

Creating High Quality Statistical Graphs for Publications

Kriss Harris, SAS Specialists Ltd, Hertfordshire, United Kingdom

ABSTRACT

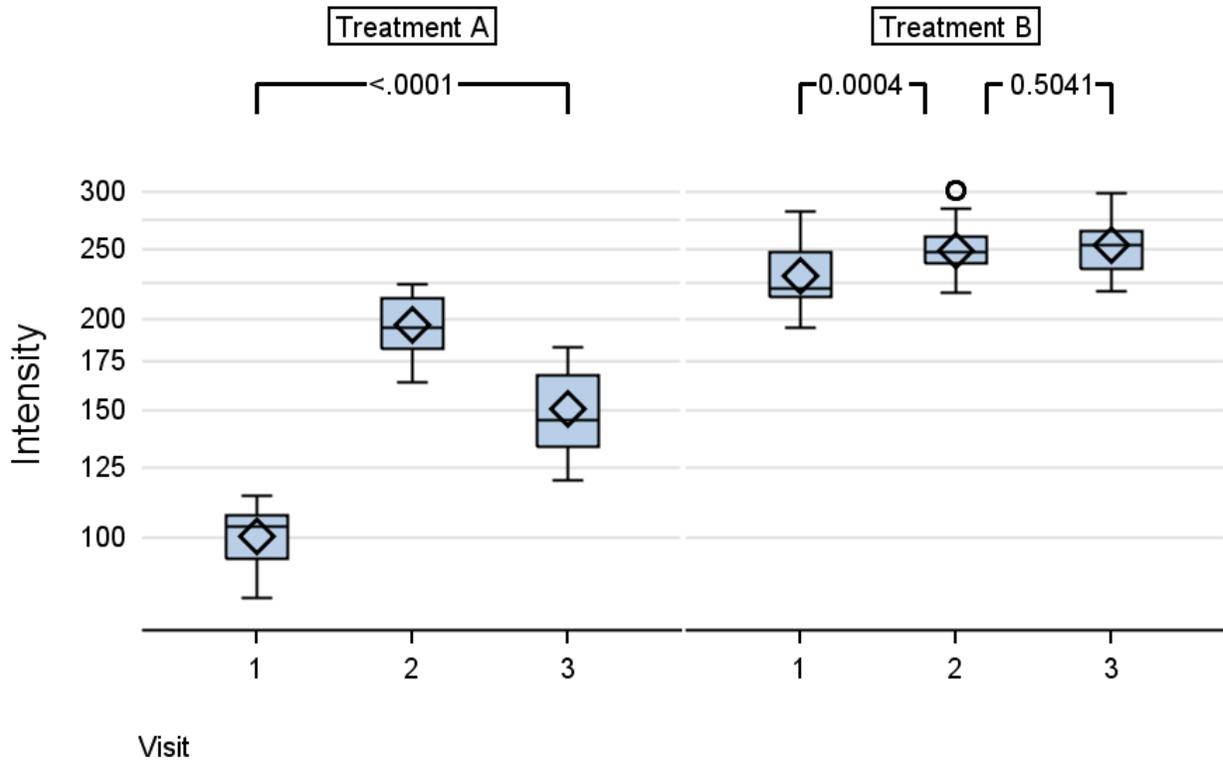
Do you want to produce high quality graphs for publications or presentations? Do you want to add p-values on the graphs with annotations that show which groups are being compared? Do you want to learn more about the DPI option and how to use it? If you answered yes to any of the questions or have an interest in SAS Graphics then this paper is for you. This paper will demonstrate how to do the above plus show how to create first-rate Kaplan Meier Graphs, and Forest Plots using SAS® 9.2.

INTRODUCTION

ADDING P-VALUES TO FIGURES

Figure 1 Figure of a Lab Result by Visit and Treatment Group with Annotated p -values

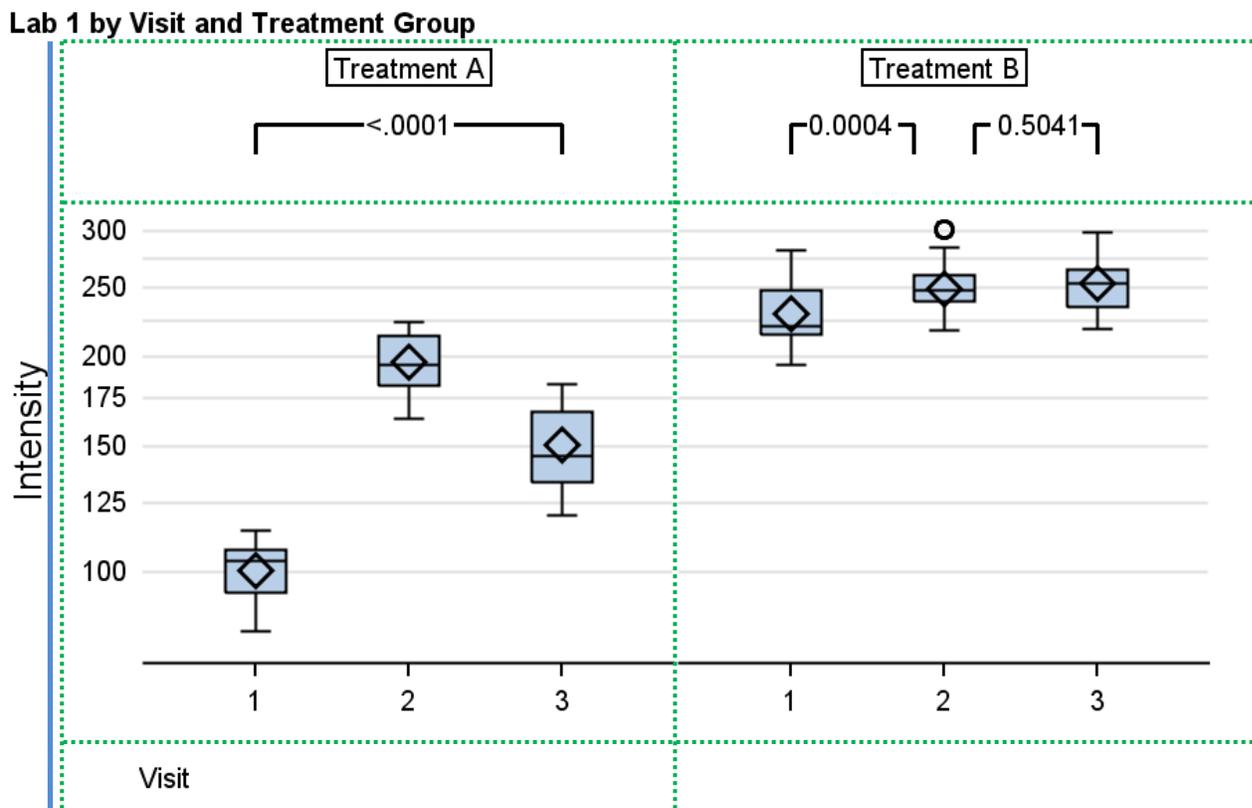
Lab 1 by Visit and Treatment Group



The plot above was created by the LAYOUT LATTICE with 2 columns and 3 rows. An easier way to think of Figure 1 is in terms of the Figure 2 below, where the plot is made up of 6 separate cells that have come together denoted by the green dotted line, and a row header that separates the "Intensity" title denoted by the blue solid line.

I am assuming that you are already familiar with producing box plots within Graph Template Language and the more creative part in this figure is highlighting the comparisons of interest and displaying the related p -value. I have always wondered how this was implemented within publication graphics, and now I know this can be done with the ENTRY and VECTORPLOT statements within Graph Template Language.

Figure 2 Figure of a Lab Result highlighting rows and columns



Concentrating on the second row and the first column of Figure 2 above, we see that the x-axis ranges from 1 to 3 and the intensity values approximately range from 1 to 250. The key with the VECTORPLOT statement is to firstly think of coordinates where you want the vectors to start from and secondly where you want them to end, and in the example above I did not want the vectors i.e. the p -value annotation to come into contact with any of the y-axis values associated with the box plot. Therefore an exaggerated value of at least 100,000 was used in the code below for the vectors y-axis start and end point coordinates.

```

data dummy_data_with_locations;
set dummy_data_transposed;

x11 = 1;
x12 = 1.7;
x13 = 2.3;
x14 = 3;
y1 = 100005;

/* Setting the scale for the y2 axis to be between 100,000 and 100,010 */
uniform = rand("uniform");
y2scale = (100010-100000)*uniform+100000;
run;

```

We will now focus on how the VECTORPLOT statement was used to add p -value annotations in the first row and column of figure 2. In the Graph Template Language below there are four VECTORPLOT statements, the first one starts at the coordinates (1, 100000) which are the values entered for the XORIGIN and YORIGIN arguments, and this vector stops at the coordinates (1, 100005) which are the values entered for the X and Y arguments which denote the vector endpoints. The vector endpoint values typically have to be selected from values that are entered into a variable. Based on the vector start and endpoint, we can work out that this vector has travelled vertically upwards for 5 units. The ARROWHEADS = false option is used so that the arrowhead is not displayed at the end of the vector, and the YAXIS = Y2 is used to specify that the vector is mapped to the secondary y-axis, in other words the vector is mapped to a separate axis from the axis associated with the box plot. The second, third and fourth VECTORPLOT statements are then used to finish off the annotation. The second and third travel horizontally right on the x-axis, and a space of 0.6 units is between the end of the second vector and start of the third vector. This is so the p -value can be displayed with the ENTRY statement in between the two vectors. The fourth VECTORPLOT statement travels downwards on the y-axis.

It was just by chance, especially for the statement "ENTRY HALIGN = CENTER_BYVAL2;" that the display of the p -value was perfectly aligned both horizontally and vertically in the figure, when used in combination with the VECTORPLOT statements. Generally, when text is not perfectly aligned, then the PAD options in the ENTRY statement can be used to shift the text left, right, up or down.

Below is an extract of the code that was to produce Figures 1 and 2.

```

layout lattice / columns=2 rows = 3 rowweights = (0.15 0.75 0.1) rowdatarange =
union;

  rowaxes;
    rowaxis / display=NONE type = linear ;
    rowaxis / display=(tickvalues LABEL) type = log griddisplay=on label =
      "Intensity" ;
  endrowaxes;

  cell;
    cellheader;
      entry "Treatment A" / border =true;
    endcellheader;
    layout overlay / yaxisopts = (display = NONE type = linear) xaxisopts =
      (display = NONE) y2axisopts = (display = NONE);
      SCATTERPLOT X = visit Y = y2scale / markerattrs =(size = 0) YAXIS=
        Y2; /* Setting up the scale */

      vectorplot y=y1 x=x11 xorigin=x11 yorigin=100000 / ARROWHEADS=false
        yaxis = y2; /* P-value annotation */

      vectorplot y=y1 x=x12 xorigin=x11 yorigin=100005 / ARROWHEADS=false
        yaxis = y2;

      vectorplot y=y1 x=x14 xorigin=x13 yorigin=100005 / ARROWHEADS=false
        yaxis = y2;

      vectorplot y=y1 x=x14 xorigin=x14 yorigin=100000 / ARROWHEADS=false
        yaxis = y2;

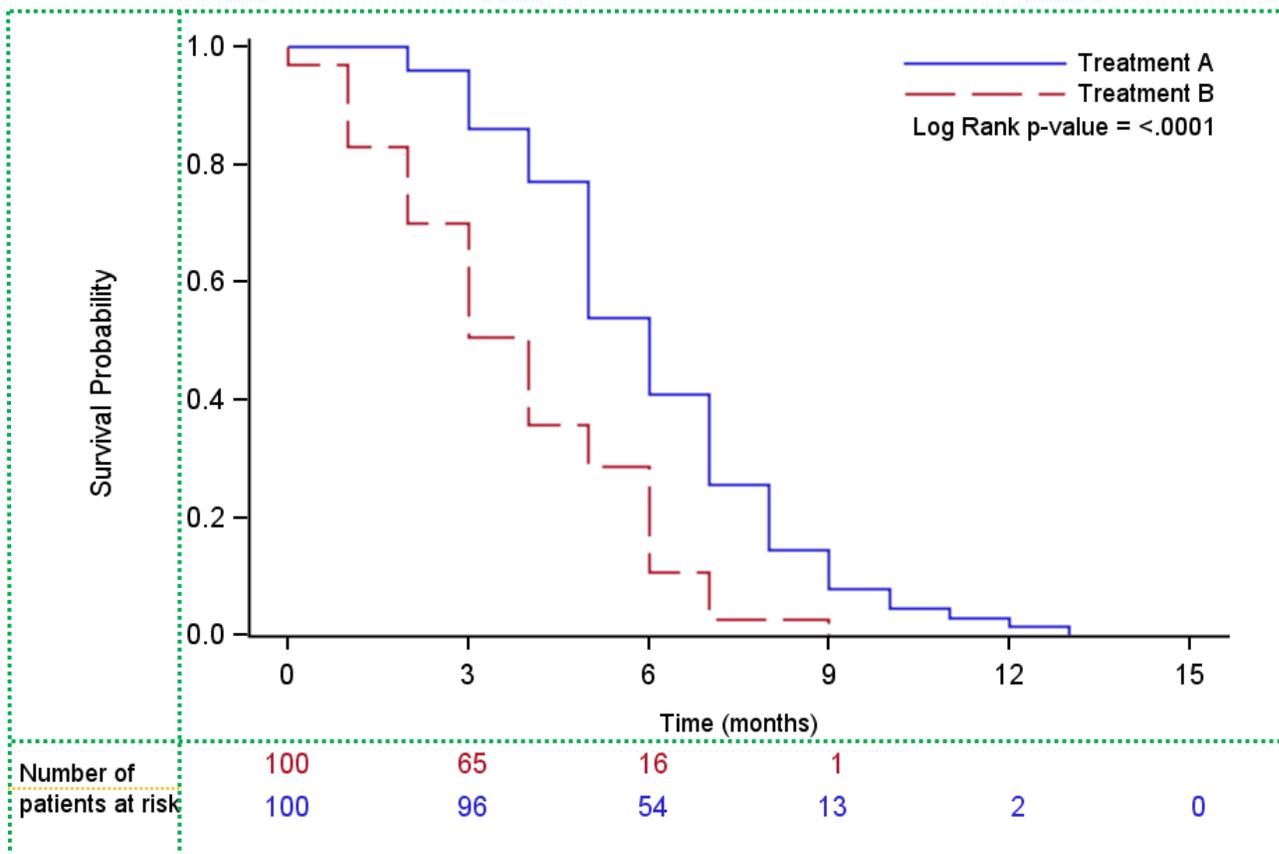
      entry halign = center _BYVAL2_;
    endlayout;
  endcell;

```

KAPLAN MEIER SURVIVAL CURVE

Firstly, a very decent survival curve can be generated within the LIFETEST procedure by using the PLOTS and the ODS GRAPHICS ON option. Figure 3 below was created instead of using the in-built survival curve because more customizations were needed, such as displaying the Log Rank p -value and producing the at risk table underneath the x-axis. Again as with Figure 2, it may be easier to visualize the plot in terms of separate cells since the plot was created using the LAYOUT LATTICE statement.

Figure 3 Kaplan Meier Survival Curve with Number of Patients at Risk



In Figure 3, the first row and the first column contains the title “Survival Probability” and this is created with the ENTRY statement and the ROTATE = 90 option. The first row and the second column host the survival curve, along with the legend and log rank p -value. The input dataset that was used for the survival curve were initially outputted by the ODS OUTPUT SURVIVALPLOT statement within the LIFETEST procedure. The survival curve is created with the STEPLOT statement and the legend is created with the DISCRETELEGEND statement. Again the ENTRY statement is used to position the desirable text in the right place, and this time it is used to display the log rank p -value which is assigned as a dynamic macro in this example, and the result is extracted from the ODS table HOMTESTS.

A GRIDDED layout is used to nest and display “Number of patients at risk” over two rows, within the second row of the first column and the patients at risk table uses the ODS OUTPUT SURVIVALPLOT dataset and is created in the second row of the second column with the SCATTERPLOT statement along with the MARKERCHARACTER option. The Graph Template Language code that was used to create Figure 3 is below.

```

layout lattice /rows = 2 columns = 2 columnweights = (0.14 0.86) rowweights =
(0.9 0.1) rowgutter = 4px columngutter = 0px;

layout overlay;
  entry "Survival Probability" / valign = center ROTATE=90;
endlayout;

layout overlay / xaxisopts=( Label="Time (months)" labelattrs =(size = 5pt)
type=linear linearopts=( tickvaluelist=( 0 3 6 9 12 15 ) viewmin=0
viewmax=15 ) ) yaxisopts=( display=(LINE TICKS TICKVALUES) offsetmin=0
type=linear linearopts=( viewmin=0 ) );
  StepPlot X=Time Y=Survival / primary=true Group=Stratum
  LegendLabel="Survival Probability" NAME="censored";

  DiscreteLegend "censored" / Location=Inside across=1 valign=top halign
= right border = false;

  Entry halign=right TEXTATTRS=(SIZE=5pt) "Log Rank p-value = " RANK /
  valign = top pad =(LEFT=0 RIGHT=2 TOP= 36 BOTTOM=0);
endlayout;

layout overlay;
  layout gridded / columns = 1 rows = 2;
  entry halign=left TEXTATTRS=(SIZE=5pt) "Number of" / pad =(LEFT=0
RIGHT=0 TOP=0 BOTTOM=0);
  entry halign=left TEXTATTRS=(SIZE=5pt) "patients at risk" / pad
=(LEFT=0 RIGHT=0 TOP=0 BOTTOM=0);
  endlayout;
endlayout;

layout overlay / xaxisopts=( display = NONE ) yaxisopts=( display = NONE
type=discrete );
  ScatterPlot X=tAtRisk Y=stratumchar / Group=stratumchar
  MarkerCharacter=AtRisk LegendLabel="stratumchar" NAME="SCATTER";
  endlayout;
endlayout;

```

PROC TEMPLATE AND DOTS PER INCHES

In Figures 1 to 3, I have set the image resolutions to equal 300 DPI (dots per inch) by specifying the options below.

```
ods listing image_dpi = 300;
```

SAS® recommend using 200 DPI for images copied and pasted to a Microsoft PowerPoint presentation or a Microsoft Word document, and graphs shown in SAS/STAT documentation are typically generated at 300 DPI for display in PDF. SAS® also recommend that you should create your graphs by using the exact size that is used to display the graphs in your paper or presentation, therefore the size I used to output the figures were HEIGHT = 2in and WIDTH = 3in. Finally I specified the size of the graph label, value and marker font by using PROC TEMPLATE because the scale of the default graph text and markers look large in comparison to the actual plot when the above DPI and HEIGHT and WIDTH are used, so these were scaled down by creating a style and then subsequently selecting it using the code below.

```

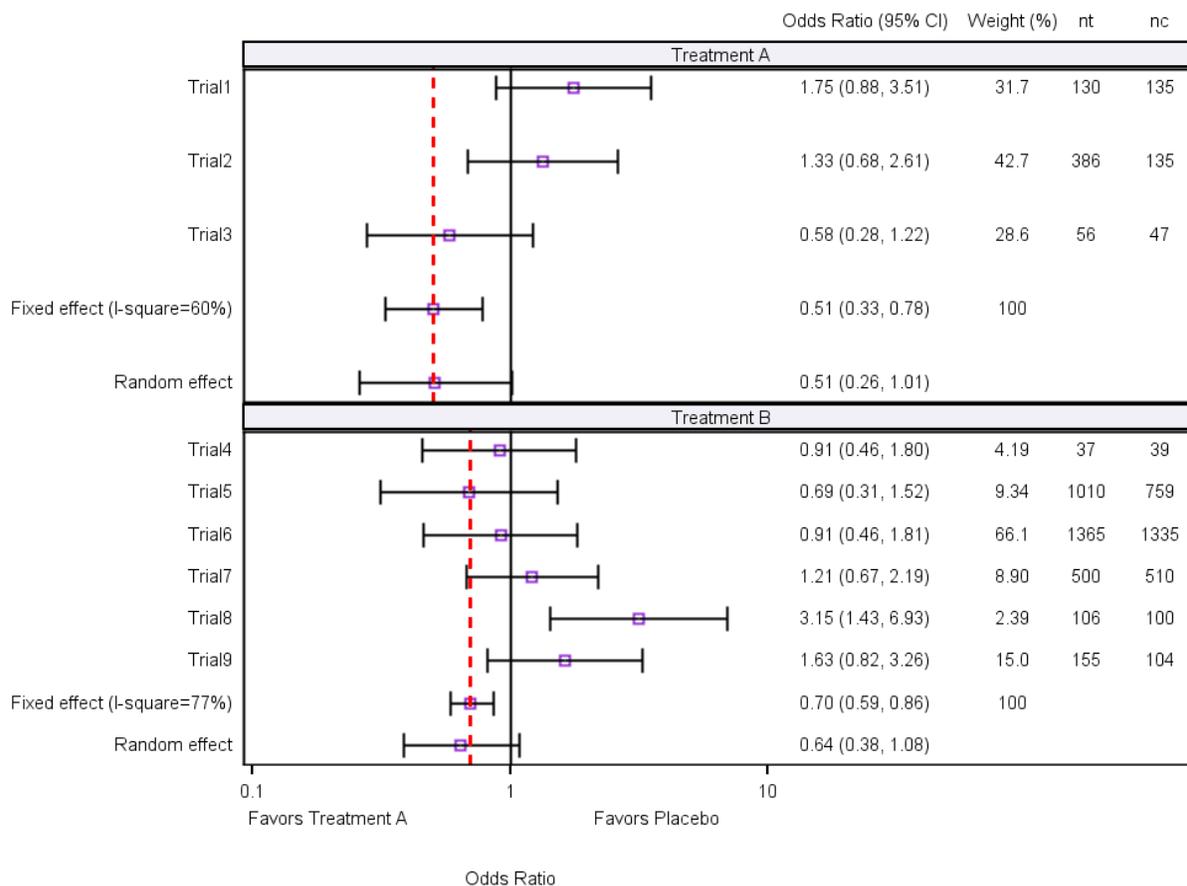
proc template;
  define style styles.TLSPlot;
    parent = Styles.listing;
    style GraphFonts from GraphFonts /
      'GraphValueFont' = ("

```

FOREST PLOT

There are a lot of great examples of how to produce a Forest Plot on the Graphically Speaking SAS® website and the SAS® Support website. The reason why the figure below is being presented is, at the time of creation, I could not find any help on producing a Forest Plot with these particular features. Figure 4 below is essentially a paneled Forest Plot with the odds ratio on a log scale, and a table of the descriptive statistics. Using the SGPLOT procedure a Forest Plot can be produced on the log scale along with a table of descriptive statistics, however it is not simple to extend this to a paneled plot using the SGPANEL procedure, and therefore Graph Template Language is used to produce Figure 4.

Figure 4 Forest Plot with Descriptive Statistics



When I initially worked on producing the figure above, the difficulties were producing a paneled Forest Plot on the log scale, whilst also displaying the odds ratio, weight, and number of subjects in the treatment and placebo descriptive statistics. I kept getting the warning "SCATTERPLOT statement has a conflict with the axis type. The plot will not be drawn." This was due to mixing different plot types. A workaround for this is to make the descriptive statistics have a numeric number, and again as in Figures 1 and 2, this number should not overlap the data on the chart. A format can then be used, to display the correct title, as in the example below.

```
proc format;
value xax 15 = "Odds Ratio (95% CI)"
          17 = "Weight (%)"
          18 = "nt"
          19 = "nc";
run;
```

The “Favors Treatment A” and “Favors Placebo” can be added onto the plot by using the familiar ENTRY statement that I mentioned in relation to the above figures. Generally the ENTRY statement can be used to align text in the position that you would like. In the Graph Template Language code which was used to create the Forest Plot below, you can see that the LAYOUT DATAPANEL has been used this time to create a plot paneled by treatment. The CLASSVARS=(TREATMENT) argument gives the instructions to panel the graph by treatment. Figure 4 essentially has the Forest Plot and descriptive statistics operating in different sections of the same graph, and the COLUMNAXISOPTS is used to set the properties for the Forest Plot column axis section and the COLUMN2AXISOPTS is used to specify the properties for the descriptive statistics column axis section.

The option TYPE = LOG is used in the COLUMNAXISOPTS to make the Forest Plot appear on the log scale. In the COLUMN2AXISOPTS the TICKVALUELIST=(15 17 18 19) is used to select only those numbers on the secondary axis in order for the graph to look neat, and as a result when combined with the generated format, the values will be substituted with "Odds Ratio (95% CI)", "Weight (%)", "nt", and "nc" respectively

```
layout datapanel classvars=(treatment) /
columns=1 rows=2 rowdatarange=union
headerlabeldisplay=value
headerbackgroundcolor=GraphAltBlock:color
rowaxisopts=(display=(tickvalues))
columnaxisopts=(type=log logopts=(MINORTICKS = 1 base = 10 viewmin =
0.1 viewmax = 10 ) offsetmin=0.01 offsetmax=0.45 display=(ticks
tickvalues) label = "Odds Ratio")
column2axisopts=(type = linear offsetmin=0.65 offsetmax=0.04
display=(tickvalues) type=linear linearopts=(TICKVALUELIST=(15 17 18
19)) );

layout prototype / cycleattrs=true;
scatterplot y = trialid x = or_ci_xaxis / markercharacter =
or_ci_compress xaxis = x2 primary=true;
scatterplot y = trialid x = weight_xaxis / markercharacter
= weightref xaxis = x2;
scatterplot y = trialid x = nt_xaxis / markercharacter =
nt_char xaxis = x2;
scatterplot y = trialid x = nc_xaxis / markercharacter =
nc_char xaxis = x2;
scatterplot y=trialid x=or / xerrorupper = UCL xerrorlower
= LCL ;
referenceline x= 1 / lineattrs = (color = black pattern =
1);
referenceline x = fixedref /lineattrs = (color = red
pattern = 2);
endlayout;

sidebar / align=bottom;
entry halign=left "Favors Treatment A" halign=right
"Favors Placebo" / valign=bottom
PAD=(LEFT=3 RIGHT=250 TOP=0 BOTTOM=0);
endsidebar;

sidebar / align=bottom;
entry halign=center "Odds Ratio" / valign=bottom PAD=(LEFT=0
RIGHT= 220 TOP=20 BOTTOM=0);
endsidebar;
endlayout;
```

CONCLUSION

SAS® 9.2 can be used to create stunning graphics for publications. These graphs often need to have a higher resolution and more complex; therefore thinking of a solution in a different way or using a work around in SAS® 9.2 is usually needed to obtain the desired graphic.

REFERENCES

- “Graph Size and Resolution”. Available at http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_odsgraph_sect034.htm
- Matange, Sanjay and Heath, Dan. 2011. “Statistical Graphics Procedures by Example: Effective Graphs Using SAS®”. Cary, NC: SAS Institute Inc.
- Mantage, Sanjay. “Forest Plots”. Available at <http://blogs.sas.com/content/graphicallspeaking/tag/forest-plot/>
- “SAS® 9.2 Graph Template Language Reference”. Available at <http://support.sas.com/documentation/cdl/en/grstatgraph/63878/HTML/default/viewer.htm#titlepage.htm>

ACKNOWLEDGMENTS

I would like to thank Sharon Carroll for the support she gave me whilst preparing the paper and I would like to thank Valerie Andre and Marie-Ange Paget for prompting me to create similar graphs.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the author to obtain the full code that was used at:

Name: Kriss Harris
Enterprise: SAS Specialists Ltd
E-mail: italjet125@yahoo.com
Web: <http://www.krissharris.co.uk>
Twitter: <https://twitter.com/krissharris>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.