PhUSE 2011

Paper CS01

Effective SAS greplay'ing and how to avoid stretching

David J. Mottershead, Quanticate Ltd., Oxford, UK

ABSTRACT

This paper resolves the issue of creating SAS® greplay figures with common titles and footnotes without stretching the original figures and their text.

Figures will be created using proc gplot. The creation of titles and footnotes using the gslide procedure will be explained. Some background of the greplay procedure will be explained as well as the creation of custom templates. The importance of setting the goptions which control the size of figures will be discussed and how to calculate the required values of the options for the exact size of figure you want to create. The relationship between text size and figure size will be explained, in particular resolving how to have uniform text size within the combined figure. Example code will demonstrate how to create the combined figure using the greplay procedure.

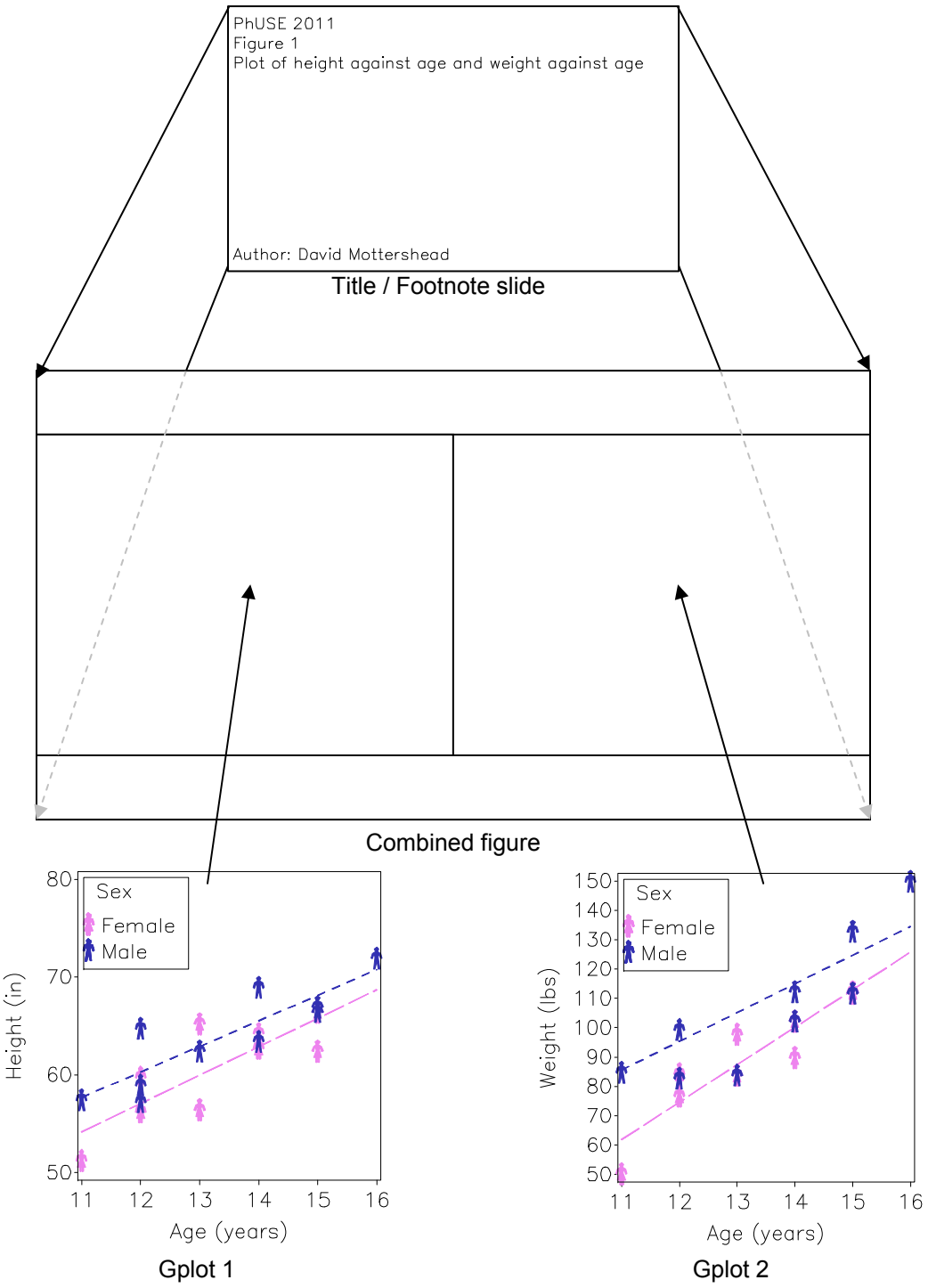
INTRODUCTION

In the pharmaceutical industry it is common to want to display multiple figures on one page. We also want to have common titles and footnotes which span the entire width of the page. Usually we require the font size to be the same throughout the whole of the combined figure.

This paper will use example code to show how to replay 2 figures horizontally with space for titles and footnotes. The example code will compare height and weight data plotted against age created with the gplot procedure. To do this we need to combine 3 figures:

- A titles and footnotes slide
- The first gplot figure we want to include
- The second gplot figure we want to include

This is visualized on the next page.

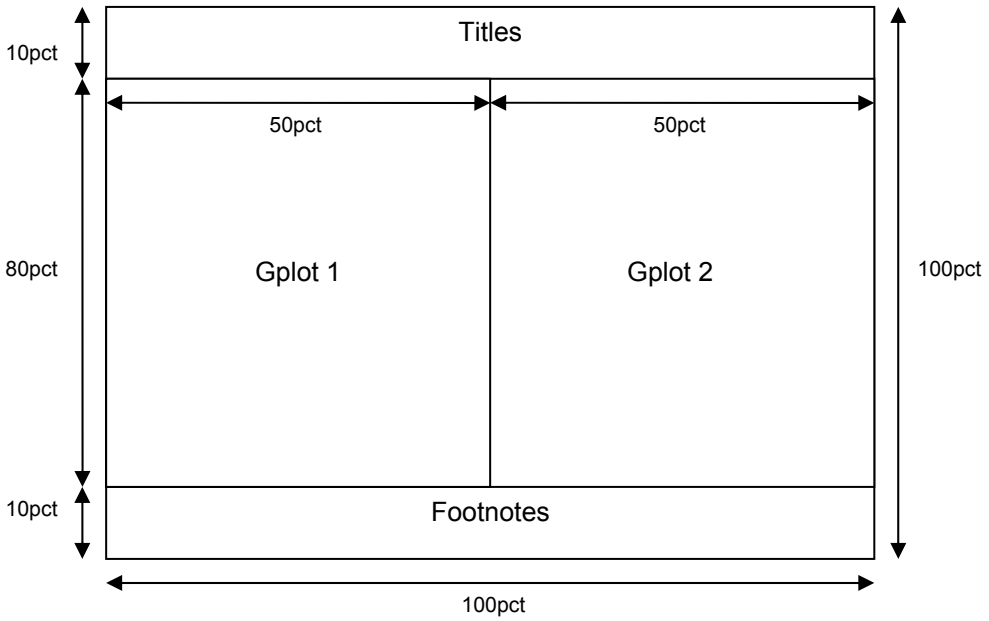


USING GOPTIONS TO CONTROL THE DIMENTIONS OF THE FIGURE

Before we create any figures we need to know the required dimensions (width and height). To avoid stretching we need to ensure:

- The dimensions of the gslide are the required width and height of the combined figure.
- The two gplots have the required dimensions so they are not stretched by the greplay procedure when they are placed into the template.

The dimensions in percentages for the template we need to create are shown in a figure on the next page.



The required width and height of gplot figures 1 and 2 can be calculated once the required width and height of the combined figure is known. In this example the required width is 24.5cm and the required height is 14.5cm. So we can calculate the width and height of gplot figures 1 and 2 as follows:

The width of gplot figures 1 and 2 = 0.5 * 24.5cm = 12.25cm.
The height of gplot figures 1 and 2 = 0.8 * 14.5cm = 11.6cm.

The relevant goption to control the width of the figure is xmax and the relevant goption to control the height of the figures is ymax. It is also a good idea to define the number of pixels using xpixels and ypixels respectively. Choosing numbers with the same ratio as xmax to ymax makes this simple. The code to correctly set up the goptions to create the gslide is shown below.

```
goptions xmax    = 24.5cm
         xpixels = 2450
         ymax    = 14.5cm
         ypixels = 1450;
```

It is important to change these before creating the gplot figures to ensure when they are used in the greplay procedure they are not stretched. The code to correctly set up the goptions for gplot figures 1 and 2 is shown below.

```
goptions xmax    = 12.25cm
         xpixels = 1225
         ymax    = 11.6cm
         ypixels = 1160;
```

OTHER IMPORTANT GOPTIONS

There are other goptions which are required throughout the whole of the session. The code for these is shown below.

```
goptions device  = emf
         htext   = 0.35cm
         ftext   = simplex;
```

The device option is used to specify the graphics device. This example uses the emf device (Enhanced Windows Metafile) which will produce a emf file containing the combined figure. The choice of device will affect the look of the figure. With some devices figures can look better electronically but worse when printed out and visa-versa. There are generally two different ways the information to produce a figure are stored: as instructions to draw the elements in the

figure, or as a static representation of the final image. An emf is an example of the former, whereas a jpeg is an example of the latter. The advantage of a file type that stores the information as a set of instructions to draw the figure rather than a static image is that when the figure is resized content such as fonts are increased in size so they retain good detail levels rather than the image becoming blocky.

The htext option is also used. It is a good idea to specify the text height in the goptions statement to save yourself having to specify it every time you create text e.g. in symbol statements, axis statements, etc. The font is specified with the ftext option.

HOW IS TEXT SIZE AFFECTED WHEN THE YMAX GOPTION IS ALTERED?

The size of SAS text is controlled by specifying its height in units. The default SAS unit for specifying text height is cells. An important property of the cell unit is that heights are in relation to the height of the figure (ymax). In our example, if a text size of 1 cell is used to create both the gslide and gplot figures 1 and 2, the size of the text in gplot figures 1 and 2 will be 80 percent of the size of the text in the gslide. The easiest way to avoid this is to use an absolute unit such as cm or in. Alternatively you can calculate the required text size in cells for gplot figures 1 and 2 so the text size will be 1 cell in the combined figure. Since the height of gplot figures 1 and 2 are 80 percent of the gslide,

$1 \div 0.8 = 1.25$.

So a text size of 1.25 cells for gplot figures 1 and 2 will appear as a text size of 1 cell in the combined figure.

CREATION THE FIGURES TO BE GREPLAYED

The SAS/GRAPH procedure proc gplot has been used to create the two figures. Some additional options have been used to enhance the look of the gplots:

- Minor tick marks removed.
- The same format for the tick marks on the vertical axes has been specified in both figures; this is an easy way to ensure the length of the horizontal axis is identical for both figures.
- Symbol statements are used to define symbols, symbol colours and specify interpolation lines.
- Axis statements are used to rotate the orientation of the vertical axis label so it is vertical, specify offsets for the axis so that symbols fit within the graph area and to reduce the height of the axis slightly so the full figure will display correctly using the length option.
- A legend statement is used to place the legend within the graph area, which creates more space for the plot. The shape of the legend has also been manipulated so only one symbol is displayed next to the legend label rather than three.

The code is shown in Appendix A.

It is worth noting that SAS has a problem of correctly calculating the required length of the vertical axis for the graph to fit within the defined area. When this occurs the following warning is printed to the log:
WARNING: The left vertical axis labeled Height was too large as specified. Specify LENGTH=XX.X PERCENT.
This can be corrected by specifying the length of the axis to be less than or equal to the length specified.

SAS CATALOGUES AND FIGURE NAMES

When the code is run, by default the figure will be stored in a catalogue called gseg in the work directory. Also by default the name of the first figure created by the gplot procedure will be gplot. The next figure created will be gplot1, then gplot2 and so on. You can use the gout= option in the gplot statement to specify the name of the SAS catalogue to store the figure. You can also use the name= option in the plot statement to change the name of the figure in the catalogue to a name of your choice up to 8 characters.

If a figure with the same name already exists within the catalogue it will not be overwritten but incremented with a 1 at the end, then a two at the end and so on. When developing code in SAS interactively it is worth being aware of this when changing gplot code and re-running a greplay procedure again because if the old figure is not deleted from the catalogue the new versions of the gplot figure will not be used in the greplay because it will have a different name in the catalogue. The contents of the catalogue can be deleted using the following code.

```
proc datasets lib=work nolist memtype=cat;
  delete <catalogue-name(s)>;
quit;
```

It is a good idea to delete the contents of the catalogue at the end of your code (like it is a good idea to delete work datasets at the end of your code).

CREATION OF TITLES AND FOOTNOTES USING PROC GSLIDE

Titles and footnotes should be presented consistently with the titles and footnotes in accompanying listings and tables. They also need to span across the whole of the graph area. A simple way to do this for the greplay figure is using the SAS/GRAPH procedure proc gslide. Title and footnote statements just need to be included within the proc gslide. The code for our figure is shown below.

```
proc gslide;
  title1 j=l "PhUSE 2011";
  title2 j=l "Figure 1";
  title3 j=l "Plot of height against age and weight against age";

  footnote1 j=l "Author: David Mottershead";
run;
quit;
```

USING THE GREPLAY PROCEDURE TO CREATE A TEMPLATE

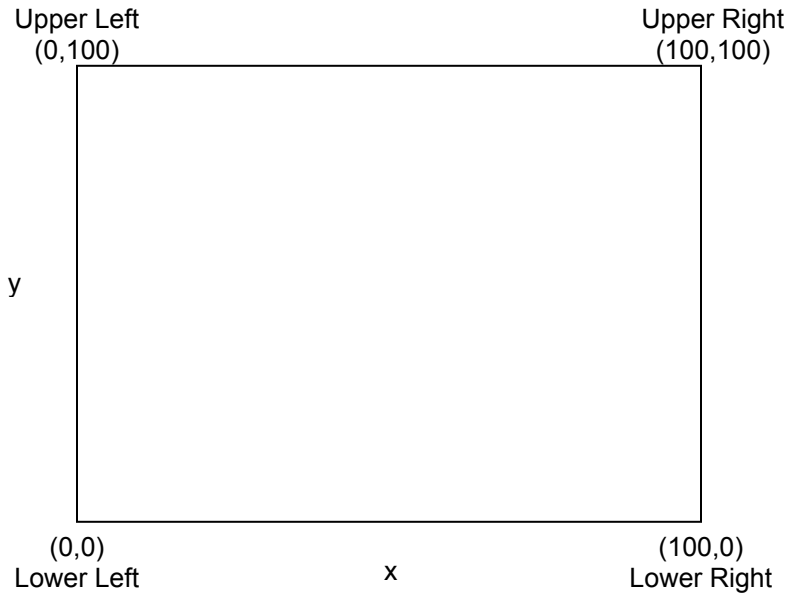
Proc greplay is a SAS/GRAPH procedure which can be used to combine multiple figures. There are some pre-defined templates for simple combinations e.g. 2x1, 2x2, etc. Custom templates can also be created to allow you to place figures exactly where you want. To create our figure we require this flexibility.

The syntax to set up our template is as follows:

```
proc greplay tc=work.tempcat nofs;
  tdef newtemp des='Two panel template'
```

The tc= option defines the template catalogue where the template will be stored. The nofs option stops windows being used in interactive SAS. The tdef statement is used to define the new template which is called newtemp in our example. The description option associates a description with the template, which is useful if you are creating permanent template catalogues. In the same tdef statement we now must define our template.

The template is created by defining the coordinates of the 4 corners where each figure will be placed. The coordinates of the whole greplay area are shown in the below figure.



For each figure within the template you need to specify 8 numeric values which are the x and y values for the four pairs of coordinates. These are ulx (upper left x), uly (upper left y), urx (upper right x), ury (upper right y), llx (lower left

x), lly (lower left y), lrx (lower right x) and lry (lower right y).

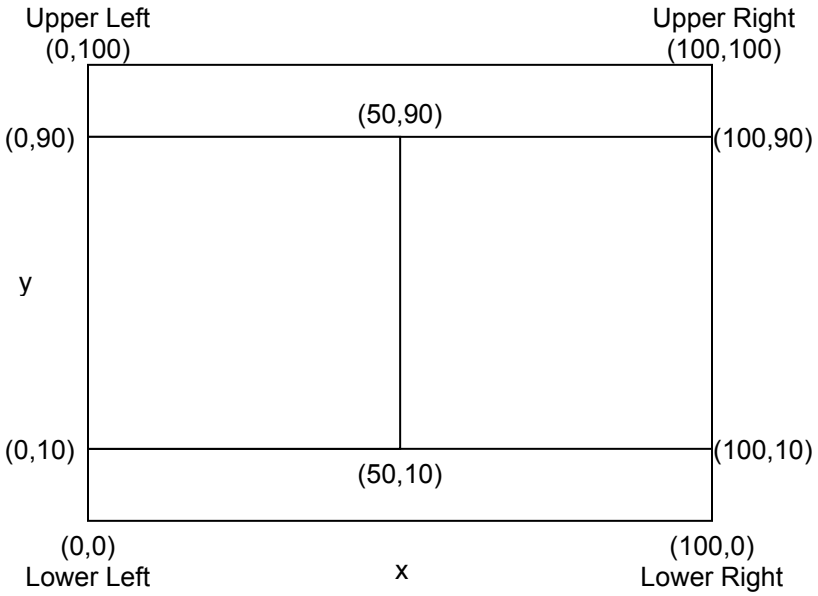
The code to create the template for the gslide (1) and gplot figures 1 and 2 (2 and 3 respectively) is shown below:

```
1/ulx=0   uly=100
  urx=100 ury=100
  llx=0   lly=0
  lrx=100 lry=0

2/ulx=0   uly=90
  urx=50  ury=90
  llx=0   lly=10
  lrx=50  lry=10

3/ulx=50  uly=90
  urx=100 ury=90
  llx=50  lly=10
  lrx=100 lry=10
```

This is visualized in the below figure.



USING THE GREPLAY PROCEDURE TO CREATE THE FINAL FIGURE

We can now create the combined figure using greplay. We need to set up the filename and goptions to save our figure to an external location. Example code is shown below.

```
filename gsasfile "C:\temp\Combined Figure.emf";

goptions gsfname = gsasfile
         gsfmode = replace;
```

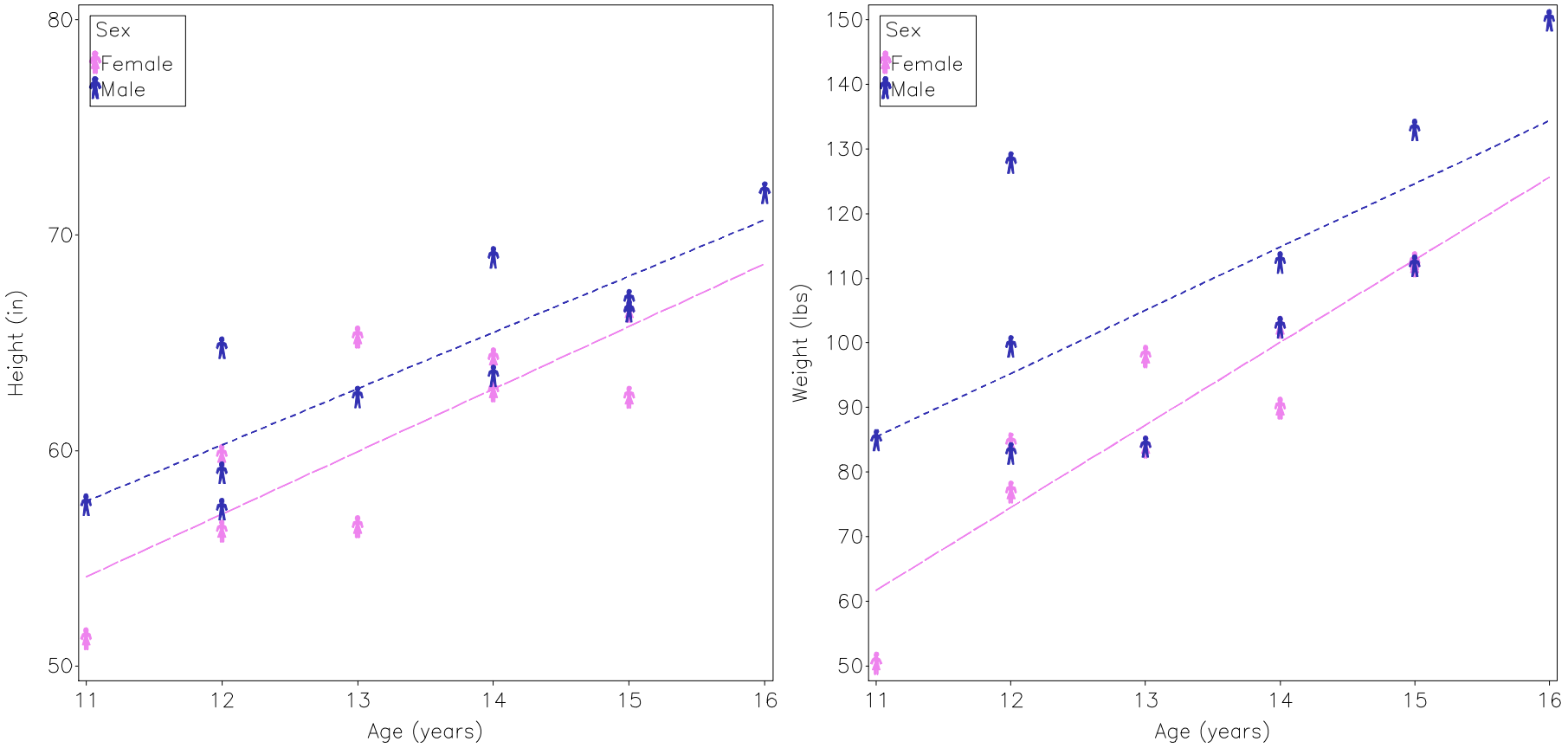
The filename statement specifies where the figure will be stored. The goption gsfname associates the filename with the figure. The option gsfmode = replace is used so each time a figure is created it will be replace by the new one.

Proc greplay is used again to create the combined figure. We specify the same template catalogue and nofs options. The template statement is used to specify the name of the template we want to use. The igout statement specifies the catalogue where we want the created figure to be stored. The treplay statement is used to specify figures from the catalogue which we want to place into the different template positions.

```
proc greplay tc=work.tempcat nofs;
  template newtemp;
  igout work.tempcat;
  treplay 1:GSLIDE
          2:FIGURE1
          3:FIGURE2;
run;
quit;
```

The final combined figure is shown on the next page.

PhUSE 2011
Figure 1
Plot of height against age and weight against age



Author: David Mottershead

APPENDIX A: SAS CODE

```
** Set up goptions - to be used for all figures **;
goptions device  = emf
         htext   = 0.35cm
         ftext   = simplex;

** Set up goptions to control size of figure **;
goptions xmax    = 12.25cm
         xpixels = 1225
         ymax    = 11.6cm
         ypixels = 1160;


**------------------------------------------------------------------------------**;
** Create gplot figures                                                         **;
**------------------------------------------------------------------------------**;
proc gplot data=sashelp.class gout=work.tempcat;
  plot height*age=sex / vaxis=axis1 haxis=axis2 vminor=0 hminor=0 legend=legend1
                        name="FIGURE1";

  ** Specify format for height so that both figures have the same width of **;
  ** vertical tick marks **;
  format height 3.;

  ** Define symbol statements **;
  symbol1 font=marker value=R color=violet i=reg line=5  width=2 h=0.4cm;
  symbol2 font=marker value=Q color=bib    i=reg line=20 width=2 h=0.4cm;

  ** Define axis options **;
  axis1 label=(a=90 "Height (in)") offset=(2pct) length=91pct;
  axis2 label=("Age (years)")      offset=(1pct);

  ** Define legned options **;
  legend1 label=(position=top "Sex") frame shape=symbol(0.001,0.4cm)
          value=(j=l "Female" "Male") mode=protect position=(top left inside)
          offset=(1.5pct,-1.5pct) down=2;
run;

proc gplot data=sashelp.class gout=work.tempcat;
  plot weight*age=sex / vaxis = axis1 haxis=axis2 vminor=0 hminor=0 legend = legend1
                        name="FIGURE2";

  ** Specify format for weight so that both figures have the same width of **;
  ** vertical tick marks **;
  format weight 3.;

  ** Symbol statements **;
  symbol1 font=marker value=R color=violet i=reg line=5  width=2 h=0.4cm;
  symbol2 font=marker value=Q color=bib    i=reg line=20 width=2 h=0.4cm;

  ** Define axis options **;
  axis1 label=(a=90 "Weight (lbs)") offset=(2pct) length=91pct;
  axis2 label=("Age (years)")       offset=(1pct);

  ** Define legned options **;
  legend1 label=(position=top "Sex") frame shape=symbol(0.001,0.4cm)
          value=(j=l "Female" "Male") mode=protect position=(top left inside)
          offset=(1.5pct,-1.5pct) down=2;
run;
quit;
```

```
**------------------------------------------------------------------------------**;
** Create gslide                                                                **;
**------------------------------------------------------------------------------**;
** Setup goptions for gslide **;
goptions xmax    = 24.5cm
         xpixels = 2450
         ymax    = 14.5cm
         ypixels = 1450;

** Create gslide containing titles and footnotes **;
proc gslide name="GSLIDE" gout=work.tempcat;
  title1 j=l "PhUSE 2011";
  title2 j=l "Figure 1";
  title3 j=l "Plot of height against age and weight against age";

  footnote1 j=l "Author: David Mottershead";
run;
quit;


**------------------------------------------------------------------------------**;
** Create greplay template                                                      **;
**------------------------------------------------------------------------------**;
proc greplay tc=work.tempcat nofs;
  tdef newtemp des='Two panel template'

  1/ulx=0   uly=100
    urx=100 ury=100
    llx=0   lly=0
    lrx=100 lry=0

  2/ulx=0   uly=90
    urx=50  ury=90
    llx=0   lly=10
    lrx=50  lry=10

  3/ulx=50  uly=90
    urx=100 ury=90
    llx=50  lly=10
    lrx=100 lry=10;
run;
quit;

** Set up goptions to save the combined figure to an external location **;
filename gsasfile "J:\Programming\Private\Personal Folders\David Mottershead\Papers\
figures\emf\PhUSE Figure.emf";

goptions gsfname = gsasfile
         gsfmode = replace;


**------------------------------------------------------------------------------**;
** Create final combined figure using greplay                                   **;
**------------------------------------------------------------------------------**;
proc greplay tc=work.tempcat nofs;
  template newtemp;
  igout work.tempcat;
  treplay 1:GSLIDE
          2:FIGURE1
          3:FIGURE2;
run;
quit;
```

CONCLUSION

I believe a SAS programmer should always aim to add value to the Tables/Figures/Listings (TFL) reporting process by making TFLs as well presented as possible so that the information is the focus. This includes not overcomplicating the TFL so it is unnecessarily fancy yet not under-formatting it or being lazy. I think we need to present our TFLs in the best possible way and to take pride in achieving an optimal TFL.

Since I have known how to use greplay I have heard people complain about the problems of stretching. I have seen this problem on many separate occasions for multiple clients. During my research for this paper I also found it in a previous PhUSE paper[1].

Upon researching similar papers which have already been written I found a paper[2] containing the required technique to solve the problem, although this was not the focus of the paper. I also found a novel solution based on changing the origin of the plot and axis lengths so that SAS does the scaling for you[3].

I think it is important that the solution in this paper is documented. To quote the last paper I referenced:

“ creating one page of output with multiple graphs can be very tedious.” “ graphs are scaled down to fit in the cookie cutter sections of the template. The scaling of the graphs can result in illegible fonts and problems with the lengths of the axis.”

I think the techniques described here will provide a useful reference for solving this common problem when using greplay.

REFERENCES

1. Automating GREPLAY to display multiple Graphs per page, KarolAnne Fitzpatrick and Erika Daly, Paper IS03, PhUSE 2007.
2. Integrating Multiple SAS/GRAPH® Procedures to Generate a Custom Image, Joshua M. Horstman, Paper 121-2010, MWSUG 2010.
3. Multiple Graphs on One Page Using GREPLAY with “100%-Templates, Dirk Spruck, Paper CC09, PharmaSUG 2009.

CONTACT INFORMATION

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

David J Mottershead
Quanticate Ltd., Airfield House
Kingston Bagpuize Business Park
Abingdon
OX13 5FE

Email: david.mottershead@quanticate.com
Web: http://uk.linkedin.com/in/davidjmottershead