Paper DV04

Don't Dream It, Be It
Harnessing SAS GTL to Produce Customized Composite Graphics

Nev Cope, Standards, Automation and Specialist Programming, Development Operations, R&D, AstraZeneca, Cambridge, UK

ABSTRACT

In some areas of the industry there is still a perception that SAS® cannot produce high quality graphics. This is, of course, untrue!
There are a number of SAS Statistical Graphics procedures (`SGPLOT`, `SGCHART`, etc) that use the Output Delivery System (ODS) to create very detailed and high-quality images. However, these tend to output a single plot, and adding extra information or customisations can sometimes be cumbersome.

An alternative is to use the SAS Graphic Template Language (GTL), which allows users to overlay multiple figures to produce composite plots, and then combine multiple plots onto a single page. This allows a large amount of data to be presented in any number of creative ways.
This paper will demonstrate how to create complex plots quickly and easily, to a high resolution and visual standard for both study monitoring and presentation purposes, using SAS GTL and a bit of imagination.

INTRODUCTION

There are many elements that make a good graphic and many books and articles have already been written on the subject (See the Recommended Reading section). However, this paper is not intended as a discussion on these artistic or conceptual topics. Instead it will be focusing on demonstrating different methods, tips and tricks for producing high quality graphics from SAS in an effort to serve as a good reference and/or starting point for your projects, and to hopefully help you to expand your SAS Graphics toolbox.

Having said that, we may need to take a moment to briefly look at one of the main points that made me write this paper in the first place: The perception that SAS cannot produce high quality graphics.

This is a line that I have heard several times in my long career as a SAS programmer, and typically comes from study teams after reviewing a set of outputs that they have either requested from their programmers who aren't well versed in SAS Graphical syntax, or that were produced by a set of "standard" programs that are typically copied and pasted from study to study, year after year. In some cases, the graphics may look simplistic or dated simply because the code that created them was a very basic call to a SAS procedure without any additional options or tweaking. In extreme cases it could be the graphic was created using the older SAS/GRAPH procedures (possibly still using the old plotter fonts!).

So, I produced a few simple examples of high-resolution SAS ODS Graphic output and showed it to small group of receptive programmers and statisticians, who went on to provide very positive feedback. Now I had convinced them that it was possible, I needed to provide some ***really*** good examples.

THE CHALLENGE

After demonstrating to some senior statisticians that SAS output can be high quality, high resolution and "publication ready", I was shown a graphic in a paper entitled "AKT Inhibition in Solid Tumors With AKT1 Mutations" by Hyman et al, from the Journal of Clinical Oncology [1].

This showed, for each patient, a number of baseline factors such as their tumour types and genetic markers, along with efficacy data such as the best objective response and progression free survival time - all using a combination of waterfall plots, bar charts, and many indicator symbols of various colours and types (Figure 1).
Each column within the graphic represents a patient, and so by looking up and down each one you can pull together a lot of information about each individual.
The colleague then went on to mention that it was likely produced using R, and that SAS probably couldn't do anything like this. I took this as a challenge and attempted to recreate it, at least conceptually, using `PROC TEMPLATE` (Figure 2).

THE SOLUTION

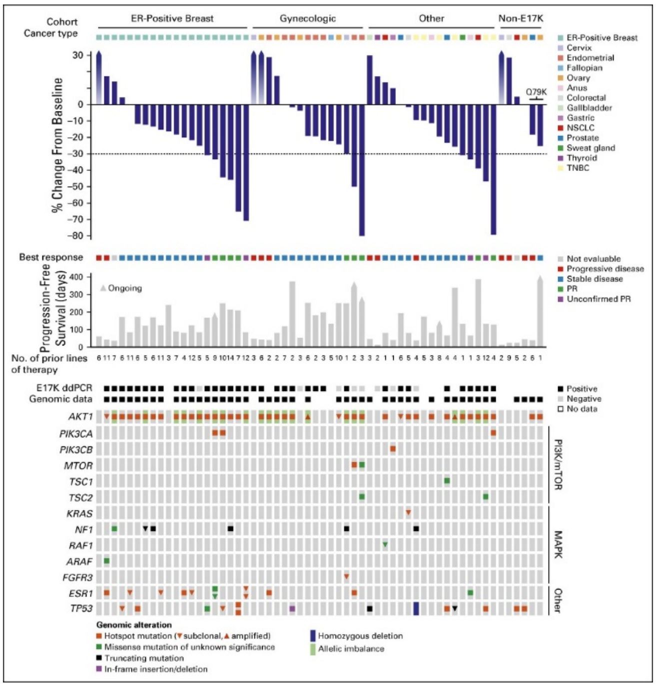


Figure 1 – original graphic (Image from "AKT Inhibition in Solid Tumors With AKT1 Mutations" [1]

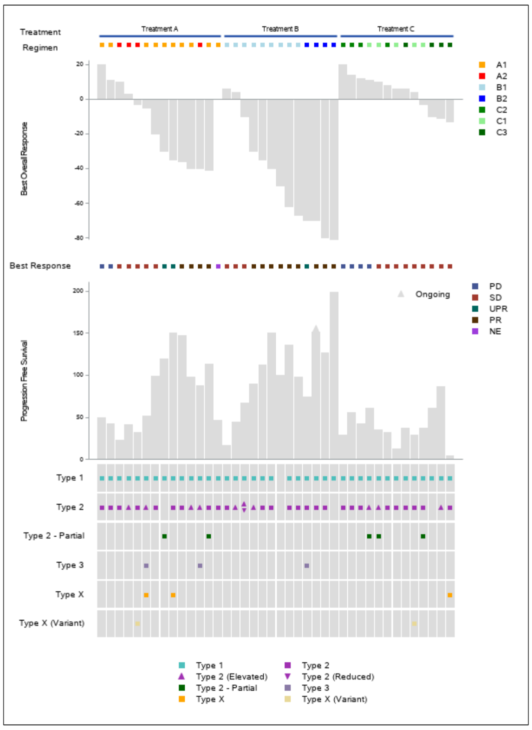


Figure 2 - mock up in SAS

To create the plot shown in Figure 2, the original graphic needed to be broken down into separate elements that can be produced by SAS. Before jumping into some code, let's look at how these elements could be produced. Here I will be focussing on ODS Statistical Graphics only.

PROC SGPLOT

SAS ODS Statistical Graphics can be created in several different ways. The most common would be to use a single procedure to produce a single plot, such as a `PROC SGPLOT` containing plot statements like `SCATTER`, `HBAR`, `VBAR`, etc. Obviously, it is not possible to use this for the complex composite graphic we are attempting.

PROC TEMPLATE

As an alternative, `PROC TEMPLATE` offers a more powerful (and more complicated!) option allowing users to predefine layout templates, that `PROC SGRENDER` uses to draw the final image.

A template is a collection of instructions that describe how a graphic will be generated. It used the concept of layouts to divide the graphical area up into sections, either automatically based on some grouping variables (using `DATALATTICE` and `DATAPANEL`) or using a completely manual approach (with `LATTICE` and `GRIDDED`).

The `GRIDDED` layout allows for every cell to have a completely independent plot, but the `LATTICE` layout offers the ability to share the same axis across rows or columns. The graphic we are seeking to create has a shared x-axis throughout, so that each patient lines up vertically, and so `LAYOUT LATTICE` was chosen.

Each element within a cell could then be created using various SAS Graphic Template Language (GTL) statements once the input data set had been constructed appropriately.

LAYOUT LATTICE

Figure 3 shows the lattice construction, as well as indicating the main GTL statements used for each element.

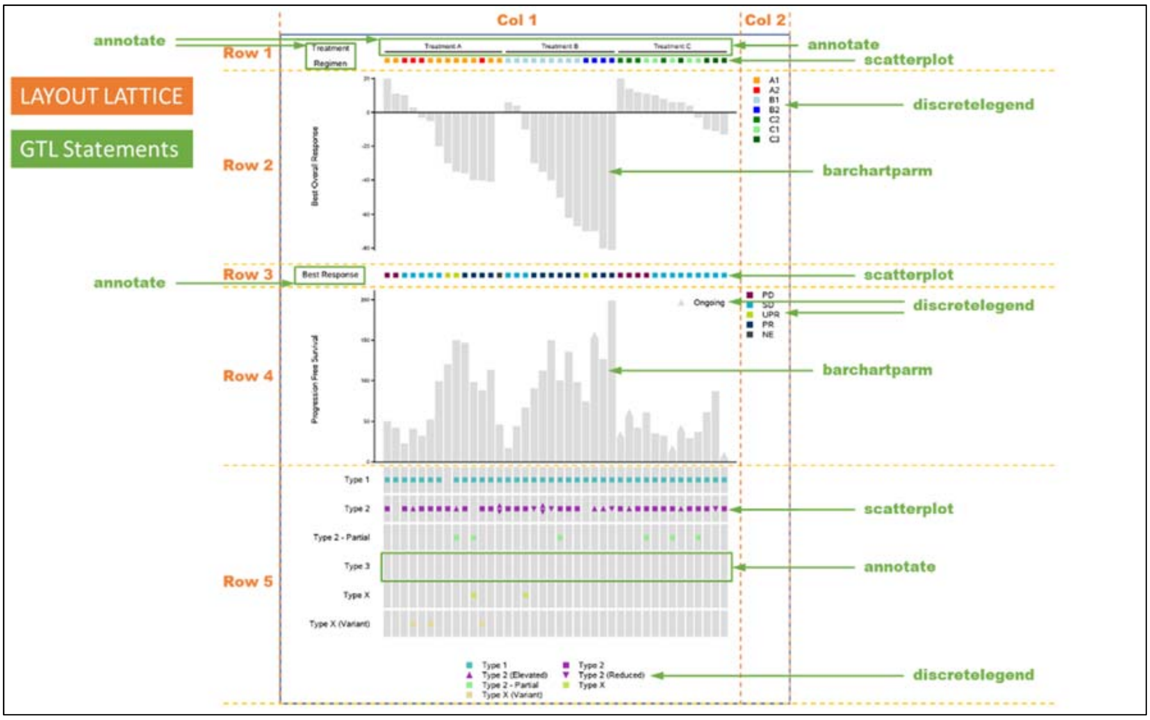


Figure 3 – Lattice construction

The lattice used was 5 rows by 2 columns.
The first column would cover most of the page and contain the main graphical elements. The second column would hold some of the legends alongside the graph they relate to.

The `PROC TEMPLATE` code used was as follows:

```
LAYOUT LATTICE / ROWS=5 COLUMNS=2 ROWWEIGHTS=(2 9 2 9 12)
   columnweights=(0.9 0.1) ROWDATARANGE=union COLUMNDATARANGE=union
   ROWGUTTER=0 COLUMNGUTTER=0;
```

One of the key elements of this lattice, and others shown in this paper, is the `COLUMNDATARANGE=union` option (and companion option `ROWDATARANGE`). This tells SAS that the axis of the various plots within each column are shared and should be lined up with each other. It is this that allows this whole graphic to hang together with the “one column per patient” concept and saves having to specify the same axis information multiple times.

COLOURED SQUARES

The coloured squares were created using a `SCATTERPLOT` within a cell as follows:

```
*--- ROW 1 ---;
*Column 1;
LAYOUT OVERLAY /
   WALLDISPLAY=none XAXISOPTS=(DISPLAY=none) YAXISOPTS=(DISPLAY=none
OFFSETMIN=0);
   annotate / id="toplabel";
   scatterplot x=subjid y=dummy /
      group=regimenname markerattrs=(symbol=squarefilled size=10px )
      name="regi" clusteraxis=x;
ENDLAYOUT;
```

Each patient is assigned a sequential integer in the underlying data set, and this is used to position everything on the x-axis across the whole template. The symbol used with the scatterplot is a simple square marker, coloured by `regimenname` and sized so that markers next to each other don’t quite touch.
Note that the scatter plot is also assigned an internal name, “regi”. This name can be used later to generate a legend.

The second cell was initially a problem as `PROC TEMPLATE` does not like empty cell definitions. However, you can trick SAS into creating one by simply inserting a blank piece of text using the `entry` statement within the cell layout block.

```
*Column 2;
LAYOUT OVERLAY / WALLDISPLAY=(fill);
   entry "";
ENDLAYOUT;
```

ANNOTATION

Another point worth mentioning is the bottom-left cell. This looks complicated but was in fact just a simple set of scatterplots that place the coloured symbols onto a gridded framework of 6 rows.
The "Type 2" row has additional processing in the source data to allow for two symbols to appear at once (one above the other, by shifting the Y coordinate +/- 0.1 as appropriate).

The grey rectangle blocks shown behind the symbols are created using the GTL Annotation Facility. This is very similar to the old SAS/GRAPH® Annotate Facility, and allows you to pass a separate data set to `PROC SGRENDER` containing instructions to draw graphical and textural annotations onto the image. The great feature with GTL is that these annotations can be drawn both over the whole image, or only for particular cells (by using the named plots as shown in the previous example).
In the code below the grey boxes are created with the annotations that are labelled "bgrid" in the annotate dataset, and are only annotated onto this particular `LAYOUT OVERLAY` and none of the others.

```
*--- Row 5 ---;
*Column 1;
LAYOUT OVERLAY / WALLDISPLAY=none XAXISOPTS=(display=none)
                 YAXISOPTS=(DISPLAY=(tickvalues) OFFSETMIN=0
                 tickvalueattrs=(size=6) gridattrs=(color=blue pattern=1)
                 reverse=true linearopts=(viewmin=0.5 viewmax=6.5 minorgrid=off
                 tickvaluelist=(1 2 3 4 5 6)));
   scatterplot x=subjid y=param1 / group=param1 name="param1"
      includemissinggroup=truemarkerattrs=(symbol=squarefilled size=7px
      color=bibg);
   annotate / id="bgrid";
   <more scatterplot statements>
   discretelegend "param1" <more references> / halign=center valign=bottom
across=2
      AUTOITEMSIZE=TRUE valueattrs=(size=6) exclude=(".") border=false
      DISPLAYCLIPPED=true;
ENDLAYOUT;
```

There were a few other annotations on the plot, the "Treatment A/B/C" text and lines at the top for example.

FINISHING UP

The rest of the plot was constructed using `scatterplot` and `barchartparm` statements, and tweaked until everything looked good.
Finally, it was output using a simple `ODS RTF` call, with a DPI of 300 and utilising a customised ODS Style.

NEXT STEPS - BUILDING BLOCKS

After this, I set about creating various examples utilising `PROC TEMPLATE` to create custom graphical output, working closely with a few study teams to create an output that truly helped their analysis. I wanted to both educate study teams on what can be done with SAS Graphics in v9.4, and empower programmers to be able to do it.
Two example outputs are shown at the end of the paper, but before we get there I wanted to show some examples of the building blocks that can be used to construct them.

A QUICK WORD ABOUT TEMPLATES

The purpose of this paper is to provide you with some examples of what can be created with `PROC TEMPLATE` and some its many options and graphical routines. However, the way that the graphic templates are rendered through the use of ODS templates and `PROC SGRENDER` can drastically affect the final image. The ODS Template contains instructions on the fonts, colours, line types and thicknesses to use for both text-based outputs and graphical procedures. However, a detailed discussion about these is out of scope for this paper. All of the single graph type examples that follow use the SAS default ODS template for the screenshots. The larger composite graphics shown do use a customised ODS template, but this is actually a very simplified one and I usually specify font sizes and colours manually within my graphical templates to ensure I have complete control over them.

BASIC PLOTS

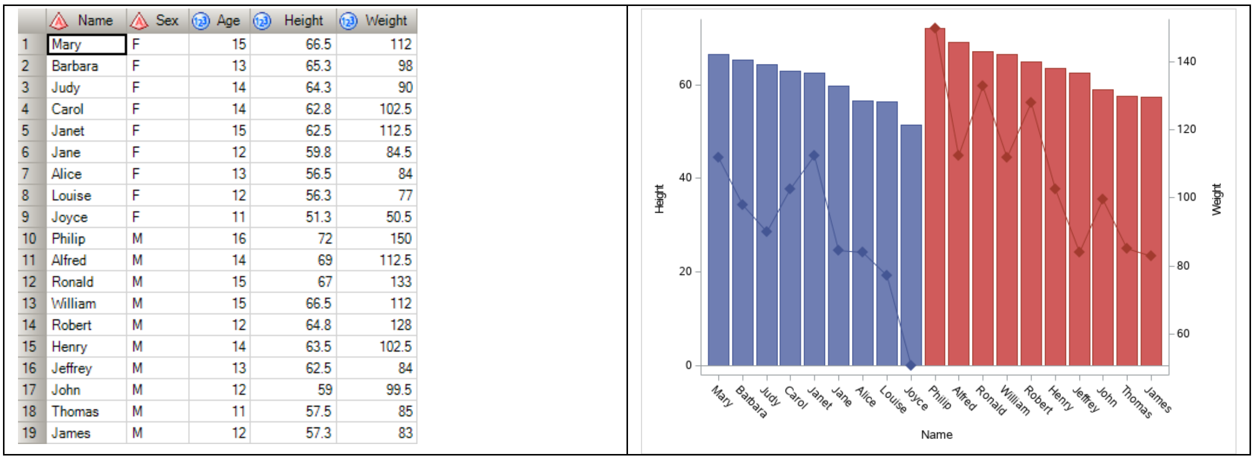
There are a number of basic plot types that SAS creates with hardly any code at all. Things like bar charts, scatter plots and line plots.

For example, using SASHELP.CLASS:

```
proc template;
   define statgraph swim;
      begingraph;
         LAYOUT OVERLAY / WALLDISPLAY=none;
            barchart x=name y=height / group=sex;
            seriesplot x=name y=weight / group=sex yaxis=y2;
            scatterplot x=name y=weight / group=sex yaxis=y2
               markerattrs=(size=10 symbol=diamondfilled);
         endlayout;
     endgraph;
   end;
run;

proc sort data=sashelp.class out=class;
   by sex descending height;
run;

proc sgrender data=class template=swim;
run;
```



	Name	Sex	Age	Height	Weight
1	Mary	F	15	66.5	112
2	Barbara	F	13	65.3	98
3	Judy	F	14	64.3	90
4	Carol	F	14	62.8	102.5
5	Janet	F	15	62.5	112.5
6	Jane	F	12	59.8	84.5
7	Alice	F	13	56.5	84
8	Louise	F	12	56.3	77
9	Joyce	F	11	51.3	50.5
10	Philip	M	16	72	150
11	Alfred	M	14	69	112.5
12	Ronald	M	15	67	133
13	William	M	15	66.5	112
14	Robert	M	12	64.8	128
15	Henry	M	14	63.5	102.5
16	Jeffrey	M	13	62.5	84
17	John	M	12	59	99.5
18	Thomas	M	11	57.5	85
19	James	M	12	57.3	83

Figure 4 - Basic graphic examples

Of course, the resulting plot is not particularly good from a design perspective, but it shows how easy SAS GTL can create different types of graphs.

SWIMMER PLOT

A swimmer plot typically shows an overview of subject's timeline through a study. There is one horizontal bar or "swim lane" per subject, typically showing a key start/end milestone (first/last dose, randomisation/withdrawal, etc). Additionally, other key dates can be shown on this timeline with different symbols (perhaps occurrences of an adverse event of interest, or some sort of response event). Finally, the colours of the swim lanes can be changed too (perhaps to show a key baseline categorisation). Note that graphical elements are built in layers sequentially, with later statements appearing over the top of previous ones, and so when building a `layout overlay` it is good practice to put more important data points last so they aren't obscured by lower priority ones.

```
layout overlay / walldisplay=none
                 xaxisopts=(griddisplay=on)
                 yaxisopts=(reverse=true display=(tickvalues) griddisplay=on);
   highlowplot low=start high=endy  y=subjid / type=bar group=invbor
display=(fill)
      barwidth=0.7;
   scatterplot x=scandy y=subjid / markerattrs=(color=orange
symbol=diamondfilled
      size=14);
   scatterplotx=ongdy y=subjid / markerattrs=(color=big
symbol=trianglerightfilled
      size=20);
   scatterplot x=enddthdy y=subjid / markerattrs=(color=red symbol=squarefilled
      size=14);
endlayout;
```

	subjid	invbor	endy	scandy	ong	trtwdfl	enddthdy	start	ongdy	wddy
1	1001101	SD	72	.	Y		.	0	74	.
2	1001102	PR	64	12		Y	.	0	.	64
3	1001103	SD	51	.	Y		.	0	53	.
4	1001104	CR	50	14	Y		.	0	52	.
5	1002101	PR	48	6	Y		.	0	50	.
6	1002102	SD	42	.		Y	.	0	.	42
7	1002201	SD	35	.	Y		.	0	37	.
8	2001001	SD	30	.	Y		.	0	32	.
9	2001003	SD	28	.		Y	.	0	.	28
10	2001004	SD	23	.		Y	.	0	.	23
11	2002003	PD	9	.		Y	12	0	.	9

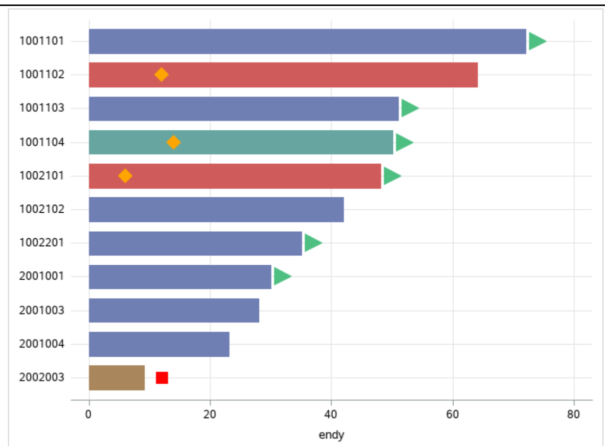
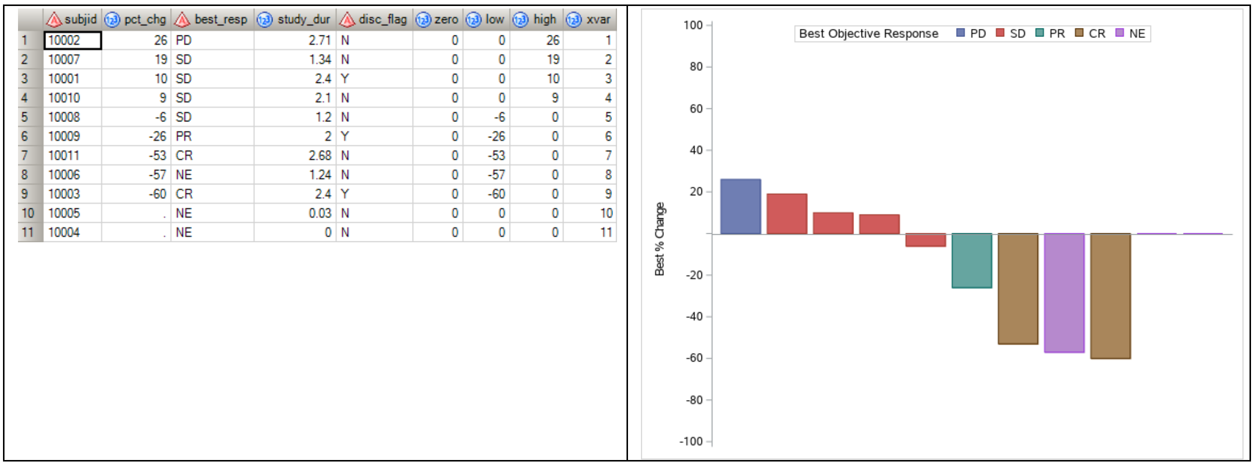


Figure 5 - Swimmer plot example

WATERFALL PLOT

The Waterfall plot is a variant of the bar chart, the main difference being that the bars can cross the axis and show negative values. They are typically used in oncology studies to show overall tumour response, or some other change from baseline parameter. The example below uses `HIGHLOWPLOT`, after setting up the data with low and high variables that start/end at y=0 as appropriate:

```
layout overlay / walldisplay=none
                 xaxisopts=(display=(line) )
                 yaxisopts=(label="Best % Change" reverse=false
                 linearopts=(tickvaluelist=(-100 -80 -60 -40 -20 0 20 40 60 80
100)
                 origin=0 tickvaluepriority=TRUE));
    highlowplot x=xvar low=low high=high / name="bor" type=bar group=best_resp;
    discretelegend "bor" / title="Best Objective Response" location=inside
                           halign=center valign=top across=5;
endlayout;
```



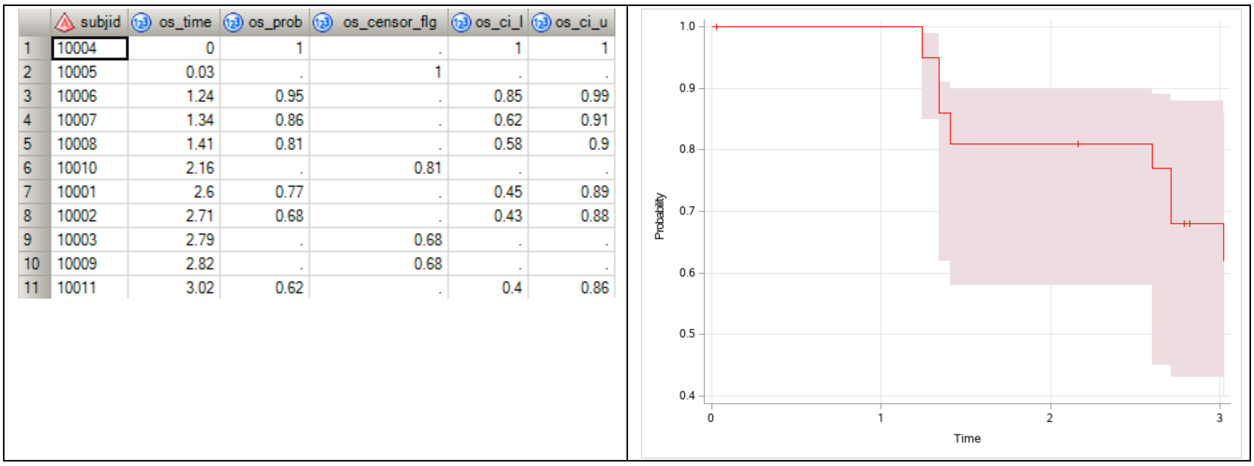
	subjid	pct_chg	best_resp	study_dur	disc_flag	zero	low	high	xvar
1	10002	26	PD	2.71	N	0	0	26	1
2	10007	19	SD	1.34	N	0	0	19	2
3	10001	10	SD	2.4	Y	0	0	10	3
4	10010	9	SD	2.1	N	0	0	9	4
5	10008	-6	SD	1.2	N	0	-6	0	5
6	10009	-26	PR	2	Y	0	-26	0	6
7	10011	-53	CR	2.68	N	0	-53	0	7
8	10006	-57	NE	1.24	N	0	-57	0	8
9	10003	-60	CR	2.4	Y	0	-60	0	9
10	10005	.	NE	0.03	N	0	0	0	10
11	10004	.	NE	0	N	0	0	0	11

Figure 6 - Waterfall plot example

KAPLAN MEIER

The Kaplan Meier plot is a type of step plot, showing how a discrete variable changes over time. It is common in survival analyses for oncology studies, typically showing the overall survival or progression-free survival rates. The data typically comes out of a `PROC LIFETEST`, and can then be plotted using a `STEPPLOT`. 95% Confidence limits can also be added using `BANDPLOT`.

```
layout overlay / walldisplay=none
                 yaxisopts=(label="Probability" griddisplay=on)
                 xaxisopts=(label="Time" griddisplay=on);
   bandplot x=os_time limitlower=os_ci_l limitupper=os_ci_u / type=step
      fillattrs=(color=pkwh);
   stepplot x=os_time y=os_prob / lineattrs=(color=red);
   scatterplot x=os_time y=os_censor_flg / markerattrs=(symbol=plus color=red);
endlayout;
```



	subjid	os_time	os_prob	os_censor_flg	os_ci_l	os_ci_u
1	10004	0	1	.	1	1
2	10005	0.03	.	1	.	.
3	10006	1.24	0.95	.	0.85	0.99
4	10007	1.34	0.86	.	0.62	0.91
5	10008	1.41	0.81	.	0.58	0.9
6	10010	2.16	.	0.81	.	.
7	10001	2.6	0.77	.	0.45	0.89
8	10002	2.71	0.68	.	0.43	0.88
9	10003	2.79	.	0.68	.	.
10	10009	2.82	.	0.68	.	.
11	10011	3.02	0.62	.	0.4	0.86

Figure 7 - Kaplan Meier example

PULLING IT ALL TOGETHER

So far in this paper I have shown how a complex plot could be created using `PROC TEMPLATE` and some tricks that allow multiple plots in a `LAYOUT LATTICE` to use the same axis. I have then gone on to show a selection of common plot types using in clinical trial reporting.

By combining all these techniques together, some very detailed and informative graphics can be created. The following two examples demonstrate this, with graphics that I created for some AstraZeneca Oncology projects.

Please note that the underlying data for both plots has been regenerated manually, and any identifiers/labels altered to ensure the graphic does not show any real patient data.

EXAMPLE 1 - WATERFALL PLOT AND BAR CHARTS

From an Oncology study, Figure 8 shows a tumour response, treatment and baseline information on a single graphic. There are 10 different data elements:

- Best Objective Tumour response (colour of waterfall bar)
- Biomarker categorisation ("Biomarker" or "Other", using shading in the waterfall bars)
- Another baseline categorisation ("AB" or XY", as text)
- Best tumour change from baseline (the waterfall plot)
- Treatment flags (Discontinuation and Combination, via "X" or "#" above/below the waterfall bars)
- Subject number (below/above the waterfall bars)
- Time on treatment (upper bar chart in bottom section)
- Number of prior lines of therapy (Value between bar charts in lower section)
- Duration of previous therapy (lower bar chart in bottom section)
- Type of previous therapy (colour of lower bar chart in bottom section)

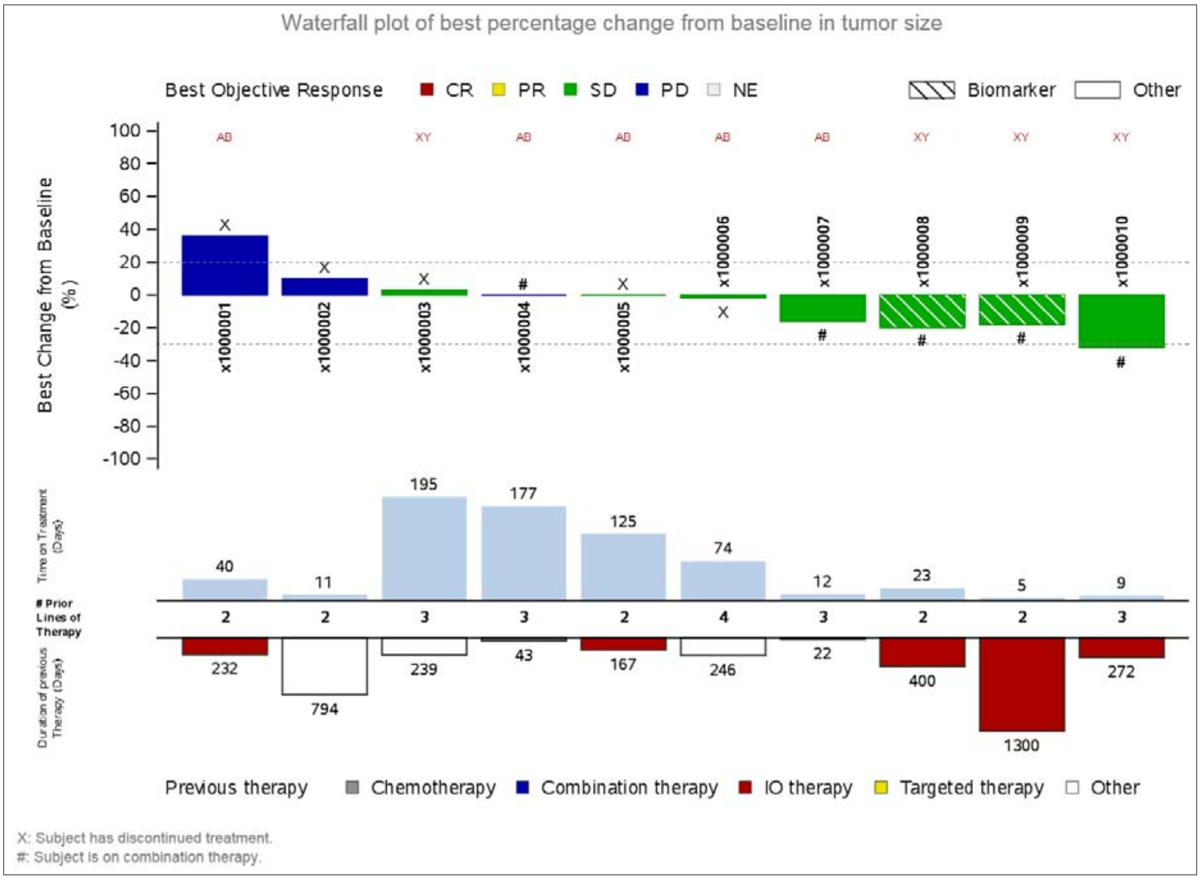


Figure 8 - Advanced composite graphic showing waterfall and bar charts

EXAMPLE 2 – KM, SWIM AND PROGNOSTIC FACTORS

Another Oncology graphic, this time combining data in two directions, is shown in Figure 9.
There is a KM plot, with a Swimmer plot below that share a common x-axis, allowing the reader to follow each KM event/censor marker down onto a subject timeline to look at in context.
Additionally, for each subject in the Swimmer plot, baseline and prognostic factor information is shown on the same y-axis position in an area to the left.
This plot has over a dozen data points, including:

- KM plot for Overall Survival, with confidence interval
- KM plot for Progression Free Survival
- Swim plot, showing:
  - o Time on treatment
  - o Death marker
  - o Progression day marker
  - o Ongoing treatment indicator
- Time on previous therapy mini bar chart
- 5 sets of baseline prognostic factors
  - o colour coded to guide the eye in terms of trend
  - o with Unknown/Low/High indicator

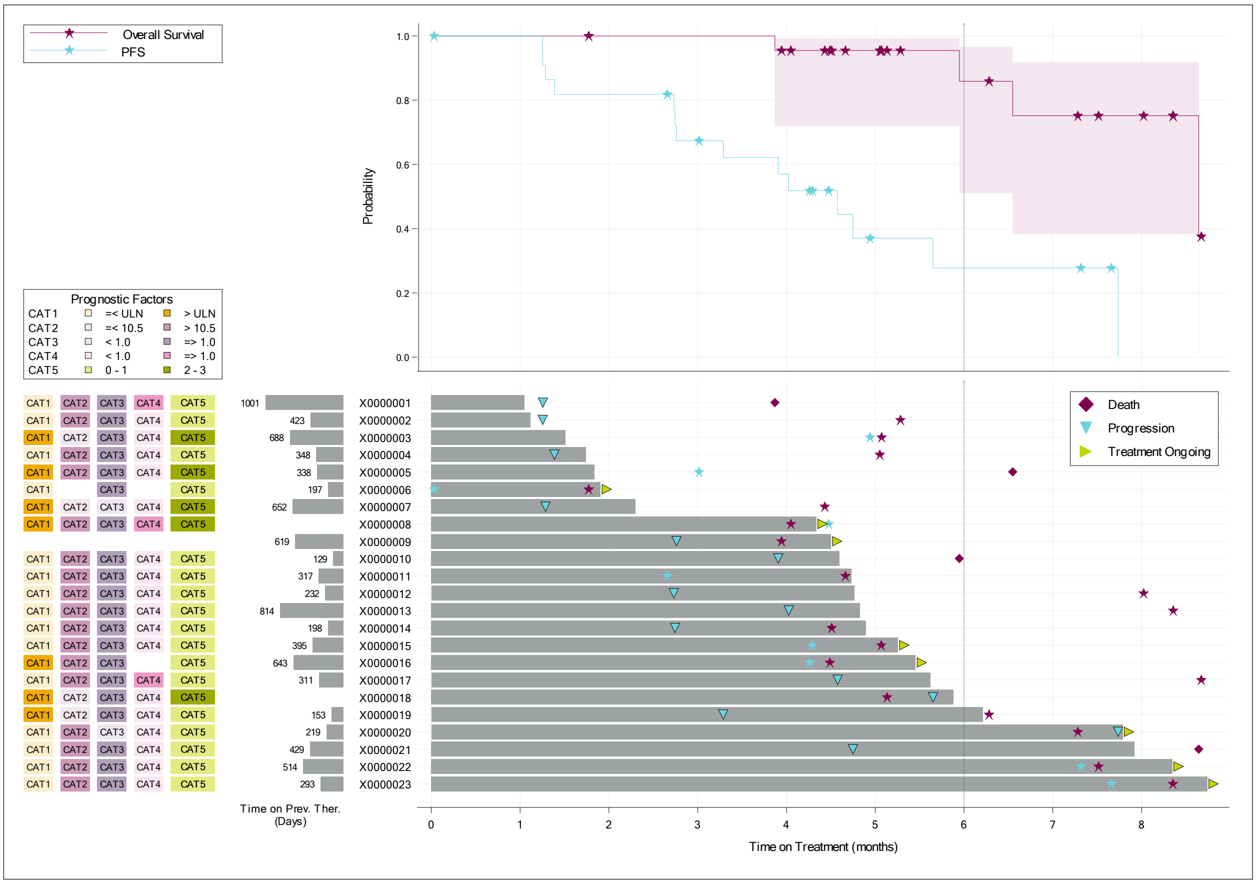


Figure 9 - Advanced composite graphic with KM plot and Swimmer plot connected vertically, and also baseline/prognostic factors and Swimmer plot linked horizontally

CONCLUSION

What began as a simple statement from a statistician (that there is a perception that "SAS cannot produce high quality graphics") turned into over a year and a half of research and development, in addition to my daily project work.

In that time, I not only developed some internal macros to aid in the creation of standardised ODS templates, but also took a lot of the graphics currently in use across Early Clinical Biometrics, stripped them down to their component parts, and then reassembled them to produce clear and concise outputs that went some way to proving SAS graphics can be high quality.

To hammer this point home, I collaborated with the statistician and programmer of a very high-profile study to develop several unique dashboard-style graphics, two of which I have demonstrated in this paper, that made the interpretation of the data more straightforward, given it is all displayed within one graphic.

The graphics were created in a GxP compliant system, and were run regularly to allow quick decision making on the very latest data, which made a big impact on the study teams. It also made it easier for the medics to highlight specific patients to external investigators, and provide a deeper dive into the data of that particular patient. We received very positive feedback, both across the company and externally at investigator meetings.

Rather than rehashing the same tired-looking graphics time and again on projects, often producing extraneous figures that nobody looks at, I began with a clean sheet. By carefully looking at the questions the team wanted to answer, selecting the appropriate data points, and investing time designing a graphic that will communicate this quickly and effectively, I produced some unique and visually appealing graphics that were high quality, high resolution, highly customisable and publication-ready.

This proved that far from the quirky plotter outputs of old – if you can dream it, SAS ODS Graphics can produce it.

REFERENCES

[1] Hyman Et Al, "AKT Inhibition in Solid Tumors With AKT1 Mutations," *Journal of Clinical Oncology,* vol. 35, no. 20, p. 2254, 10 July 2017.

ACKNOWLEDGMENTS

I would like to thank Pat Mitchell for setting me the initial challenge, Jayne Marshall for then setting the bar so high and Rafi Rahi for study programming assistance. Finally, I would like to thank Steven Fox for his patience and perseverance in giving me the chance to put my money where my mouth is and unleash my ideas onto an unsuspecting clinical team.

RECOMMENDED READING

SAS 9.4 graph Template Language Reference, Fifth Edition
http://documentation.sas.com/?docsetId=grstatgraph&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=en

The Visual Display of Quantitative Information, Second Edition by Edward Tufte
https://www.edwardtufte.com/tufte/books_vdqi

Blog: Graphically Speaking – Data Visualisation with a focus on SAS ODS Graphics
https://blogs.sas.com/content/graphicallyspeaking/

Blog: Sanjay Mantange (Director, SAS R&D)
https://blogs.sas.com/content/author/sanjaymatange/

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
neville.cope@astrazeneca.com