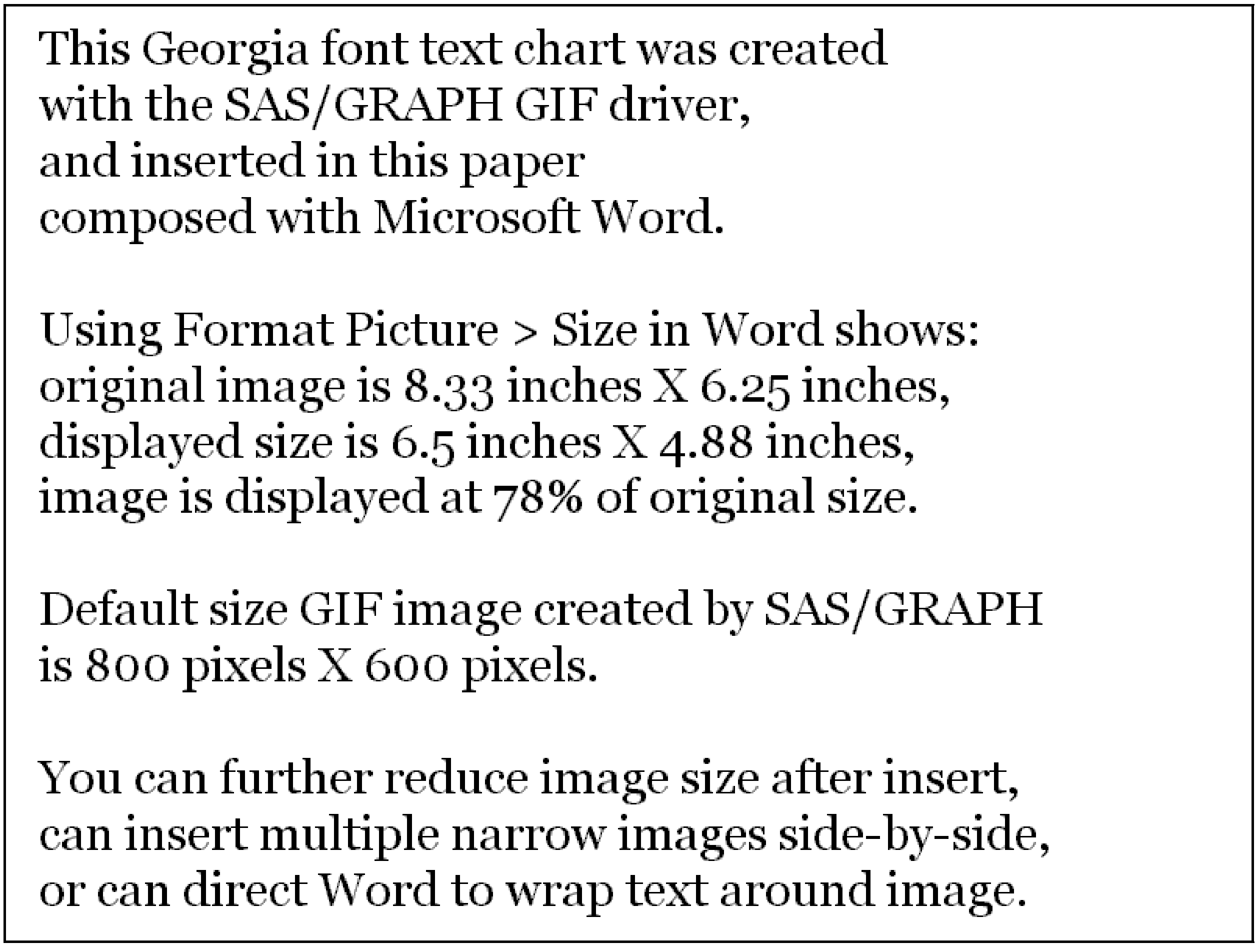


This Verdana font text chart was created
with the SAS/GRAPH GIF driver,
and inserted in this paper
composed with Microsoft Word.

After insertion, this image
was manually resized
by using the Word Format Picture function.

Using Format Picture > Size in Word now shows:
original image is 8.33 inches X 6.25 inches,
displayed size is 3.2 inches X 2.4 inches,
image is displayed at 38% of original size.

Default size GIF image created by SAS/GRAPH
is 800 pixels X 600 pixels.



This Georgia font text chart was created
with the SAS/GRAPH GIF driver,
and inserted in this paper
composed with Microsoft Word.

Using Format Picture > Size in Word shows:
original image is 8.33 inches X 6.25 inches,
displayed size is 6.5 inches X 4.88 inches,
image is displayed at 78% of original size.

Default size GIF image created by SAS/GRAPH
is 800 pixels X 600 pixels.

You can further reduce image size after insert,
can insert multiple narrow images side-by-side,
or can direct Word to wrap text around image.

My Conclusions:

1. It is safer to create a GIF image at default size with the SAS/GRAPH GIF driver and resize it inside Microsoft Word than to right-size the original with GOPTIONS modifications at driver invocation. Since the image insertion in a Word document is a manual operation, taking a very few more manual steps to resize the image is simpler and more reliable than going through extra coding rigmarole to TRY to get an acceptable half-page-width original image file.

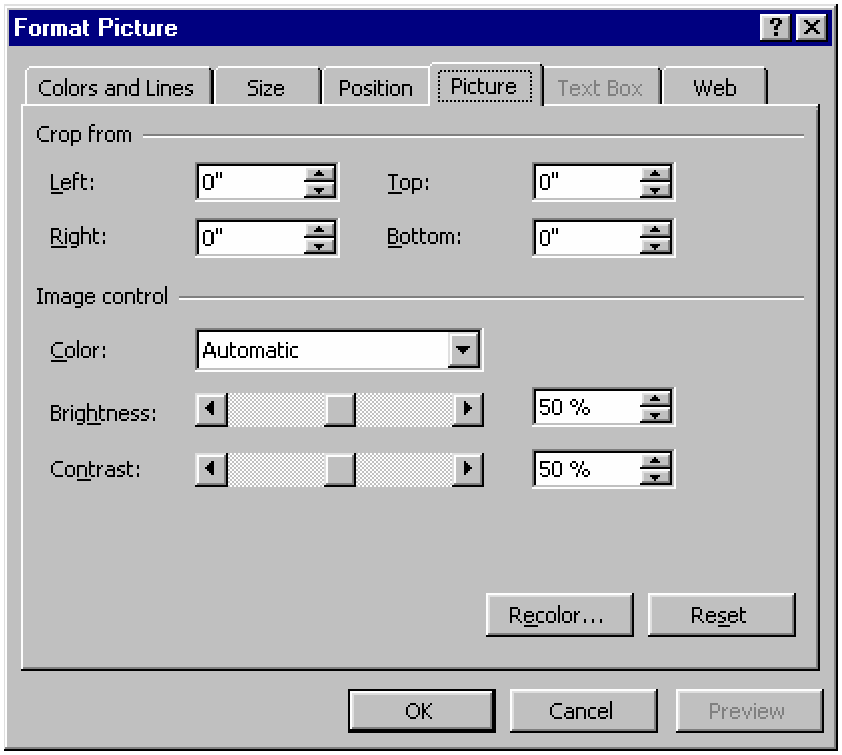
2. Just as Verdana is more readable than Georgia for small text parts of a graph, so, too, Verdana is more readable than Georgia for a graph that is manually resized after insertion in Word.

Working with SAS/GRAPH Output in Microsoft PowerPoint

First, open either a new or existing slide presentation and create an empty slide. To import SAS/GRAPH output, use Insert > Picture > From File. As slide layouts for displaying an image, I have found the “Blank” slide and the “Title Only” slide to be the easiest ones to work with. “Title Only” does permit you to insert an image.

NOTE: It is possible to insert multiple images on a single slide. You simply need to resize and reposition them as desired. Readability of the result depends on design of the original images.

When you have the image in your slide, it will have been automatically shrunk (preserving its aspect ratio) to fit within the PowerPoint default slide margins. (I do not know whether you can permanently change those default margins.) Then you can right-click on the image to bring up this panel:



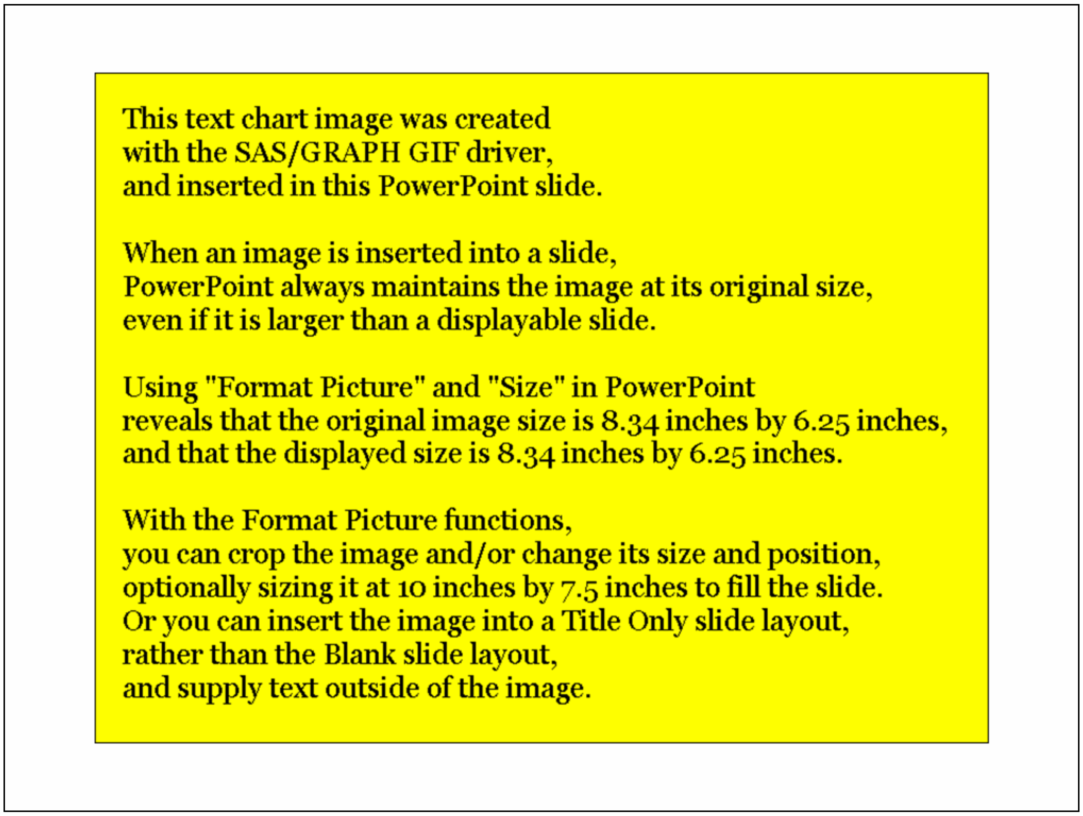
Left: 0"
Top: 0"
Right: 0"
Bottom: 0"
Color: Automatic
Brightness: 50 %
Contrast: 50 %

The Picture tab comes up in front, and you could use it to crop the image. The Position tab can be used to relocate the image after resizing. Unless I am using a title that I type into PowerPoint, I usually use the Size tab to expand the image to fill the slide, with “Lock aspect ratio” checked. So far, I have never found image quality to degrade after expansion to fill the slide.

The full size of the display space for PowerPoint slides is 10 inches wide by 7.5 inches high. These dimensions respect the 4-to-3 ratio of traditional, non-wide-screen computer monitors, and of projection screens. **SAS/GRAPH GIF image files, with their 4-to-3 aspect ratio, are “PowerPoint-Ready”.**

Below are screen prints from PowerPoint slide displays, whose imbedded images were created with PROC GSLIDE. The same dimension effects apply to all SAS/GRAPH procedure output.

Screen print of a PowerPoint slide without resizing the inserted image:



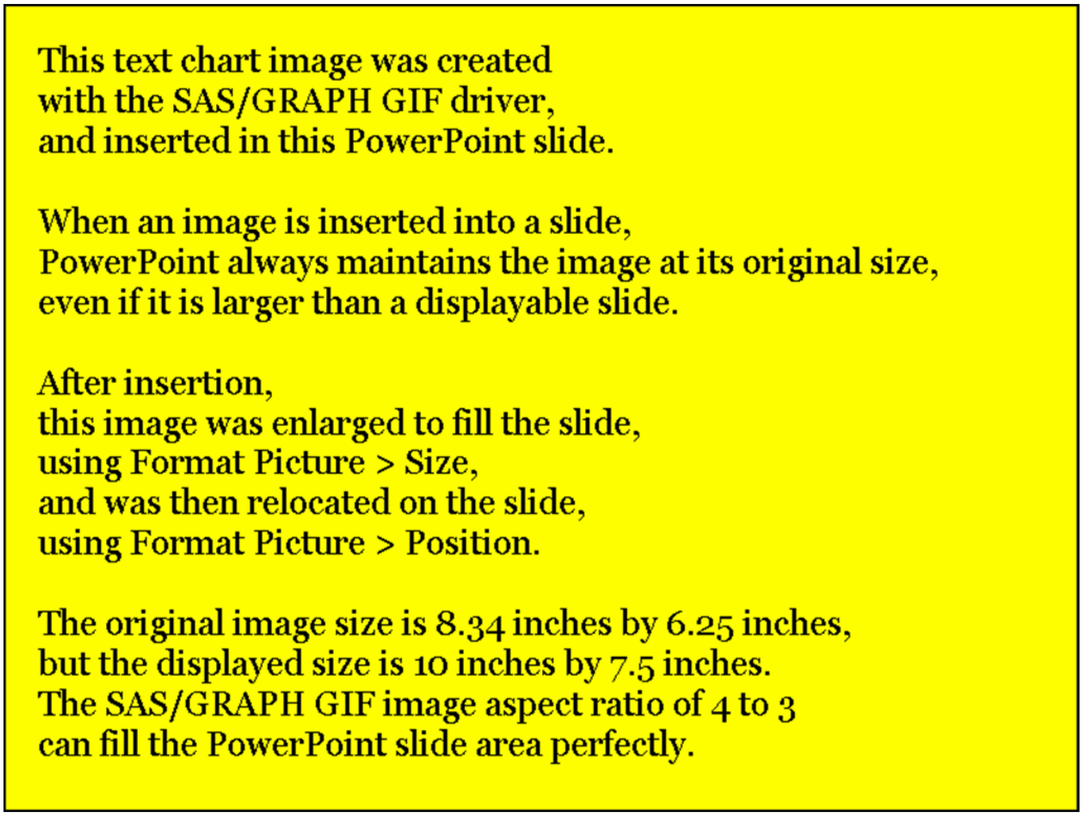
This text chart image was created
with the SAS/GRAPH GIF driver,
and inserted in this PowerPoint slide.

When an image is inserted into a slide,
PowerPoint always maintains the image at its original size,
even if it is larger than a displayable slide.

Using "Format Picture" and "Size" in PowerPoint
reveals that the original image size is 8.34 inches by 6.25 inches,
and that the displayed size is 8.34 inches by 6.25 inches.

With the Format Picture functions,
you can crop the image and/or change its size and position,
optionally sizing it at 10 inches by 7.5 inches to fill the slide.
Or you can insert the image into a Title Only slide layout,
rather than the Blank slide layout,
and supply text outside of the image.

Screen print of a PowerPoint slide with image manually resized to fill the slide, with no loss of text quality:



This text chart image was created
with the SAS/GRAPH GIF driver,
and inserted in this PowerPoint slide.

When an image is inserted into a slide,
PowerPoint always maintains the image at its original size,
even if it is larger than a displayable slide.

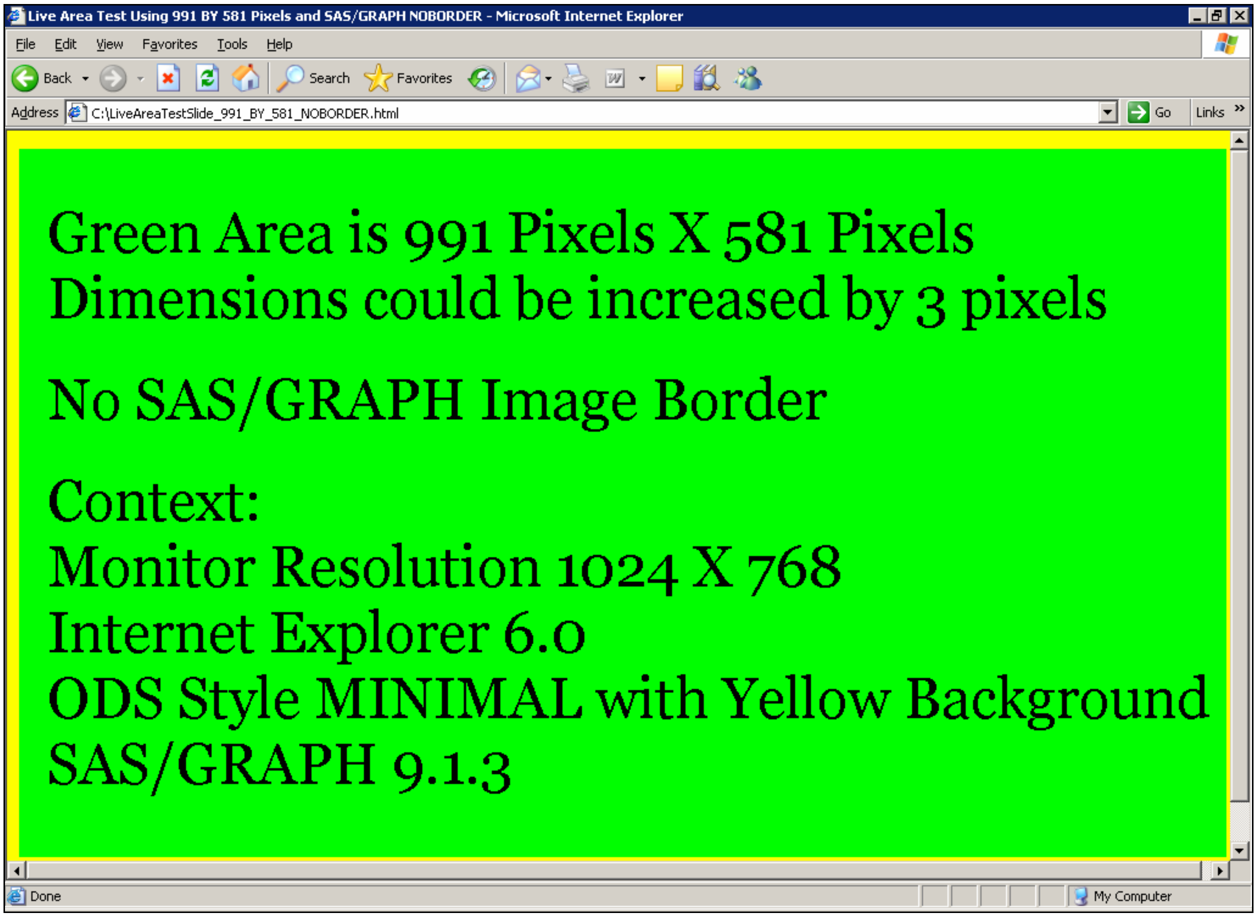
After insertion,
this image was enlarged to fill the slide,
using Format Picture > Size,
and was then relocated on the slide,
using Format Picture > Position.

The original image size is 8.34 inches by 6.25 inches,
but the displayed size is 10 inches by 7.5 inches.
The SAS/GRAPH GIF image aspect ratio of 4 to 3
can fill the PowerPoint slide area perfectly.

ODS Destination HTML:
Optimizing your use of web page space (a.k.a. “Live Area” or “Live Space”)

Web page users do not really like to scroll. In almost all cases, a web graph should be sized so that it can be seen in full without scrolling. Graphs that are needlessly designed to require scrolling are irritating, prevent getting what could have been a coherent “full visual impression”, and are apt to be unprintable as a convenient unit. At the same time, to maximize readability, graphs should be designed as large as will fit in the space available.

Since Reference 1 covers web design and the ODS HTML destination more broadly, here my focus is on sizing a graph to make maximum use of the space inside the web browser frame and below any optional toolbar(s) that the web user might have enabled. The SAS/GRAPH GSLIDE output in the web page below is deliberately under-sized only so that you can see the right-hand-side and bottom edges of the image where the yellow web page background is peeking out.


Live Area Test Using 991 BY 581 Pixels and SAS/GRAPH NOBORDER - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
Go
Links
Done
My Computer

Address: C:\LiveAreaTestSlide_991_BY_581_NOBORDER.html

Green Area is 991 Pixels X 581 Pixels
Dimensions could be increased by 3 pixels

No SAS/GRAPH Image Border

Context:
Monitor Resolution 1024 X 768
Internet Explorer 6.0
ODS Style MINIMAL with Yellow Background
SAS/GRAPH 9.1.3

“Live Area” or “Live Space” is the area available inside the browser window to display web page content. The frame used by Windows itself and the frame used by a web browser reduce the usable display space of a CRT monitor or a flat panel display.

The size of the live area is affected by the screen resolution of the CRT monitor or flat panel. The commonest resolution in use today is 1024 pixels (width) by 768 pixels (height), with an aspect ratio of 4 X 3.

You can determine the resolution of your own screen on a Windows machine by clicking Start > Settings > Control Panel > Display Properties > Settings. The Screen Area slider shows your current setting, which you can change.

Other factors that affect the live space are the choice of web browser, the version of the web browser, and the absence or presence of optional toolbars, such as those for Google or Yahoo.

Within that context, the available space for your image is affected by the choice of ODS style (e.g., Minimal vs. Default) and the version of SAS. Also, SAS 8 and SAS 9 differ in the version of HTML language used as well as in how that HTML language is used. (For SAS 8.2, which creates a larger top margin above the image in the live area, the example above would be limited to pixel dimensions of 991 by 551.)

Similar to the size of the available live area, the size of a GIF graphic image is measured in pixels. Its size is controlled with GOPTIONS XPIXELS and YPIXELS.

Here is the code used to create the test web page above:

```
proc template;
edit Styles.Minimal AS Styles.Minimal_CXFFFF00;
  style body / background = CXFFFF00; /* browser-safe yellow */
end;
run;

ODS NORESULTS;
ODS LISTING CLOSE;

ODS HTML PATH="C:\" (URL = NONE)
         FILE="LiveAreaTestSlide_991_BY_581_NOBORDER.html"
         (TITLE=" Live Area Test Using 991 BY 581 Pixels and SAS/GRAPH NOBORDER")
         STYLE=Styles.Minimal_CXFFFF00
         GTITLE GFOOTNOTE;

goptions reset=all;
goptions device=GIF;
goptions noborder;
goptions cback=CX00FF00; /* browser-safe green */
goptions XPIXELS=991 YPIXELS=581;

proc gslide;
title1 . . . ;
run; quit;

ODS HTML CLOSE;
ODS LISTING;
```

You should design your web pages for what you expect to be the typical worst-case situation of your intended web audience. During development, you should configure your own screen resolution and web browser to match those of your target.

NOTE: Although the example above shows how to fill the live space almost completely, it is prudent to instead leave a reasonable amount of extra space unused if the resulting smaller graph is still readable. Then web users with less live space than your expected target configuration may be able to see the whole picture without scrolling.

ODS Destination PDF

The key points shown are: (a) driver selection; (b) ODS packaging; (c) margin controls; (d) how to turn off the Table of Contents; and (e) compression. For PDF you are limited to three fonts: Times, Helvetica, and Courier, plus their bold, italic, and bold italic versions, coded, e.g., as 'Times/Bold/Italic'. SAS software fonts are also available but not recommended by me.

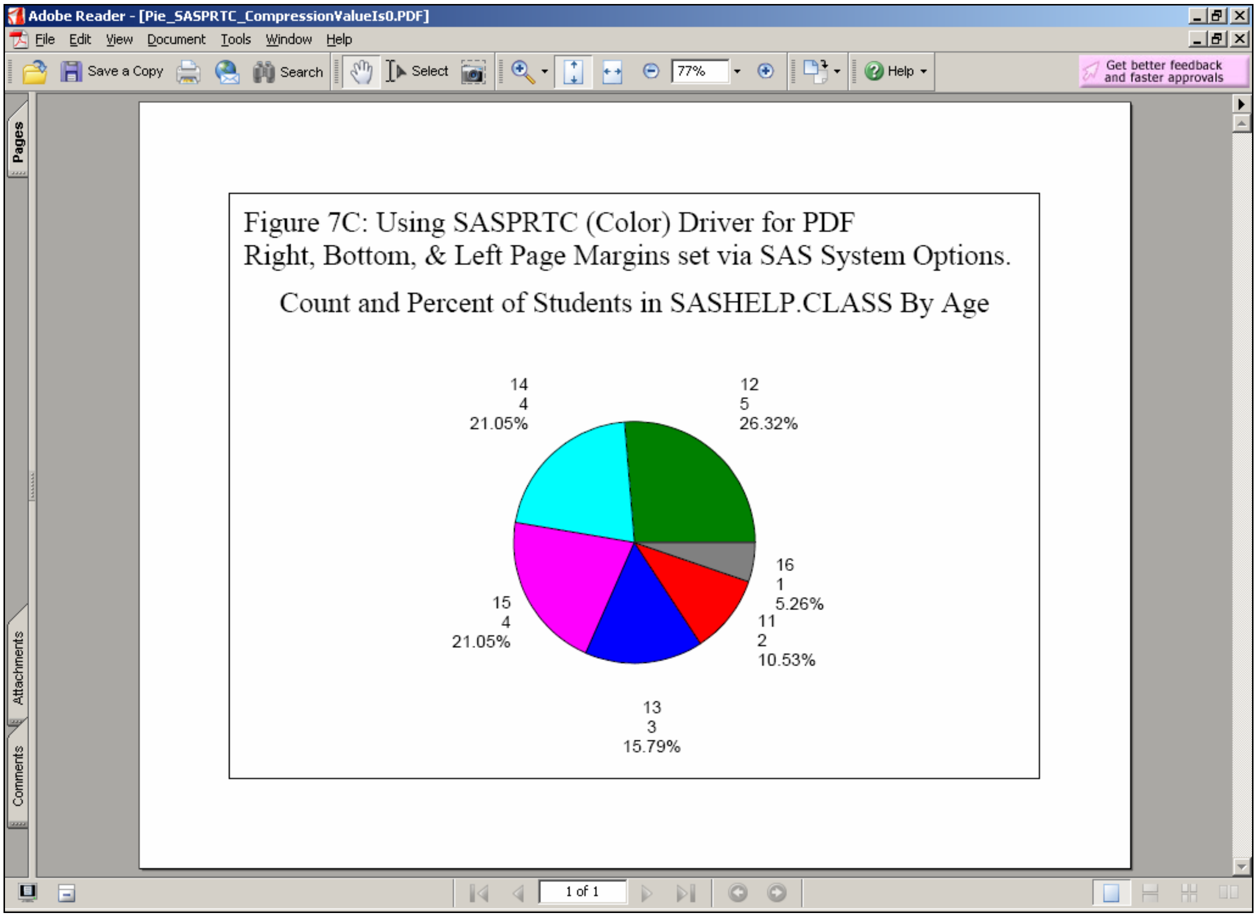

Adobe Reader - [Pie_SASPRTC_CompressionValueIs0.PDF]
File Edit View Document Tools Window Help
Save a Copy
Search
Select
77%
Help
Get better feedback and faster approvals
Pages
Attachments
Comments
1 of 1

Figure 7C: Using SASPRTC (Color) Driver for PDF
Right, Bottom, & Left Page Margins set via SAS System Options.

Count and Percent of Students in SASHELP.CLASS By Age

14
4
21.05%
12
5
26.32%
16
1
5.26%
15
4
21.05%
11
2
10.53%
13
3
15.79%

```
OPTIONS ORIENTATION=LANDSCAPE;
OPTIONS RIGHTMARGIN=1 IN BOTTOMMARGIN=1 IN LEFTMARGIN=1 IN; /* Top Margin automatic */
ODS NORESULTS;
ODS LISTING CLOSE;
ODS PDF FILE="C:\AnyFolder\AnyFileName.PDF" NOTOC /* turn off Table of Contents */
  TITLE="Custom Pie Chart Using SASPRTC Driver for PDF" AUTHOR="LeRoy Bessler PhD"
  SUBJECT="Demonstrate PDF Pie Chart with SASPRTC Driver"
  KEYWORDS="PDF SAS/GRAPH ODS 'SASPRTC Device Driver'"
  COMPRESS=0; /* No Compression. Otherwise, assign value in the range 1-9. I have
              observed little or no improved compression for values beyond 1 or 2. */
  /* accepting default ODS style Styles.Printer */
goptions reset=all;
goptions device=SASPRTC;
goptions border;
goptions ftext='Helvetica' htext=3 PCT;
title1 font='Times' height=5 PCT color=CX000000 justify=LEFT "  Figure 7C: Using
SASPRTC (Color) Driver for PDF" justify=LEFT height=2.5 PCT " ";
title2 font='Times' height=5 PCT
  'Count and Percent of Students in SASHELP.CLASS By Age';
proc gchart data=sashelp.class;
pie age / name="SASPRTC0" discrete noheading descending percent=outside;
run; quit;
ODS PDF CLOSE;
```

Samples Using the UnColor PDF-compatible Graphic Device Drivers

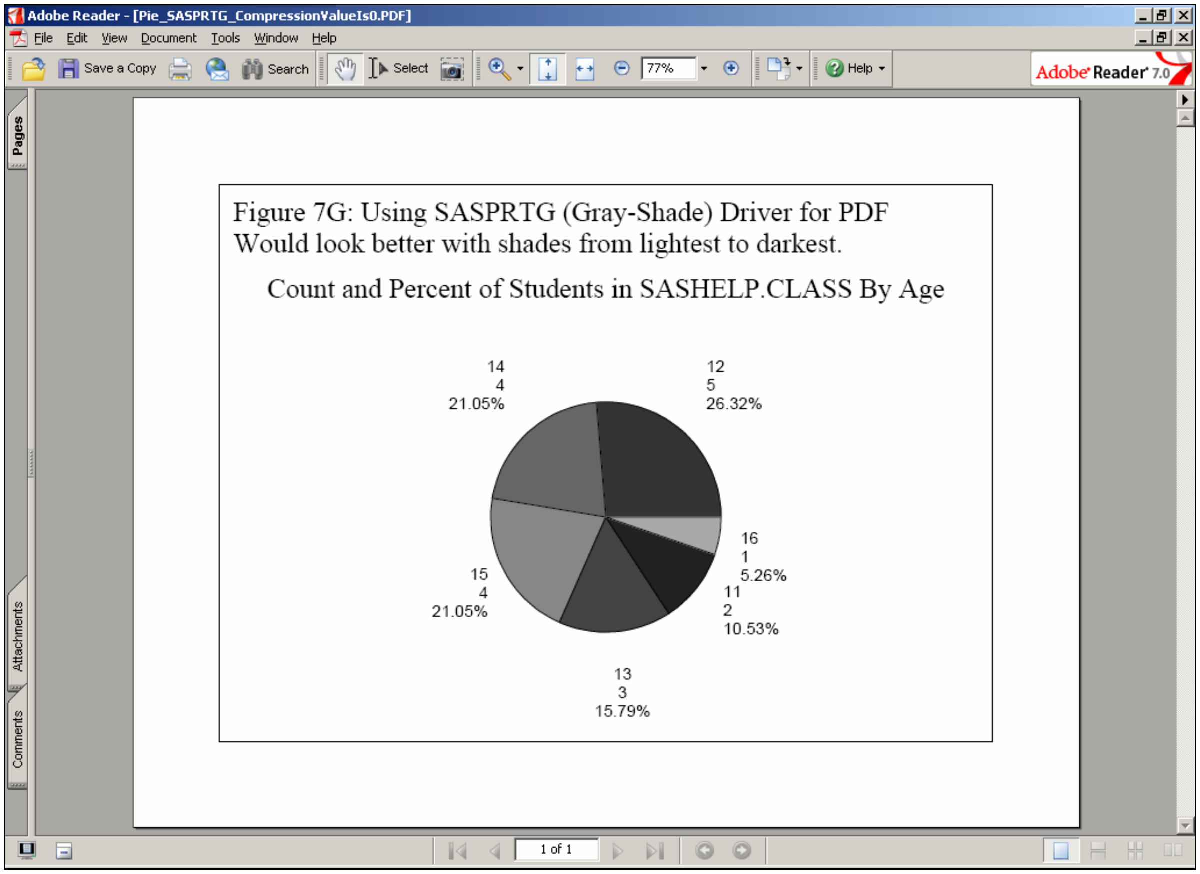

Adobe Reader - [Pie_SASPRTG_CompressionValueIs0.PDF]
File Edit View Document Tools Window Help
Save a Copy
Search
Select
77%
Help
Adobe Reader 7.0
Pages
Attachments
Comments
1 of 1

Figure 7G: Using SASPRTG (Gray-Shade) Driver for PDF
Would look better with shades from lightest to darkest.

Count and Percent of Students in SASHELP.CLASS By Age

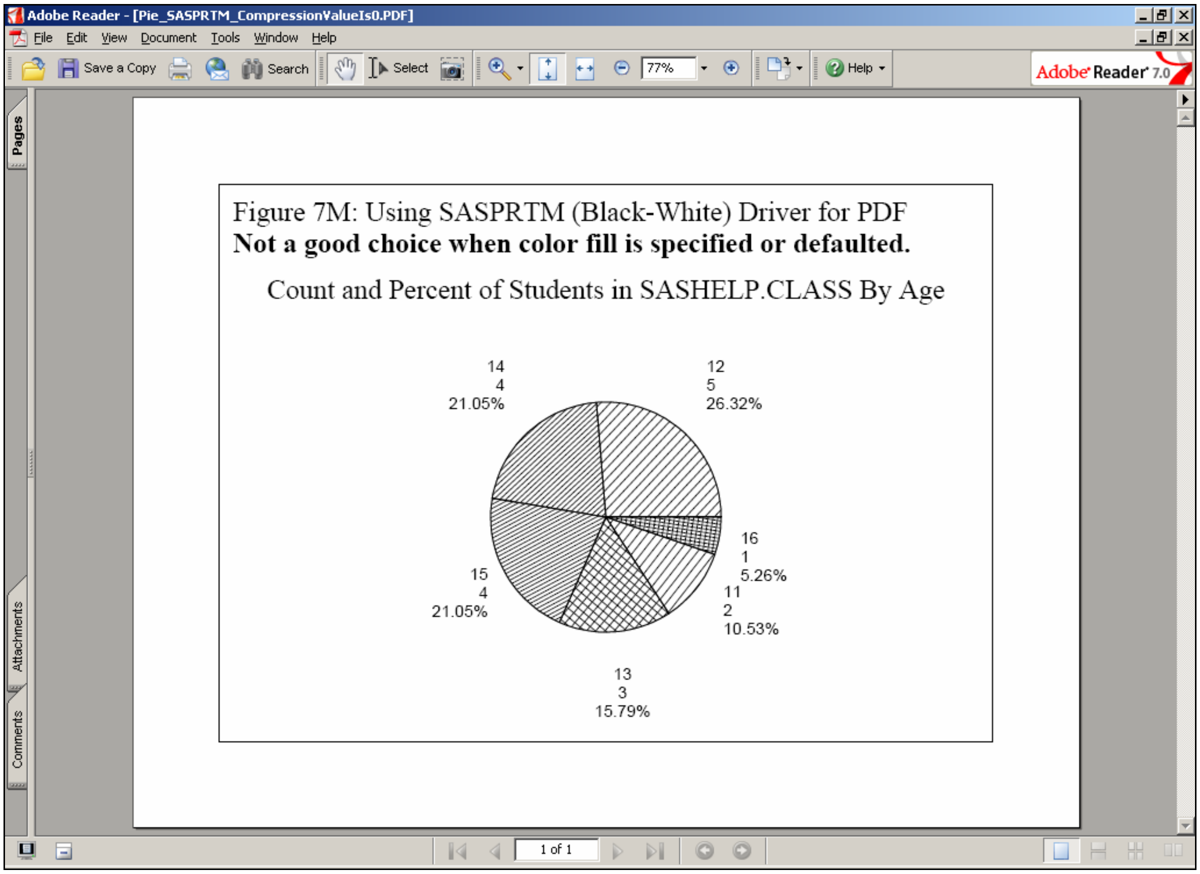

Adobe Reader - [Pie_SASPRTM_CompressionValueIs0.PDF]
File Edit View Document Tools Window Help
Save a Copy
Search
Select
77%
Help
Adobe Reader 7.0
Pages
Attachments
Comments
1 of 1

Figure 7M: Using SASPRTM (Black-White) Driver for PDF
Not a good choice when color fill is specified or defaulted.

Count and Percent of Students in SASHELP.CLASS By Age

ODS Destination RTF

The key points shown are: (a) driver selection; (b) ODS packaging; and (c) margin controls;

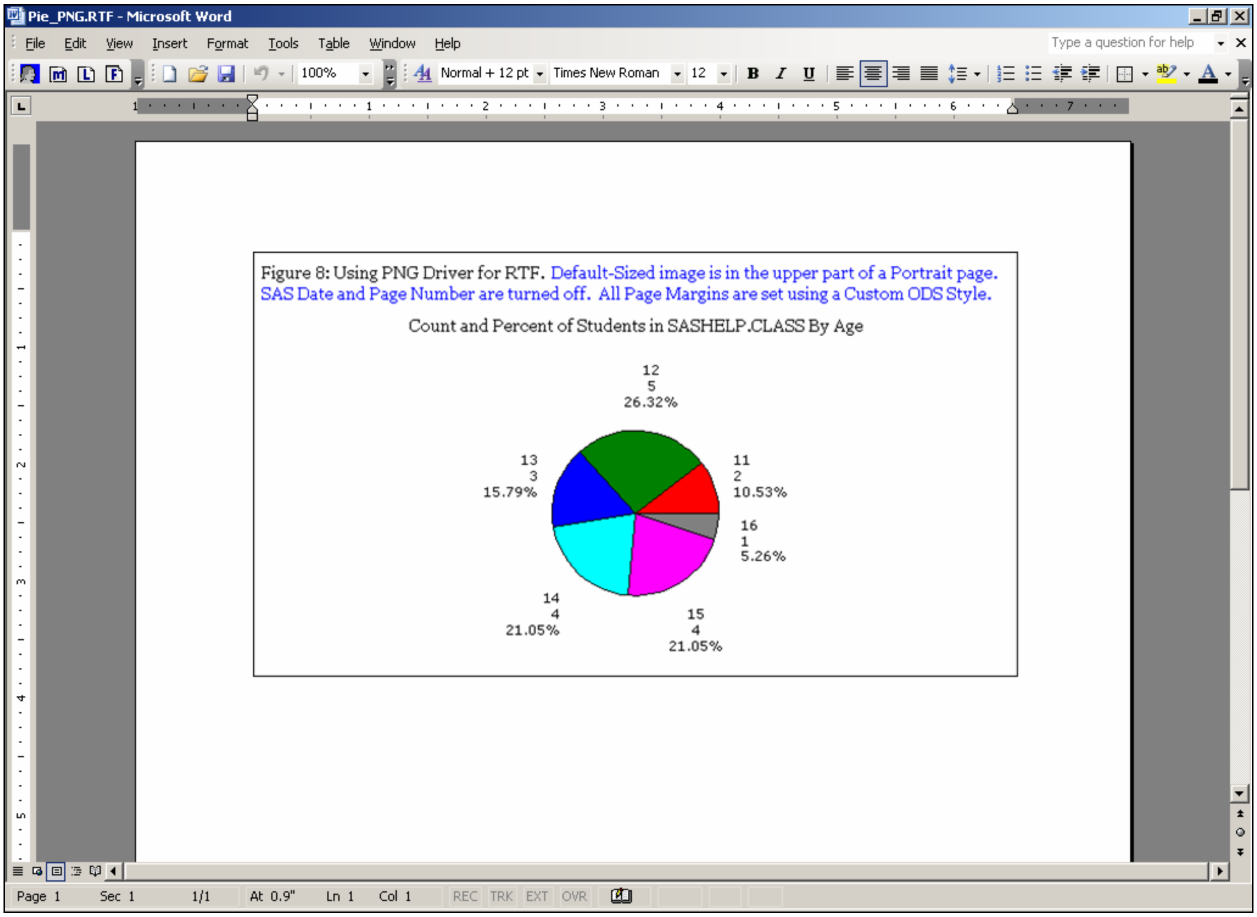


Figure 8: Using PNG Driver for RTF. Default-Sized image is in the upper part of a Portrait page. SAS Date and Page Number are turned off. All Page Margins are set using a Custom ODS Style.

Pie_PNG.RTF - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal + 12 pt
Times New Roman
12
Count and Percent of Students in SASHELP.CLASS By Age
12
5
26.32%
11
2
10.53%
16
1
5.26%
15
4
21.05%
14
4
21.05%
13
3
15.79%
Page 1 Sec 1 1/1 At 0.9" Ln 1 Col 1 REC TRK EXT OVR

For the image in the RTF page above, the right, bottom, and left margins are set to 1 inch, but the top margin is set to only 0.75 inch because 0.25 is already reserved for the SAS date and page number displays that have been turned off with OPTIONS NODATE NONUMBER.

The image is created at the default size by the PNG driver, for which HSIZE=6.474 INCHES and VSIZE=3.631 INCHES. This is a fortuitously convenient width for imbedding in a Portrait document with a USA Letter Size page width of 8.5 inches with standard 1-inch margins. But the default height is rather skimpy and makes poor use of the available space.

When you open an RTF file with Microsoft Word, you can right-click on the image and use the Size tab of Format Picture function to resize the image. However, you cannot make the image longer to fill the page without changing the aspect ratio, in which case the text (and any circle) in the graph will be distorted.

For a should-be distortion-proof programmatic solution to produce a PNG image with a larger size, higher resolution, and thicker lines, please see Technical Note TS-674 at support.sas.com, for its discussion of macro %PNGSZ.

Here is the code used to create the RTF document with imbedded PNG image:

```
proc template; /* Create a Custom Style based on the default style designed for RTF */
define style styles.RTFwithOneInchMargins;
parent=styles.rtf;
style body from document /
  topmargin=0.75 in
  rightmargin=1 in
  bottommargin=1 in
  leftmargin=1 in;
end;
run;

OPTIONS ORIENTATION=PORTRAIT;
OPTIONS NODATE NONUMBER;

ODS NORESULTS;
ODS LISTING CLOSE;

ODS RTF
  FILE="C:\AnyFolder\AnyFileName.RTF"
  TITLE="Custom Pie Chart Using PNG Driver for RTF"
  AUTHOR="LeRoy Bessler PhD"
  GTITLE GFOOTNOTE
  STYLE=Styles.RTFwithOneInchMargins;

goptions reset=all;
goptions device=PNG;
goptions border;
goptions ftext='Verdana' htext=3.75 PCT;

title1 height=2.5 PCT ' ';
title2 font='Georgia' height=5 PCT color=CX000000 justify=LEFT
  "  Figure 8: Using PNG Driver for RTF.  "
color=CX0000FF "Default-Sized image is in the upper part of a Portrait page."
justify=LEFT
  "  SAS Date and Page Number are turned off.  All Page Margins are set using a Custom
ODS Style."
justify=LEFT height=2.5 PCT " ";
title3 font='Georgia' height=5 PCT
  'Count and Percent of Students in SASHELP.CLASS By Age';

proc gchart data=sashelp.class;
pie age / discrete noheading percent=outside name="Pie_PNG";
run; quit;

ODS RTF CLOSE;
```

Accessibility

Accessible output on the web is communication-effective for people with impairments—i.e., it is designed and constructed to try to reach all of the possible audience. For more expansive information on Accessibility than I can provide here, please consult Lisa.Pappas@sas.com, do a search at support.sas.com, and do a web-wide search with, say, Google.

To achieve web accessibility of SAS/GRAPH output, one of the solutions, the ActiveX graphic device driver, requires the presence of special software on the web viewer's PC. So, even if a user has no impairments, that special software is required in order to display ActiveX output. Thus, to meet the possible needs of some of your audience with ActiveX, a definite extra software requirement is imposed on all of your audience. On three different PCs, I had three different experiences with ActiveX output, and only one of the experiences was satisfactory.

Nevertheless, there is good news:

(1) There are things that you can do with respect to fonts, colors, and backgrounds to promote accessibility without ActiveX, and most of them also benefit people with no visual impairment.

(2) Some of the ActiveX automatic deliverables can instead be created by explicit coding. These opportunities include provision of ALT text and use of the DES= (DESCRIPTION=) parameter on the PIE statement, BAR statement, PLOT statement, etc. The benefit of coding these yourself is that you can control and enhance the content rather than take the default. See Reference 1 for details and examples of those design principles and implementation methods.

(3) If you wish to use ActiveX with certainty that you can reach all of your web audience (i.e., users with or without ActiveX support on their PCs), you could build your web-delivered application with a home page that allows the web user to select an ActiveX branch or a GIF branch for viewing the rest of the web package. It would require creating two sets of graphs and two sets of web pages, but that can be worth the effort.

This tutorial made much use of the GIF driver. Other device drivers may have their admirers, but, except for the JPG (aka JPEG) image format, which is really appropriate only for photos, the GIF image is the most widely used format in the world and on the web. The probability of meeting a PC that cannot handle GIF is probably zero. If a programmer can still find a way to meet the needs of impaired users, then GIF qualifies as the most accessible web graphic format.

If your definition of "Accessibility" includes presenting data and information via the web or PDF in ways other than the usual, please see Reference 3, my paper on Multi-Media Wizardry with ODS. You can add more channels and features to information delivery and maybe even have some fun.

Accessibility of Fonts and Colors

For people with normal or near-normal vision, a designer's or programmer's use of fonts, colors, and backgrounds can limit the communication-effectiveness of data presentation. Please see Reference 1 for guidelines. For an extensive discussion of communication-effective color, please see Reference 4.

Conclusion

Here is a summary of my device driver findings.

- For PDF, I agree with the SAS recommendation of drivers SASPRTC and SASPRTG for color and gray shades. (Formerly, SAS recommended PDFC and PDF, respectively.) I also found the GIF driver to work for PDF, but, not having done exhaustive testing, I can not recommend it over SASPRTC and SASPRTG.

- For RTF, I got acceptable results only with PNG (but not PNG generated with the Java Image driver or the ActiveX Image driver).

- For everything else within the scope of this tutorial, the GIF driver performed either better than, or as well as, any other driver.

Web output can easily be made accessible, benefiting all users as well as any impaired.

Beware of the SIMULATE font and know how to recognize it (Reference 1 explains how to create a sample) when SAS/GRAPH substitutes it for your requested font, possibly without any notification in the SAS log. Readability is essential to communication. Elegance enhances the impact made by communication. The SIMULATE font can be hard to read and is never elegant.

References (Related Work by this Author)

1. Chart Smart: Design and Build SAS Graphs That Inform and Influence, elsewhere in these MWSUG 2007 Conference Proceedings.

2. Communication-Effective Pie Charts, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

3. Multi-Media Wizardry: How To Make ODS Outputs That “Dance and Sing”, elsewhere in these MWSUG 2007 Conference Proceedings.

4. Communication-Effective Use of Color for Web Pages, Graphs, Tables, Maps, Text, and Print, *Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference*, 2004. Cary, NC: SAS Institute Inc.

Acknowledgments

I am grateful to Alix Riley for her helpful review of this paper, and to John Xu and Delayne Stokke, MWSUG 2007 Co-chairs, for the opportunity to share my ideas with other SAS users.

Contact Information

Your comments, questions, and suggestions are welcome.

LeRoy Bessler PhD
Email: bessler@execpc.com
Phone: 1 414 351 6748 (evenings and weekends—time is six hours earlier than Greenwich Mean Time)

Appendix 1: Graphs for the Small Screen.

SAS® Graphs for a BlackBerry, iPhone, or Other Small Email Screen: Extreme SAS/GRAPH® and the Necessity and Power of Simplicity

LeRoy Bessler, Assurant Health, Milwaukee, USA, bessler@execpc.com

Abstract

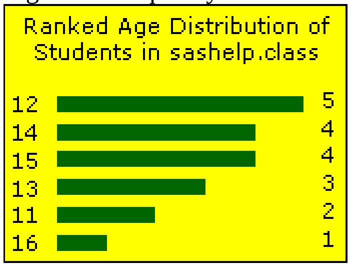
For over twenty years, I have been advocating for minimalist graphic design to focus on the message, not the media. I have demonstrated the power of simplicity by stripping out traditional graphic presentation paraphernalia while making communication-effective use of SAS and SAS/GRAPH. Delivery of a readable graph to a BlackBerry, an iPhone, or any small screen demands a simple design. This paper shows you how to provide not only visual indication of size or trend for quick and easy inference, but also precise numbers for reliable inference. You can manually attach the graphs to an email message, or you can make SAS programmatically email the graphs for you.

Introduction

You might already know how to send email from SAS. The challenge addressed here is how to format a readable graph for a BlackBerry, an iPhone, or other very small screen communication device. The minimalist image in Figure 1 is one example, published in the August 2007 issue of *SAS Tech Report* and of *SAS Training Report*, both available at http://support.sas.com. The example in Figure 4 was published in *VIEWS News: News, Hints, Tips, and Information for SAS Users*, Issue 39, 3rd Quarter 2007, available at http://www.views-uk.org.

In this paper you will find communication-effective design for the small screen with image and sufficient detail, and SAS/GRAPH code needed, for all of the popular management reporting formats. Here is a simple bar chart:

Figure 1. Frequency Bar Chart



Here is the code used to create and send the image above:

```
goptions reset=all;
goptions device=GIF;
goptions ypixels=129 xpixels=172; /* Default size for a GIF image is 800 X 600.
  Preserve 4-to-3 aspect ratio, but adjust the size for a BlackBerry or iPhone. */
goptions border; /* Put the graph in a box. */
goptions cback=CXFFFF00; /* Use RGB Yellow for background. */
goptions htext=10pt ftext='Verdana';
      /* Specify height and font used for those parts of the graph
         for which you do not make an explicit assignment,
         or for which no direct controls are available in SAS/GRAPH. */
goptions gsfmode=replace gsfname=anyname;
filename anyname "C:\FolderName\FileName.GIF";

title1 font='Verdana'
  color=CX000000   /* RGB black */
  height=3 PCT ' ' /* Insert space at the top. */
  height=10pt
  justify=CENTER   /* Force a new line. */
  "Ranked Age Distribution of"
  justify=CENTER   /* Force a new line. */
  "Students in sashelp.class";
```

```
footnote1 angle=+90 font=none height=1 ' ';
  /* This "footnote" is a blank "right-side-note"
     to increase space at the side of the image. */
  /* Use angle=-90 to create a "left-side-note" if needed. */

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

  /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none
      offset=(5 PCT,5 PCT);
   /* Offset the horizontal axis to insert extra space
      between bar midpoint label and bar start,
      and between bar end and bar response value. */

proc gchart data=sashelp.class;
hbar Age /
  discrete      /* Use Ages, not Age subranges. */
  freq
  freqlabel=' ' /* Suppress the Frequency column heading. */
  descending
  maxis=axis1 raxis=axis2
  width=2.5 space=2.5; /* bar width and spacing */
run; quit;

filename anyname2 email
  to="bessler@execpc.com"
  subject="Graph for Your BlackBerry"
  attach="C:\FolderName\FileName.GIF";

data _null_;
file anyname2;
put 'Please see the attached graph.';
run;
```

Your SAS program can send email to multiple addresses, with multiple attachments, with a CC and/or BCC, etc. For more information about SAS email in the SAS 9.1.3 OnlineDoc, go to:
http://support.sas.com/onlinedoc/913/getDoc/en/lrdict.hlp/a002058232.htm

Note: To get a similar looking result with a different release of SAS/GRAPH, it may be necessary to change one or more of the following: bar width, bar space, inserted space at the top of the graph, or inserted space at the right-hand side of the graph.

In the remainder of this paper I present groups of related figures and comments, but the associated SAS code is in the Appendix. For some figures, the code is not provided. Omitted code can be requested from the author. Code to send email attachments from SAS is presented only in the Introduction above.

I want to display the first gallery of graphs all on one page. So, I have some extra space here. Let me fill it with a macro that is used to prepare for the creation of each graph and that is referenced in the Appendix. Explanatory comments for some of its internal statements were given earlier in the Introduction.

```
%macro CommonPreliminaryCode(PathToGraph=,FigureNumber=,BackgroundColor=CXFFFF00);
goptions reset=all;
goptions device=GIF;
goptions ypixels=129 xpixels=172;
goptions border;
goptions cback=&BackgroundColor;
goptions gsfmode=replace gsfname=anyname;
filename anyname "&PathToGraph.\Figure&FigureNumber..GIF";
%mend  CommonPreliminaryCode;
```

Bar Charts and Pie Charts for the Small Screen

Figure 2. Sum Bar Chart

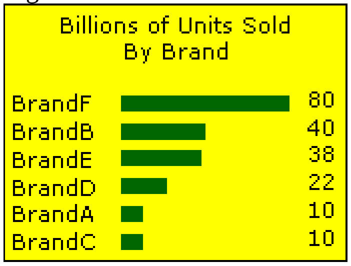
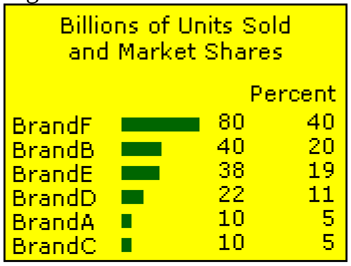


Figure 3. Pie Chart Alternative



To maximize the number of bars that can be displayed, remove the space between them and append the column of bar values to the bar names as shown in Figures 4 and 5.

Figure 4. Top Ten Chart

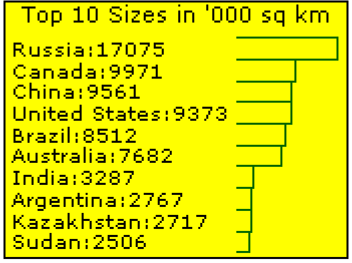
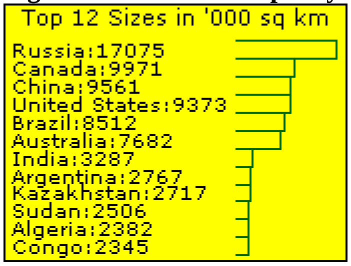
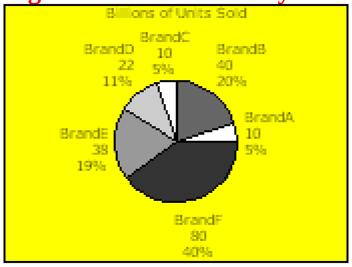


Figure 5. Maximum Capacity



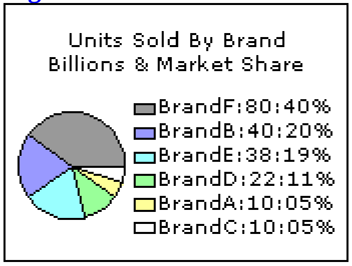
With 13 bars, the rows of text start to touch each other. The next smaller font is readable, but fuzzy as in Figure 6.

Figure 6. Text Too Fuzzy



The text in Figure 6 is too fuzzy, but the next larger font, used in Figures 4 and 5, causes pie slice label overlap. I have long recommended ordering bars and pie slices by size to "Show the Viewer What Is Important." Even with the small fuzzy font, descending slice order would force slice label overlap. The custom legend in Figure 7 is the only very readable pie chart solution and can always support descending or ascending slice size. Since Figure 7 uses a color-coded legend, I prefer to use a white background. Amazingly, the fuzzy text in Figure 6 is harder to read on a white background than on a yellow background.

Figure 7. Readable and Ordered



Trend Charts for the Small Screen

For at least fifteen years, I have been making the case that a sparse trend chart is a sufficient trend chart. It enables easy and quick visual inferences. Precise numbers are best provided in a companion table. However, there are always four precise values of interest to a typical viewer of a trend chart: start, end, minimum and maximum. There can be other points of interest on a trend chart, but those points are not amenable to automated programmatic detection and can be leveraged only by iterative ad hoc graph creation. I prefer a robust solution for hands-off, fully automated graph production applications. For iterative development of a graph for an ad hoc request, a robust solution reduces your turnaround time to meet the needs of the requestor.

Figure 8. Trend by Trading Day

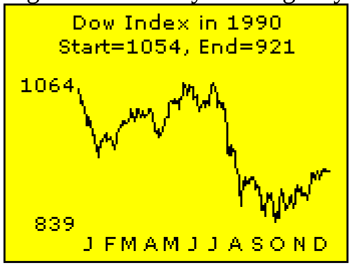


Figure 9. Different Use of Display Space

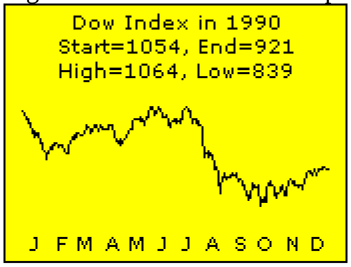


Figure 10. Reference Lines

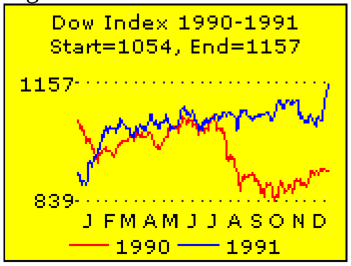


Figure 11. Best In Category

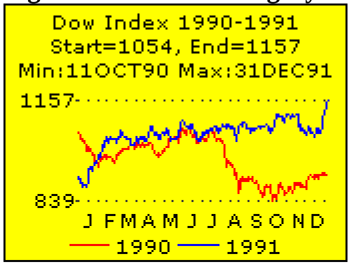
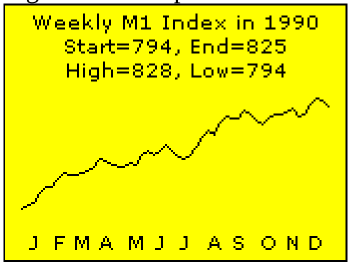


Figure 12. No Space for Week Numbers



In all of these trend charts, there is not enough space on the horizontal axis to present many tick mark labels. The irregular spacing of the month name initials above is due to the programming algorithm used to locate them. For Figures 8-11, since there are an average of 22 trading days in each month, trading day 11 is used. Since the actual number of trading days varies by month, so does the resulting spacing of labels. For Figure 12, where there are a variable number of weekly values in each month, week 3 is used, with irregular consequences for label spacing.

Figure 13. Vertical Tick Values

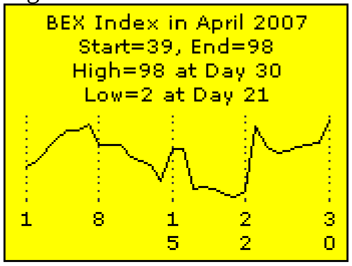
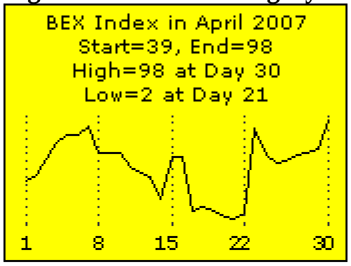


Figure 14. Best In Category



The stacked digits in Figure 13 are SAS/GRAPH default action for this data. I do not recommend vertical or tilted horizontal axis values—hence my preference for Figure 14. English is read left-to-right, not vertically or tilted. The solution for daily tick marks in a month always displays only 1, 8, 15, 22, and the last day of the month. The trick used for Figure 14 actually lays down the first digit of each two-digit day at the location of the day prior—to outwit SAS/GRAPH. The reference lines are used to overcome ambiguity. For a month of days, there are few enough points to be able to display plot symbols/markers, but I did not do so. For denser trends, they merge into a thick line.

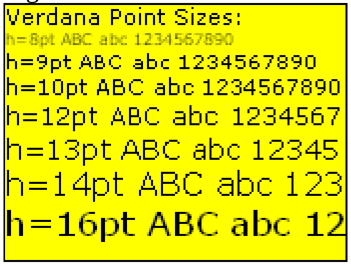
If the trend chart design solutions above do not provide sufficient information, additional precise numbers can be provided in a companion text message or a companion table. The usability of a large table on the small screen is a challenge—scrolling is a sub-optimal user experience even on the large screen.

Fonts and Font Sizing for the Small Screen

I evaluated all of the SAS/GRAPH “Software Fonts”, i.e., the fonts that are supplied with SAS/GRAPH and therefore work on any computing platform. I also evaluated all of the common Windows fonts, which I prefer. The SAS/GRAPH software fonts did not look very good, but if you do not have access to Windows fonts on your computer, the SWISS sans serif font and ZAPF and CENTX serif fonts may be acceptable, even if sub-optimal.

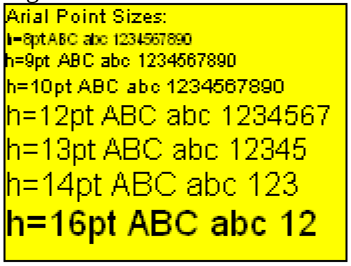
My favorite Windows sans serif font is Verdana. (This font was developed for Microsoft specifically for readability in on-screen display and in web pages.) I recommend its use on the small screen and anywhere else. Arial, undoubtedly the most commonly used Windows sans serif font, is acceptable, and offers the benefit of being narrower. This allows you to squeeze more text into the narrow width of the small screen, as seen by comparing Figures 15 and 16. For both of these fonts, the smallest practical point size is 9 pt. For the small text parts of graphs, sans serif fonts are more readable. The samples below are not a full display of the upper and lower case alphabets.

Figure 15. Verdana Point Sizes



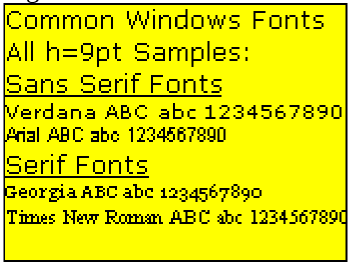
Verdana Point Sizes:
h=8pt ABC abc 1234567890
h=9pt ABC abc 1234567890
h=10pt ABC abc 1234567890
h=12pt ABC abc 1234567
h=13pt ABC abc 12345
h=14pt ABC abc 123
h=16pt ABC abc 12

Figure 16. Arial Point Sizes



Arial Point Sizes:
h=8pt ABC abc 1234567890
h=9pt ABC abc 1234567890
h=10pt ABC abc 1234567890
h=12pt ABC abc 1234567
h=13pt ABC abc 12345
h=14pt ABC abc 123
h=16pt ABC abc 12

Figure 17. Sans Serif vs. Serif



Common Windows Fonts
All h=9pt Samples:
Sans Serif Fonts
Verdana ABC abc 1234567890
Arial ABC abc 1234567890
Serif Fonts
Georgia ABC abc 1234567890
Times New Roman ABC abc 1234567890

Plot Symbols/Markers for the Small Screen

I evaluated all of the possibilities in the SAS/GRAPH symbol-oriented fonts. I also evaluated symbols that are available by specifying a string of two hexadecimal characters, as in VALUE='HH'X, on the SYMBOL statement when using the SAS/GRAPH text-oriented fonts. I further evaluated symbols available in a wide variety of Windows fonts—not only symbols available, by using VALUE='HH'X, in the extensions to fonts that are primarily Windows text fonts, but also symbols that are provided in Windows fonts of symbols, icons, and widgets—fonts such as Webdings, Wingdings, Monotype Sorts, etc. After all of this experimentation, I identified only two acceptable symbol shapes for the small screen. The color of tiny plot symbols is too hard to distinguish. Therefore, on the small screen it is necessary to use line color in order to distinguish several trend lines in the same graph.

On the small screen, it is infeasible, at least with SAS/GRAPH, to get nicely formed round plot symbols of significant size by any means. VALUE=POINT appears to produce round points, but they are very small and would not enlarge by increasing the HEIGHT (the size parameter for the SYMBOL statement). The larger VALUE=DOT produces varying amorphous shapes rather than round symbols of consistent shape. When trying to get squares, I usually got rectangles. I found no other shapes that were drawn well and consistently for all points in the graph. So, when you have few enough plot points so that they do not just merge into a thick line or curve, I recommend:

```
symboln color=black interpol=none height=1 font=Special value='K';
```

Figure 18: Point Symbol

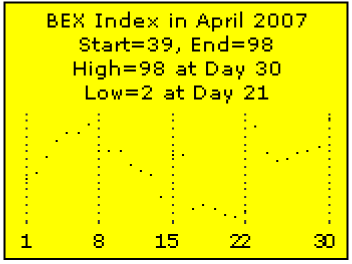


Figure 19: Symbol Special `'K'`

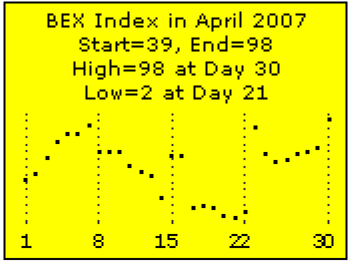
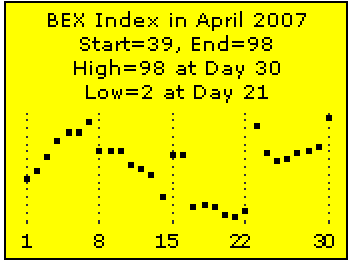


Figure 20: Special `'K'` `HEIGHT=2`



Adding a plot line with a symbol present adds no value and has an unattractive effect. Enlarging the symbol does not improve the result with a line present. For plot symbols on the small screen, these are the only good choices.

Conclusion

SAS/GRAPH can create readable images that work in the extreme conditions of the small screen. As has long been known, SAS can email graphs (and tables). Except for the problem demonstrated in Figure 6, the designs here are communication-effective for their respective purposes. Graphs facilitate and accelerate inference, but reliable inference requires detail numbers. All of the designs shown here provide both visual and precision. However, there is an obvious limitation on how much text and numbers you can present on the small screen.

References

This recent related work by the author on Communication-Effective Graphic Design and Construction is all available in Online Proceedings at http://support.sas.com/events/sasglobalforum/previous/online.html

1. Get the Best out of SAS/GRAPH and ODS*, Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

2. Communication-Effective Pie Charts*, Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

3. How to Make the "Best Choice" from the Many Ways to Create and Deliver SAS Graphs*, Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

Acknowledgment

I am grateful to Alix Riley for her helpful review of this paper and for testing the output on an iPhone.

Contact Information

Your comments, questions, and suggestions are welcome. I am always interested in design ideas or construction solutions that enhance graphic communication or address new information delivery needs.

LeRoy Bessler PhD
Email: bessler@execpc.com
Phone: 1 414 351 6748 (evenings and weekends—time is six hours earlier than Greenwich Mean Time)

Appendix. Code to Create Graphs for the Small Screen. NOTE: For some figures no code is presented.

NOTE: When creating graphs for the small screen, you might get messages in the SAS log saying that axis values have been overwritten. Only believe the warning if your inspection of the graph confirms it.

Input Data Not in SASHELP Data Library

Data for Figures 2, 3, 6, 7

```
data DataForSimpleCharts;
infile datalines; input @1 Name $6. @8 Value 2.;
datalines;
BrandA 10
BrandB 40
BrandC 10
BrandD 22
BrandE 38
BrandF 80
; run;
```

Data for Figures 4 & 5

```
data CountriesbySizeTop12
     CountriesbySizeTop10;
length NameAndValue $ 19;
infile datalines;
input @1 Name $13. @15 Value comma6.;
NameAndValue = trim(left(Name)) || ':' || trim(left(put(Value,5.)));
if _N_ LE 12 then output CountriesbySizeTop12;
if _N_ LE 10 then output CountriesbySizeTop10;
datalines; /* extra data present below */
Russia        17,075
Canada         9,971
China          9,561
Brazil         8,512
Australia      7,682
India          3,287
United States  9,373
Argentina      2,767
Kazakhstan     2,717
Sudan          2,506
Algeria        2,382
Congo          2,345
Saudi Arabia   2,200
Greenland      2,176
Mexico         1,973
;
run;
```

Data for Figures 8-11

```
data work.DOW19901991;
drop snydjcm;
retain FirstTradingDayIn1991Found 'N' TradingDay 0;
length MonthAbbrev $ 1;
set sashelp.citiday
   (keep=date snydjcm
    where=(snydjcm ne .));
Year = year(date);
if Year in (1990 1991);
if Year eq 1990
  or
   FirstTradingDayIn1991Found eq 'Y'
then TradingDay = TradingDay + 1;
else do;
  TradingDay = 1;
  FirstTradingDayIn1991Found = 'Y';
end;
Dow=round(snydjcm,1);
Month = month(date);
MonthAbbrev=substr(left(put(date,monname3.)),1,1);
run;
```

Data for Figures 13 & 14

```
data BEX2007April;
infile datalines;
input @1 Day 2. @4 BEX 2.;
datalines;
1  39 200704
2  46 200704
3  61 200704
4  76 200704
5  84 200704
6  84 200704
7  93 200704
```

```
8  67 200704
9  67 200704
10 67 200704
11 53 200704
12 48 200704
13 43 200704
14 21 200704
15 63 200704
16 63 200704
17 11 200704
18 13 200704
19 11 200704
20  4 200704
21  2 200704
22  8 200704
23 90 200704
24 65 200704
25 57 200704
26 59 200704
27 65 200704
28 67 200704
29 70 200704
30 98 200704
;
run;
```

Macros

Macros for Figure 7

```
%macro PatternStatements;
%do i = 1 %to 6;
pattern&i v=psolid color=&&SliceColor&i repeat=1;
%end;
%mend  PatternStatements;

%macro LegendEntries;
%do i = 1 %to 6;
  "&&LegendEntry&i"
%end;
%mend  LegendEntries;
```

Macros for All Trend Charts

```
%macro GetHtickmarkValues;

%do i = 1 %to &NumberOfTickMarks %by 1;
  "&&htickvalue&i"
%end;

%mend  GetHtickmarkValues;

%macro IdentifyYaxisCriticalPoints(data=,filter=,yVar=,yVarFormat=);

%global yVarStart yVarEnd;

data Selected;
set &data;
%if %length(&filter) ne 0 %then %do;
where &filter;
%end;
run;

data _null_;
set Selected end=LastOne;
if _N_ eq 1
```

```
then call symput('yVarStart',trim(left(put(&yVar,&yVarFormat))));
else
if LastOne
then call symput('yVarEnd'  ,trim(left(put(&yVar,&yVarFormat))));
run;

proc means data=Selected min max range noprint;
var &Yvar;
output out=MinMaxRange min=yMin max=yMax range=yMaxMinDiff;
run;

%global
  yAxisStartTickMarkValue yAxisStartTickMarkDisplay
  yAxisEndTickMarkValue   yAxisEndTickMarkDisplay
  yAxisIncrement;

data _null_;
set MinMaxRange;
call symput('yAxisStartTickMarkValue'  ,trim(left(yMin        )));
call symput('yAxisStartTickMarkDisplay',trim(left(put(yMin,&yVarFormat))));
call symput('yAxisEndTickMarkValue'    ,trim(left(yMax        )));
call symput('yAxisEndTickMarkDisplay'  ,trim(left(put(yMax,&yVarFormat))));
call symput('yAxisIncrement',trim(left(yMaxMinDiff)));
run;

data _null_;
set Selected;
if &yVar LT &yAxisStartTickMarkValue
then put &yVar= " is below yAxis Start &yAxisStartTickMarkValue";
if &yVar GT &yAxisEndTickMarkValue
then put &yVar= " is above yAxis End &yAxisEndTickMarkValue";
run;

%mend  IdentifyYaxisCriticalPoints;
```

Macro for Trend Chart Figures 8-11

```
%macro CreateTradeDayHtickmarkValues(data=,filter=);

* Maximum number of trading days in a year is 263. *;
* Possible extra GLOBAL statements are harmless. *;
%do i = 1 %to 262 %by 1;
  %global htickvalue&i;
%end;

* For the actual number of trading days found: *;
%global NumberOfTickMarks;

data Selected;
set &data;
%if %length(&filter) ne 0 %then %do;
where &filter;
%end;
run;

data _null_;
retain DayInMonth 0;
set Selected end=LastOne;
by Month;
if first.Month
then DayInMonth = 1;
else DayInMonth = DayInMonth + 1;
* The monthly average number of trading days is 22.    *;
* Want to put the month abbrev label at the mid-month. *;
if DayInMonth eq 11
then call symput('htickvalue'||trim(left(_N_)),MonthAbbrev);
```

```
else call symput('htickvalue'||trim(left(_N_)),'');
if LastOne;
call symput('NumberOfTickMarks',trim(left(_N_)));
run;

%mend  CreateTradeDayHtickmarkValues;
```

Macro for Trend Chart Figure 11

```
%macro IdentifyXdatesForCriticalPoints(data=,yVar=,xDateVar=);

%global LocateMinAndOrMaxWhenUnique;

data _null_;
length MinLoc MaxLoc $ 11;
retain HowManyXfoundForMinY HowManyXfoundForMaxY 0;
retain xForMinY . xForMaxY .;
set &data end=LastOne;
if &Yvar EQ &yAxisStartTickMarkValue
then do;
  if HowManyXfoundForMinY eq 0
  then do;
    HowManyXfoundForMinY = 1;
    xForMinY = &xDateVar;
  end;
  else HowManyXfoundForMinY + 1;
end;
else
if &Yvar EQ &yAxisEndTickMarkValue
then do;
  if HowManyXfoundForMaxY eq 0
  then do;
    HowManyXfoundForMaxY = 1;
    xForMaxY = &xDateVar;
  end;
  else HowManyXfoundForMaxY + 1;
end;
if LastOne;
put HowManyXfoundForMinY= HowManyXfoundForMaxY= xForMinY= xForMaxY=;
if HowManyXfoundForMinY eq 1
then MinLoc = "Min:" || trim(left(put(xForMinY,date7.)));
else MinLoc = " ";
if HowManyXfoundForMaxY eq 1
then MaxLoc = "Max:" || trim(left(put(xForMaxY,date7.)));
else MaxLoc = " ";
call symput('LocateMinAndOrMaxWhenUnique',trim(left(MinLoc)) || " " ||
trim(left(MaxLoc)));
run;

%mend  IdentifyXdatesForCriticalPoints;
```

Macros for Trend Chart Figures 13 & 14

```
%macro IdentifyXvalsForCriticalPoints(data=,yVar=);

%global AtXvalueWhenMinYisUnique AtXvalueWhenMaxYisUnique;

data _null_;
retain HowManyXfoundForMinY HowManyXfoundForMaxY 0;
retain xForMinY . xForMaxY .;
set &data end=LastOne;
if &Yvar EQ &yAxisStartTickMarkValue
then do;
  if HowManyXfoundForMinY eq 0
  then do;
    HowManyXfoundForMinY = 1;
    xForMinY = Day;
  end;
  else HowManyXfoundForMinY + 1;
end;
else
if &Yvar EQ &yAxisEndTickMarkValue
then do;
  if HowManyXfoundForMaxY eq 0
  then do;
    HowManyXfoundForMaxY = 1;
    xForMaxY = Day;
  end;
  else HowManyXfoundForMaxY + 1;
end;
if LastOne;
put HowManyXfoundForMinY= HowManyXfoundForMaxY= xForMinY= xForMaxY=;
if HowManyXfoundForMinY eq 1
then call symput('AtXvalueWhenMinYisUnique',
" at Day " || trim(left(xForMinY)));
if HowManyXfoundForMaxY eq 1
then call symput('AtXvalueWhenMaxYisUnique',
" at Day " || trim(left(xForMaxY)));
run;

%mend  IdentifyXvalsForCriticalPoints;
```

Macro for Trend Chart Figure 14

```
%macro BetterDayOfMonHtickmarkValues(data=);

%global NumberOfTickMarks;

data _null_;
set &data nobs=HowMany;
call symput('NumberOfTickMarks',trim(left(HowMany)));
call symput('SecondLast',trim(left(HowMany - 1)));
stop;
run;

%do i = 1 %to &NumberOfTickMarks %by 1;
  %global htickvalue&i;
%end;

data _null_;
set &data;
if _N_ in (1 8)
then call symput('htickvalue'||trim(left(_N_)),trim(left(_N_)));
else
if _N_ eq 14
then call symput('htickvalue'||trim(left(_N_)),'1');
else
if _N_ eq 15
then call symput('htickvalue'||trim(left(_N_)),'5');
else
if _N_ in (21 22)
then call symput('htickvalue'||trim(left(_N_)),'2');
else
if _N_ eq %superq(SecondLast)
then call symput('htickvalue'||trim(left(_N_)),
                 trim(left(%substr(%superq(NumberOfTickMarks),1,1))));
else
if _N_ eq %superq(NumberOfTickMarks)
then call symput('htickvalue'||trim(left(_N_)),
                 trim(left(%substr(%superq(NumberOfTickMarks),2,1))));
else call symput('htickvalue'||trim(left(_N_)),'');
run;
```

```
%mend  BetterDayOfMonHtickmarkValues;
```

Code

See the Introduction for the source listing of the CommonPreliminaryCode macro used repeatedly in the remainder of the Appendix.

```
options mprint symbolgen;

%let ThePath = C:\Folder;
```

Code for Figure 2

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=2);

goptions htext=10pt ftext='Verdana';
      /* Specify height and font used for those parts of the graph
         for which you do not make an explicit assignment,
         or for which no direct controls are available in SAS/GRAPH. */

title1 font='Verdana'
  color=CX000000   /* RGB black */
  height=3 PCT ' ' /* Insert space at the top. */
  height=10pt
  justify=CENTER   /* Force a new line. */
  "Billions of Units Sold"
  justify=CENTER   /* Force a new line. */
  "By Brand";

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

  /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none
      offset=(5 PCT,5 PCT);
   /* Offset the horizontal axis to insert extra space
      between bar midpoint label and bar start,
      and between bar end and bar response value. */

proc gchart data=DataForSimpleCharts;
hbar Name /
  sumvar=Value
  sum
  sumlabel=' ' /* Suppress the Sum column heading. */
  descending
  maxis=axis1 raxis=axis2
  width=2.5 space=2.5; /* bar width and spacing */
run; quit;
```

Code for Figure 3

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=3);

goptions htext=10pt ftext='Verdana';
      /* Specify height and font used for those parts of the graph
         for which you do not make an explicit assignment,
         or for which no direct controls are available in SAS/GRAPH. */

title1 font='Verdana'
  color=CX000000   /* RGB black */
  height=3 PCT ' ' /* Insert space at the top. */
  height=10pt
  /* height=9pt smaller text for two-line title */
  justify=CENTER   /* Force a new line. */
  "Billions of Units Sold"
  justify=CENTER   /* Force a new line. */
  "and Market Shares";

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

  /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0 offset=(0,15 PCT);
      /* Without the OFFSET as a problem circumvention,
         the column labels would overlap the first row of values */
axis2 label=none major=none minor=none style=0 value=none
      offset=(5 PCT,5 PCT);
   /* Offset the horizontal axis to insert extra space
      between bar midpoint label and bar start,
      and between bar end and bar response value. */

proc gchart data=DataForSimpleCharts;
hbar Name /
  freq=Value
  freq
  freqlabel=' ' /* Suppress the Freq column heading. */
  percent
  percentlabel='Percent'
  descending
  maxis=axis1 raxis=axis2
  width=2 space=2.5; /* slightly reduce bar width */
run; quit;
```

Code for Figure 5

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=5);

 /* increase space at left-hand side */
footnote1 angle=-90 font=none height=1 ' ';

title1 font='Verdana' height=10pt "Top 12 Sizes in '000 sq km";

goptions htext=9pt ftext='Verdana';

  /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none;

pattern1 v=empty color=CX006600; /* RGB Medium Dark Green bars, outline only */

proc gchart data=CountriesbySizeTop12;
hbar NameAndValue /
     sumvar=Value
     descending
     maxis=axis1 raxis=axis2
     nostats  /* no values at right margin */
     space=0; /* default width for bars, no space between them */
run; quit;
```

Code for Figure 7

Prepare to Create the Custom Pie Chart

```
proc means data=DataForSimpleCharts noprint sum;
var Value;
output out=PieTotal sum=TotalValue;
run;
```

```
 /* concatenate Percent and Value with the original Slice Name */
data SliceNameWithPercentAndValue;
length NameWithPercentAndValue $ 15;
set DataForSimpleCharts;
if _N_ eq 1 then set PieTotal;
Percent = (Value / TotalValue) * 100;
NameWithPercentAndValue =
                    trim(left(Name)) || ':' || trim(left(put(Value,2.))) || ':' ||
                    trim(left(put(Percent,Z2.))) || '%';
run;

proc sort data=SliceNameWithPercentAndValue;
by descending Percent; /* sequence the slices from largest to smallest */
run;

data SliceNameWithPercentAndValue;
set SliceNameWithPercentAndValue;
pctseq = _N_; /* add a key to merge this data with the ColorList    */
call symput('LegendEntry'||trim(left(_N_)),trim(left(NameWithPercentAndValue)));
              /* Store the legend text entries in the symbol table. */
run;

data ColorList; /* Sequence the colors from darkest to lightest,
                   if you want small slices to stand out. */
pctseq = 1; color='CX999999'; output;
pctseq = 2; color='CX9999FF'; output;
pctseq = 3; color='CX99FFFF'; output;
pctseq = 4; color='CX99FF99'; output;
pctseq = 5; color='CXFFFF99'; output;
pctseq = 6; color='CXFFFFFF'; output;
run;

data SliceWithColor;
merge SliceNameWithPercentAndValue ColorList;
by pctseq;
run;

proc sort data=SliceWithColor;
by NameWithPercentAndValue; /* because SAS/GRAPH will always apply
  your PATTERN statements in the sort order of the SLICE name text */
run;

data _null_;
set SliceWithColor;
call symput('SliceColor'||trim(left(_N_)),trim(left(color)));
            /* Store the slice colors in the symbol table. */
run;
```

Create the Custom Pie Chart

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=7,
  BackgroundColor=CXFFFFFF); /* RGB White background because color legend for pie. */

footnote1 angle=0   font=none height=1 ' '; /* white space at bottom */
footnote2 angle=-90 font=none height=1 ' '; /* push pie to the right */

title1 font='Verdana'
  height=9pt ' '; /* Insert space at the top. */
title2 font='Verdana'
  color=CX000000   /* RGB black */
  height=9pt
  'Units Sold By Brand'
  justify=CENTER   /* Force a new line. */
  'Billions & Market Share';

goptions htext=9pt ftext='Verdana';
```

```
legend1 order=(%LegendEntries) label=none shape=bar(6 PCT,4 PCT)
        across=1 position=(middle right outside) offset=(-2 PCT,0);

%PatternStatements;

proc gchart data=SliceNameWithPercentAndValue;
pie NameWithPercentAndValue / sumvar=Value
    noheading coutline=CX000000 descending
    legend=legend1
    slice=none value=none percent=none; /* turn off all pie labels */
run; quit;
```

Code for Figure 11

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=11);

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';
 /* for multi-line plots, presence of legend creates extra white space at bottom. */
footnote3; /* turn off white space at bottom. */
title1 font=none height=1 ' ';

goptions htext=9pt ftext='Verdana';

* 1990 and 1991 had 262 and 263 trading days.       *;
* On a two-line plot, use 263 invisible tick marks. *;
%CreateTradeDayHtickmarkValues(data=DOW19901991,filter=%str(Year eq 1991));
axis1 label=none major=none minor=none style=0
      order=1 to &NumberOfTickMarks by 1
      value=(%GetHtickmarkValues);

%IdentifyYaxisCriticalPoints(data=DOW19901991,yVar=Dow,yVarFormat=4.);
axis2 label=none major=none minor=none style=0
      order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
      value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

title2 font='Verdana'
justify=CENTER
height=9pt
"Dow Index 1990-1991"
justify=CENTER
"Start=&yVarStart, End=&yVarEnd";

%IdentifyXdatesForCriticalPoints(data=DOW19901991,yVar=DOW,xDateVar=date);
title3 font='Verdana' height=9pt
  "&LocateMinAndOrMaxWhenUnique";

symbol1 color=CXFF0000 interpol=join v=none w=1;
symbol2 color=CX0000FF interpol=join v=none w=1;

legend1 label=none shape=symbol(15,1);

proc gplot data=DOW19901991;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2 legend=legend1
     vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue cvref=CX333333 lvref=35;
     /* dark gray dotted reference lines at top and bottom of vaxis range */
run; quit;
```

Code for Figure 14

```
%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=14);
goptions htext=9pt ftext='Verdana';

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';
footnote3 angle=0   font=none height=1 ' ';
```

```
title1 font=none height=1 ' ';

%IdentifyYaxisCriticalPoints(data=BEX2007April,yVar=BEX,yVarFormat=2.);
title2 font='Verdana'
justify=CENTER
height=9pt
"BEX Index in April 2007"
justify=CENTER
"Start=&yVarStart, End=&yVarEnd";
%IdentifyXvalsForCriticalPoints(data=BEX2007April,yVar=BEX);
title3 font='Verdana' height=9pt
  "High=&yAxisEndTickMarkDisplay"
  "&AtXvalueWhenMaxYisUnique"
  justify=CENTER
  "Low=&yAxisStartTickMarkDisplay"
  "&AtXvalueWhenMinYisUnique";

* In situations where SAS/GRAPH would insist on
  rendering multi-digit tick mark values vertically,
  this macro delivers them horizontally.
  Ambiguity as to location of the tick mark vs. the point
  can be eliminated with reference lines. *;
%BetterDayOfMonHtickmarkValues(data=BEX2007April);
axis1 label=none major=none minor=none style=0
      order=1 to &NumberOfTickMarks by 1
      value=(%GetHtickmarkValues);
axis2 label=none major=none minor=none style=0 value=none;
symbol1 color=CX000000 interpol=join v=none;
proc gplot data=BEX2007April;
plot BEX*Day / haxis=axis1 vaxis=axis2
     href=1 8 15 22 30 chref=CX333333 lhref=35;
     /* dark gray dotted reference lines at one-week intervals of haxis range */
run; quit;
```

Appendix 2: Graphs Direct to PowerPoint.

Create and Deliver Graphs Directly in a Set of PowerPoint Slides Using SAS®

LeRoy Bessler, Assurant Health, Milwaukee, USA, bessler@execpc.com

Abstract

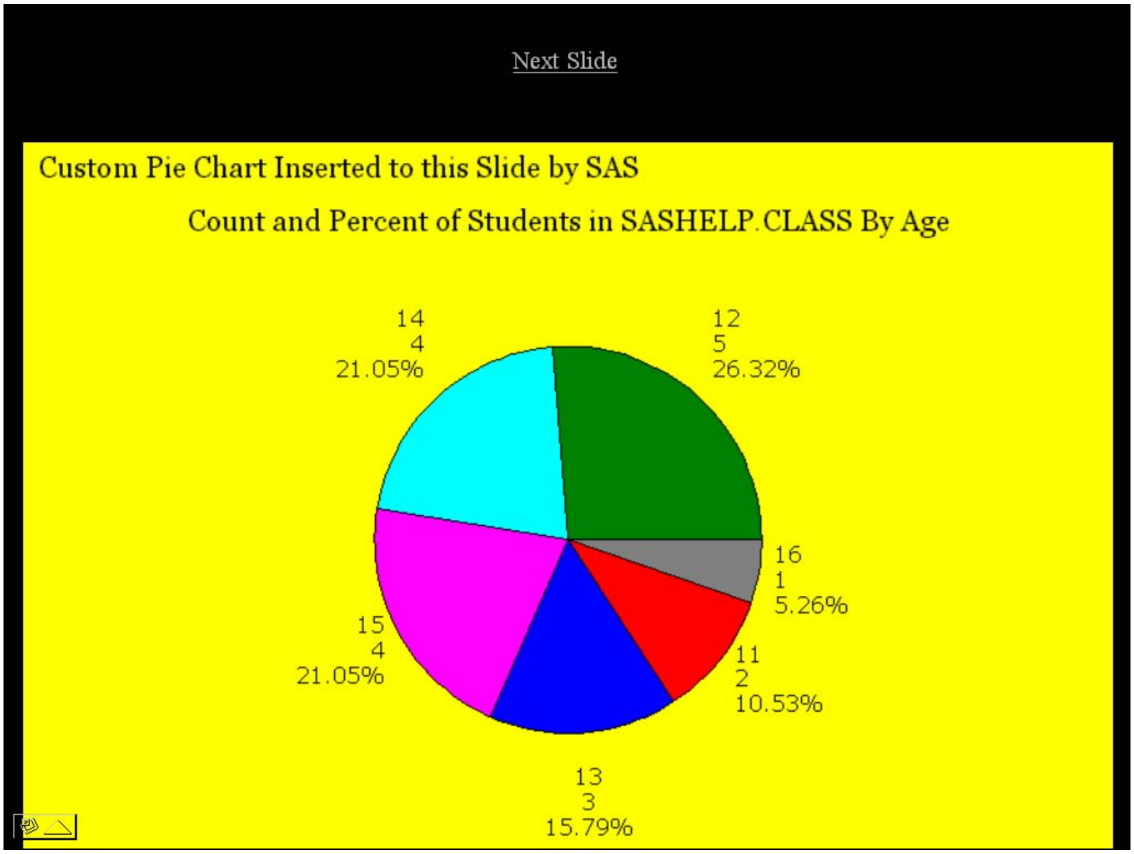
This is not a formal write-up as typically provided in a paper. Instead, it is a transcription of some text slides, screen image prints of the resulting graph slides, and a full listing of the program used to create the set of graphs-direct-to-slides. For Wisconsin Illinois SAS Users Conference 2007, I provided a slide presentation, a sample program, and the folder of PowerPoint slides created with the program. What follows is based on those materials.

How I Did It

- ODS HTML FILE=‘C:\RelocatableFolder\File.pps’;
  pps = Open As Show
  ppt = Open in Editable / Printable mode
- Idea from Phil Holland’s “Everyday Uses of ODS”
- Problem: creates standalone single-slide files
- Solution: Use macro to hyperlink the files, and store results in common relocatable folder

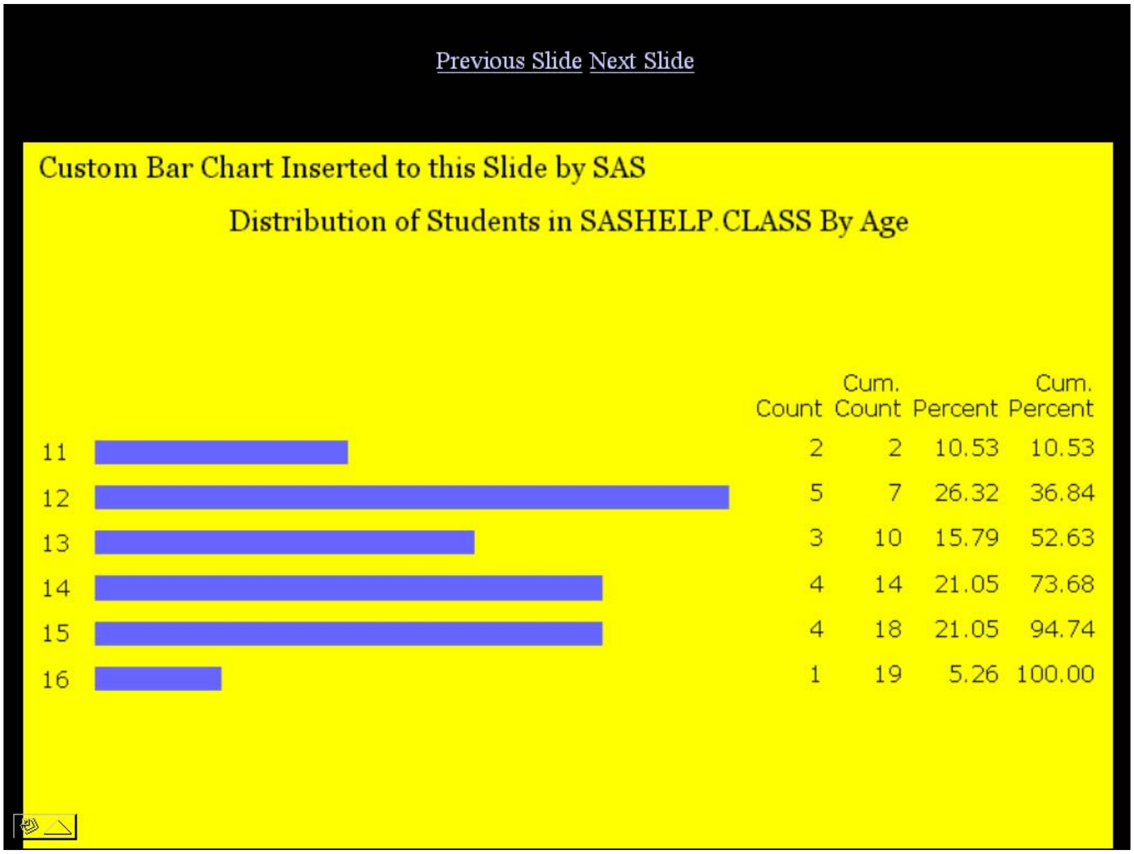
Screen Images of Three Linked PowerPoint Slides

One can go to the folder of slides, open any one of them, and proceed to view the entire set. Below is what they look like.



Custom Pie Chart Inserted to this Slide by SAS

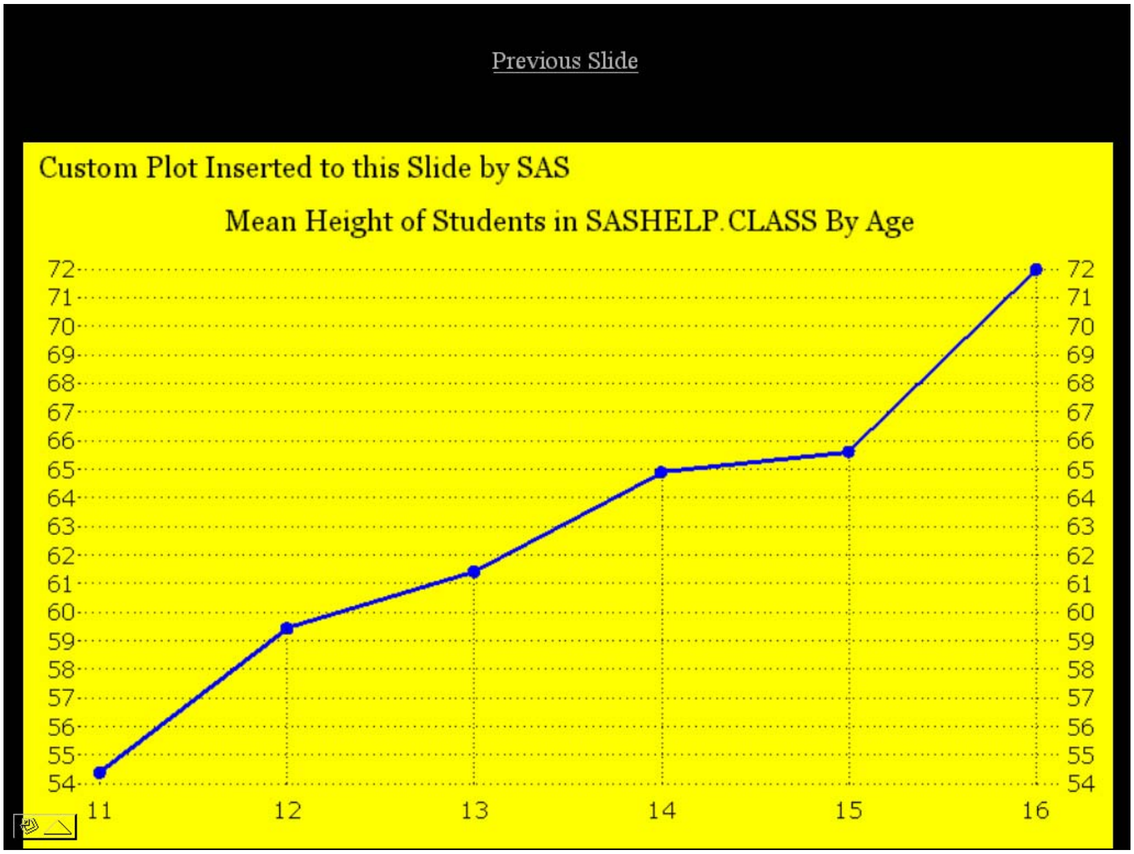
Count and Percent of Students in SASHELP.CLASS By Age



Custom Bar Chart Inserted to this Slide by SAS

Distribution of Students in SASHELP.CLASS By Age

	Count	Cum. Count	Percent	Cum. Percent
11	2	2	10.53	10.53
12	5	7	26.32	36.84
13	3	10	15.79	52.63
14	4	14	21.05	73.68
15	4	18	21.05	94.74
16	1	19	5.26	100.00



Custom Plot Inserted to this Slide by SAS

Mean Height of Students in SASHELP.CLASS By Age

Typical Macro Invocation

To create graphs directly in PowerPoint slides requires the use of two macros:

```
%SlideStart(
Path=C:\RelocatableFolderOfSlides,
PreviousSlideFileName=SlideNminus1,
ThisSlideFileName=SlideN,
NextSlideFileName=SlideNplus1,
SlideFileType=pps, /* Open as Show, ppt for edit/print */
Xpixels=775, Ypixels=505,
SlideBackgroundColor=CX000000,
GraphBackgroundColor=CXFFFF00,
LinkTextColor=CX000000,
LinkTextSize=16 PT);

 /* your SAS/GRAPH PROC Step code here */

%SlideEnd
```

Full Macro and Program Code Used To Create Linked Slides in a Folder

```
* Author: LeRoy Bessler PhD - bessler@execpc.com *;
* Date:   11 June 2007 *;

%let GraphImageFileWidth  = 775;
%let GraphImageFileHeight = 505;
%let PathToSlidesFolder   = C:\RelocatableFolderOfSlides;

%macro SlideEnd;
ods html close;
ods listing;
%mend SlideEnd;

%macro SlideStart(
Path=,
PreviousSlideFileName=,
ThisSlideFileName=,
NextSlideFileName=,
SlideFileType=pps, /* Use ppt for slide file that opens in editable mode,
                      but pps can be renamed to ppt later (and vice versa). */
Xpixels=775,
Ypixels=495,
SlideBackgroundColor=CXFFFFFF,
GraphBackgroundColor=CXFFFFFF,
LinkTextColor=CX000000,
LinkTextSize=16 PT);

proc template;
edit styles.Minimal as styles.Minimalbackground&SlideBackgroundColor;
style body /
  pagebreakhtml = _undef_
  background = &SlideBackgroundColor;
  /* web page background:
     all background not otherwise specified */
end;
run;

ods noresults;
ods listing close;

ods html path  = "&Path" /* folder for the PowerPoint files */
         (url  = none) /* use Path folder for the graph output
                          for portability of html and GIF */
         file  = "&ThisSlideFileName..&SlideFileType"
         style = styles.Minimalbackground&SlideBackgroundColor
         gtitle gfootnote;

goptions reset=all;
title h=&LinkTextSize c=&LinkTextColor
%if %length(&PreviousSlideFileName) ne 0 %then %do;
link="&PreviousSlideFileName..&SlideFileType" 'Previous Slide'
%end;
%if %length(&NextSlideFileName)     ne 0 %then %do;
link="&NextSlideFileName..&SlideFileType"     'Next Slide'
%end;
;
footnote;
PROC PRINT data=sashelp.class(where=(name='Carol')) label noobs
  style=[
  Foreground=&SlideBackgroundColor
  Background=&SlideBackgroundColor
  Font_Size=1pt
  rules = NONE
  frame = VOID
  borderwidth = 0
  cellpadding = 0
  cellspacing = 0];
var name;
label name='00'X;
RUN;

/* Above, style=[ . . . ] can use (but some may have no effect here):
ASIS, BACKGROUND, BACKGROUNDIMAGE, BODYSCROLLBAR, BODYSIZE,
BORDERCOLOR, BORDERCOLORDARK, BORDERCOLORLIGHT, BORDERWIDTH,
BOTTOMMARGIN, BULLET,
CELLHEIGHT, CELLPADDING, CELLSPACING, CELLWIDTH,
CONTENTPOSITION, CONTENTSCROLLBAR, CONTENTSIZE,
CSSSTYLE, CSSTEXT, FILLRULEWIDTH, FLYOVER,
FONT, FONT_FACE, FONT_SIZE, FONT_STYLE, FONT_WEIGHT, FONT_WIDTH,
FOREGROUND, FRAME, FRAMEBORDER, FRAMEBORDERWIDTH, FRAMESPACING,
HREFTARGET, HTMLCLASS, HTMLCONTENTTYPE, HTMLDOCTYPE, HTMLID, HTMLSTYLE,
JUST, LEFTMARGIN, LINKCOLOR, LISTENTRYANCHOR, LISTENTRYDBLSPACE,
NOBREAKSPACE, OUTPUTHEIGHT, OUTPUTWIDTH, OVERHANGFACTOR, PAGEBREAKHTML,
POSTHTML, POSTIMAGE, POSTTEXT, PREHTML, PREIMAGE, PRETEXT,
PROTECTSPECIALCHARS, RIGHTMARGIN, RULES, TAGATTR, TOPMARGIN, URL,
VISITEDLINKCOLOR, VJUST, WATERMARK. */

 /* Clear the graphics catalog */
proc catalog cat=work.gseg kill;
run; quit;

GOPTIONS
RESET = ALL
DEVICE = GIF
CBACK = &GraphBackgroundColor
NOBORDER
 /* IF YOU TURN ON BORDER, YOU MUST DECREASE XPIXELS & YPIXELS */
XPIXELS=&Xpixels
YPIXELS=&Ypixels;

%mend SlideStart;

options mprint;

%SlideStart(
Path=&PathToSlidesFolder,
PreviousSlideFileName=,
ThisSlideFileName=CustomPieChart,
NextSlideFileName=CustomBarChart,
Xpixels=&GraphImageFileWidth,
Ypixels=&GraphImageFileHeight,
SlideBackgroundColor=CX000000,
GraphBackgroundColor=CXFFFFFF,
LinkTextColor=CX000000,
LinkTextSize=16 PT);

goptions ftext='Verdana'; /* font  for text for which no font=   is assigned */
goptions htext=3.75 PCT;  /* height of text for which no height= is assigned */
title1 height=1 PCT ' ';              /* white space at top    of graph */
footnote1 angle=0 height=1 PCT ' ';   /* white space at bottom of graph */
footnote2 angle=+90 height=1 PCT ' '; /* white space at right  of graph */
footnote3 angle=-90 height=1 PCT ' '; /* white space at left   of graph */
title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Custom Pie Chart Inserted to this Slide by SAS'
  justify=LEFT height=2.5 PCT " "; /* white space before the next TITLE line */
title3 font='Georgia' height=5 PCT color=CX000000
  'Count and Percent of Students in SASHELP.CLASS By Age';

/* in HEIGHT= above, and in OFFSET= later, PCT means Percent of total graph height */

proc gchart data=sashelp.class;
* For a pie chart of total weight of students in each Age group,
  use SUMVAR=WEIGHT after the / in the PIE statement. *;

pie age /* create pie slices for AGE */
  /
  name='PieSlid'
  discrete   /* use the discrete values of AGE,
                not midpoints of default subranges  */
  noheading  /* suppress the default heading
                (in this case, "FREQUENCY of Age")  */
  descending /* order the slices by decreasing size */
  percent=outside;
/* Display Percent of Total Frequency for each slice.
   VALUE and SLICE (Slice Name) are defaults.
   Other options are NONE, INSIDE, and ARROW.
   "ARROW" connects display outside to the slice. */
/* To turn off pie slice color, include a PATTERN statement like this:
   PATTERN1 VALUE=PEMPTY REPEAT=99;
   99 can be any number greater than or equal to the number of slices.
   For how to control pie slice color,
   please see Reference 2 or the SAS/GRAPH manual. */

run; quit;

%SlideEnd;

%SlideStart(
Path=&PathToSlidesFolder,
PreviousSlideFileName=CustomPieChart,
ThisSlideFileName=CustomBarChart,
NextSlideFileName=CustomPlot,
Xpixels=&GraphImageFileWidth,
Ypixels=&GraphImageFileHeight,
SlideBackgroundColor=CX000000,
GraphBackgroundColor=CXFFFFFF,
LinkTextColor=CX000000,
LinkTextSize=16 PT);

goptions ftext='Verdana'; /* font  for text for which no font=   is assigned */
goptions htext=3.75 PCT;  /* height of text for which no height= is assigned */
```

```
title1 height=1 PCT ' ';              /* white space at top    of graph */
footnote1 angle=0 height=1 PCT ' ';   /* white space at bottom of graph */
footnote2 angle=+90 height=1 PCT ' '; /* white space at right  of graph */
footnote3 angle=-90 height=1 PCT ' '; /* white space at left   of graph */

title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Custom Bar Chart Inserted to this Slide by SAS'
  justify=LEFT height=2.5 PCT " "; /* white space below TITLE2 */

title3 font='Georgia' height=5 PCT color=CX000000
  'Distribution of Students in SASHELP.CLASS By Age';

pattern1
  value=solid     /* solid fill for the bars   */
  color=CX6666FF; /* lighter browser-safe blue */

axis1
  label=none /* suppress the axis label */
  major=none /* suppress tick marks for every value along the axis */
  minor=none /* suppress tick marks between any major tick marks */
  style=0    /* suppress the axis line */
  offset=(0,7 PCT);
  /* OFFSET for the top of the axis prevents Count and Percent labels from
     overlapping the values listed below them for the top bar in the charts. */

axis2 label=none major=none minor=none style=0
  value=none /* suppress values along the axis */
  offset=(2 PCT, 2 PCT); /* create white space at each end of the axis */

proc gchart data=sashelp.class;

hbar age /
  name='BarSlid'
  discrete /* use the discrete values of AGE,
              not midpoints of default subranges                             */
  freq /* Frequency of the discrete values of AGE is the RESPONSE variable   */
       /* list the values for each AGE in a column to the right of the bars  */
  freqlabel='Count' /* custom label for Frequency column                     */
  cfreq  /* provide a Cumulative Frequency column  to the right of the bars  */
  cfreqlabel='Cum. Count' /* custom label for Cumulative Frequency column    */
  percent /* list the Percent of total frequency for each AGE in a column    */
  percentlabel='Percent' /* custom label for Percent column                  */
  cpercent  /* provide a Cumulative Percent column  to the right of the bars */
  cpercentlabel='Cum. Percent' /* custom label for Cumulative Percent column */
  maxis=axis1 /* AXIS1 statement defines the axis for the MIDPOINT variable  */
  raxis=axis2 /* AXIS2 statement defines the axis for the RESPONSE variable  */
  width=1.6   /* make bar width close to the height of the text              */
  space=1.6;  /* make space between bars equal to the bar width              */

run; quit;

%SlideEnd;

%SlideStart(
Path=&PathToSlidesFolder,
PreviousSlideFileName=CustomBarChart,
ThisSlideFileName=CustomPlot,
NextSlideFileName=,
Xpixels=&GraphImageFileWidth,
Ypixels=&GraphImageFileHeight,
SlideBackgroundColor=CX000000,
GraphBackgroundColor=CXFFFFFF,
LinkTextColor=CX000000,
LinkTextSize=16 PT);

proc means data=sashelp.class mean noprint nway;
class age;
```

```
var height;
output out=MeanHeightByAge mean=MeanHeight;
run;

goptions ftext='Verdana'; /* font  for text for which no font=   is assigned */
goptions htext=3.75 PCT;  /* height of text for which no height= is assigned */

title1 height=1 PCT ' ';              /* white space at top    of graph */
footnote1 angle=0 height=1 PCT ' ';   /* white space at bottom of graph */
footnote2 angle=+90 height=1 PCT ' '; /* white space at right  of graph */
footnote3 angle=-90 height=1 PCT ' '; /* white space at left   of graph */

title2 font='Georgia' height=5 PCT
  justify=LEFT color=CX000000 '  Custom Plot Inserted to this Slide by SAS'
  justify=LEFT height=2.5 PCT " ";

title3 font='Georgia' height=5 PCT color=CX000000
'Mean Height of Students in SASHELP.CLASS By Age';

symbol1
  value=dot /* other possibilities include POINT, NONE, etc. */
  height=1.5
  interpol=join /* connect the dots */
  line=1 /* solid line */
  width=2 /* thicken it */
  color=CX0000FF; /* browser-safe blue */

symbol2
  value=none /* suppress the marker */
  interpol=needle /* line from the axis to the marker */
  line=33 /* finest dotted line */
  color=CX000000; /* browser-safe black */

axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 offset=(2 PCT, 2 PCT);

/* For this data, there was no need to use the ORDER= and VALUE=
   axis tick mark value controls available for the AXIS statement. */

proc gplot data=MeanHeightByAge;

* PLOT  statement defines a plot using the Left-Hand-Side vertical axis *;
plot MeanHeight*age=1 /* use SYMBOL1 statement for this plot */
  /
  name='PlotSlid'
  vaxis=axis1 /* AXIS1 statement defines the vertical   axis          */
  haxis=axis2 /* AXIS2 statement defines the horizontal axis          */
  autovref /* a reference line at every vertical axis tick mark value */
  lvref=33; /* linetype 33 for vertical axis reference lines          */
  /* (You can use cvref= to customize color of reference lines.)      */

* PLOT2 statement defines a plot using the Right-Hand-Side vertical axis *;
plot2 MeanHeight*age=2 /* use SYMBOL2 statement for this plot */
  / vaxis=axis1;

run; quit;

%SlideEnd;
```

Caution

Your results may differ based on version of SAS, SAS/GRAPH, and PowerPoint.

Challenge

Let me know if you can find a way to link slide files with, e.g., VB script, so that the slide header links can be omitted. (More vertical space for image.)

Tables Direct To PowerPoint

In principle, you could create a tabular listing directly in one or more PowerPoint slides.

- If many rows, split across multiple slides
- Table presented as one table part per slide
- Link first table part back to prior table / graph slide
- Link first table part forward to next table part
- Link middle table parts backward and forward
- Link last table part to next table or graph slide

I did some development work for this purpose, but am not presenting it here.

References (Recent Work by this Author on Communication-Effective Graphic Design and Construction)

1. Get the Best out of SAS/GRAPH and ODS, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

2. Communication-Effective Pie Charts, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

Acknowledgments

I am grateful to Dr. Philip R. Holland for sharing a paper that gave me the idea for this work.

Contact Information

Your comments, questions, and suggestions are welcome. I am always interested in ways to get SAS and SAS/GRAPH software to do more.

LeRoy Bessler PhD
Email: bessler@execpc.com
Phone: 1 414 351 6748 (evenings and weekends—time is six hours earlier than Greenwich Mean Time)