

Visualising Adverse Events in 3D

Tim Palmer, Pfizer Global Research and Development, Sandwich, England

Jaya Ramakrishnan, Pfizer Global Research and Development, Sandwich, England

ABSTRACT

Adverse events are commonly reported through the use of conventional summary tables and listings. This paper looks at a novel method of presenting adverse event information graphically. These 'patient profile' type plots are produced in SAS® and aim to mimic 3D plots previously created within Excel presenting Subject Number, Time and Event Severity on the three axes. The paper describes the use of these plots within the company and uses a discussion on the development of the code to touch on a number of related SAS topics such as the SAS/GRAPH® annotate facility and colour-naming schemes.

INTRODUCTION

Adverse event data collected in early development studies are important in supporting a project clinician's assessment of safety. A clinician working in the Allergy and Respiratory Therapeutic Area prepared a 3-dimensional plot in Microsoft® Office Excel 2003 which he found useful in visualising the data. It helped emphasize trends and had a positive impact when used in presentations. An example is shown in figure 1 below.

Producing these plots, however, was time consuming for the clinician and inflexible so he approached the programming team to request that a more generic and efficient process be developed. The specification was to create plots that showed subject number, onset time post dose or duration of the adverse event and the severity of the adverse event. In addition each subject's data had to be well differentiated by colour.

PRODUCING THE PLOTS IN SAS

The challenge was to produce output in SAS that looked similar to the Excel chart in figure 1. This would allow the plot to be incorporated within our in-house reporting system. In this environment we also have scope to define macro parameters that help drive the code and give flexibility for producing output in a variety of circumstances.

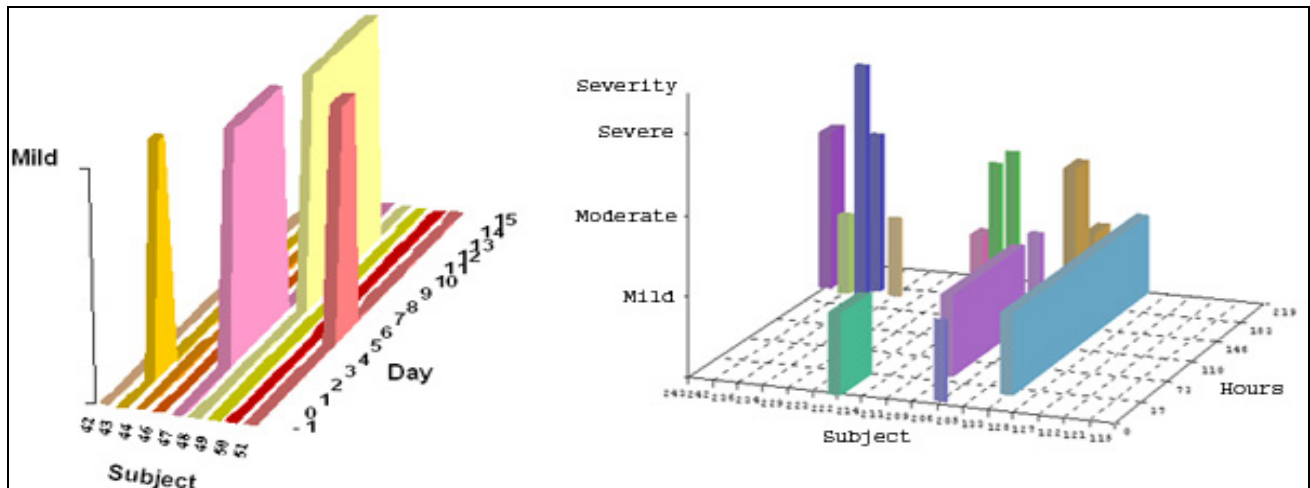


Figure 1 Original Microsoft Excel chart format, left, and example of SAS output, right

In describing the solution implemented, this paper touches on a number of SAS topics each of which are worthy of more detail than can be provided in a short paper of this kind. Further information on all these topics can be found in the references section.

PROC G3D

The need for three axes suggested that the SAS/GRAPH procedure PROC G3D would be suitable. This procedure can produce three-dimensional surface plots and scatter plots using the PLOT and SCATTER statements respectively. Although there are a variety of options one can use with this procedure, including using bars to show individual points, there was no simple method of producing the kind of *extended* bars that we needed.

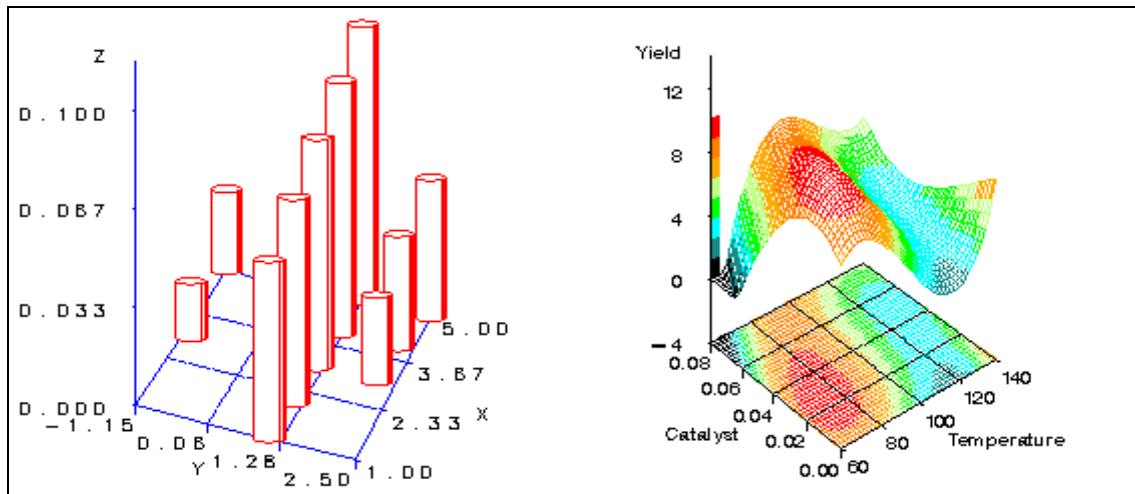


Figure 2 Sample outputs from PROC G3D (source sas.com)

The first example above shows that standard PROC G3D output can get *close* to the format we need but to get closer we would need to use the annotate facility.

THE ANNOTATE FACILITY

Part of SAS/GRAPH, the annotate facility gives you the power to add or change aspects of your graphical output. In the case of our adverse event plot we needed to use annotate to produce *all* the data points while using PROC G3D to provide a framework of correctly labelled axes that would put the annotated shapes into context.

The annotate facility makes use of an annotate dataset - a normal SAS dataset with certain standardly named variables - which when referenced within your SAS/GRAPH procedure is interpreted and converted into particular annotations to your final output.

You can choose to just add the odd box or text label but the real power of annotate comes when you use actual data to drive your annotations. The code snippet below shows a cut down version of how the extended bars required for our plot are produced using this principle.

```
data anno_ds;
  length function color maincol shadcol $8 style $15 text $2 symbol $1;
  retain xsys ysys zsys '2' size 0 line 1;
  set ae3;

  style='msolid'; color=maincol;
  function='poly';      x=aestart; y=subjid; z=0;      output;
  function='polycont';  x=aestart; y=subjid; z=aesev; output;
  function='polycont';  x=aestop; y=subjid; z=aesev; output;
  function='polycont';  x=aestop; y=subjid; z=0;      output;
run;
```

The dataset AE3 read in by this data step contains start and stop date/times and severity of all relevant adverse events. These values then drive the size and shape of the polygon created using the 'poly' (draw polygon) and 'polycont' (continue drawing the polygon) functions by assigning values from the study data to the relevant co-ordinate variable; x, y or z.

SAS COLOUR-NAMING SCHEMES AND COLOUR CYCLING

You may have noted in the code snippet above that the fill colour of the polygon has been set using a value contained in the variable maincol. Looking back to the original figure in Excel you will see that different subjects are required to

PhUSE 2006

be different colours. Also, when viewed together, the different bars should be coloured so as not to draw attention to any particular bar while also clearly distinguishing between neighbouring bars.

There are various colour-naming schemes that can be used in SAS version 8 to define colours. In summary, these are RGB (red, green, blue), CMYK (cyan, magenta, yellow, black), HLS (hue, lightness, saturation), HSV (hue, saturation, brightness), grayscale and predefined SAS colour names.

The scheme most suited to our requirement of easily producing suitably pleasing and relatively muted colours is the HLS scheme. Using this scheme it is easy to fix the lightness and saturation whilst modifying the hue to give a set of tonally consistent colours. We need to take a closer look at the HLS scheme to see how this is done.

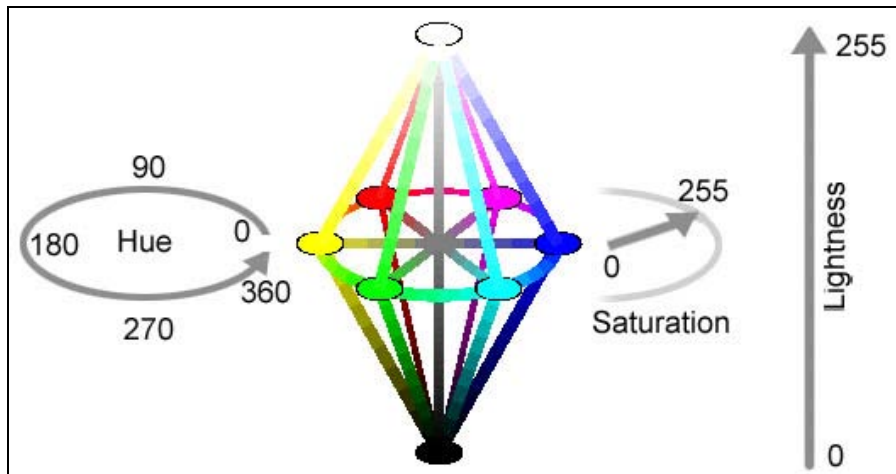


Figure 3 The HLS colour space maps as a double ended cone

The figure above attempts to explain the relationship between the three components – hue, lightness and saturation – of the HLS scheme. The scheme maps to a model in the shape of a double ended cone as illustrated with lightness running from bottom to top, saturation from the middle outwards and hue describing a circle counterclockwise from the right hand side of the figure.

The range of values that each of these parameters can take are shown as decimals. However SAS needs to have these numbers expressed in hexadecimal form so Hue takes values of 0 to 168 (360 in base 16), saturation and lightness 0 to FF (255 in base 16).

Now that we know a little more about the HLS scheme, how do we use this information in SAS? SAS requires colours in the HLS scheme to be defined in the form Hhhhlss where H indicates the HLS scheme, hhh indicates the hue, ll the lightness and ss the saturation. For example, bright blue is specified as H14066FF. As noted, each parameter is expressed in hexadecimal format.

So the data step code snippet below shows how we build up a colour definition for each subject :

```
...
if (subjnum/2) = floor(subjnum/2) then
  color = 'H' || trim(left(put(subjnum*20,hex3.))) || '80' || '70';
else color = 'H' || trim(left(put((180+subjnum*20),hex3.))) || '95' || '70';
...
```

The first line tests whether the subject number (variable SUBJNUM takes values 1, 2, 3, ...) is odd or even thus allowing alternate subjects to have distinctive colours. The following two lines define the colour using the format given above. In each case the lightness and saturation have been given constant values as determined following some experimentation. The value of hue is dependent on the subject number and whether that number is odd or even. The resulting colours generated for the first eight subjects are shown in figure 4 below.



Figure 4 Colours of bars for the first eight subjects

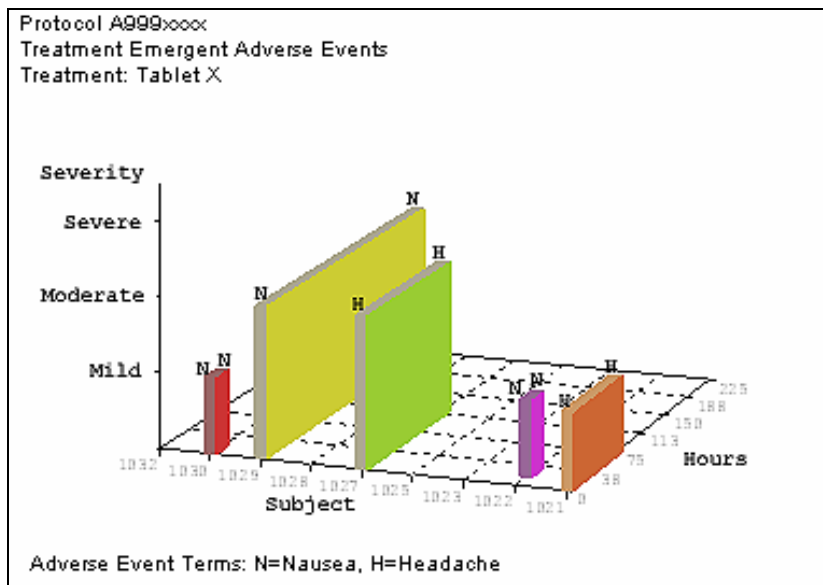


Figure 5 Example of 3D Adverse Event Plot

CONCLUSION

As noted in the introduction these plots are designed primarily for use in early development studies where we have small populations. No more than around 11 subjects can comfortably be included on each plot while still allowing the subject identifier to be used as a label on the y-axis.

Despite this restriction there is enough flexibility to allow the code to be used in a variety of circumstances. Macro parameters can be set in our reporting environment to specify adverse events of interest. Each type of adverse event is then identified on the plot through an annotated symbol on each extended bar, e.g. an 'H' to indicate headache and so on. Custom time periods of interest can also be set. Adverse event subsets such as, for example, all treatment related are also available.

This visualisation of adverse event data has proved to be of great value to clinicians and colleagues in the Allergy and Respiratory team. The positive response should lead to it being rolled out to other therapeutic areas. The use of the annotate facility and sympathetic treatment of colour in the plots creates informative and attractive outputs which can be produced from within our validated reporting environment using SAS software.

REFERENCES

- Bessler, Leroy (2004), *Communication-Effective Use of Color for Web-Pages, Graphs, Tables, Maps, Text, and Print*, Proceedings of the Twenty-Ninth SAS® User Group International Conference, Cary, NC: SAS Institute Inc., paper #176
- Mason, Phil (2004), *Introduction to SAS/GRAPH*, Proceedings of the Twenty-Ninth SAS® User Group International Conference, Cary, NC: SAS Institute Inc., paper #250
- Mink, David and Pasta, David (2006), *Improving Your Graphics Using SAS/GRAPH® Annotate Facility*, Proceedings of the Thirty-First SAS® User Group International Conference, Cary, NC: SAS Institute Inc., paper #085
- SAS Institute Inc. (1999), *Color Naming Schemes*, SAS OnlineDoc v8, Cary, NC: SAS Institute
<<http://v8doc.sas.com/sashtml/gref/zgscheme.htm>>
- SAS Institute Inc. (1999), *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute.
- Watts, Perry (2004), *Advanced Programming Techniques for Working with Color in SAS® Software*, Proceedings of the Twenty-Ninth SAS® User Group International Conference, Cary, NC: SAS Institute Inc., paper #091
- Watts, Perry (2003), *Working with RGB and HLS Color Coding Systems in SAS® Software*, Proceedings of the Twenty-Eighth SAS® User Group International Conference, Cary, NC: SAS Institute Inc., paper #136

PhUSE 2006

CONTACT INFORMATION

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

Tim Palmer, Pfizer Global Research and Development

Email: tim.palmer@pfizer.com

Jaya Ramakrishnan, Pfizer Global Research and Development

Email: jaya.ramakrishnan@pfizer.com

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

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