PhUSE 2006
Paper TU05

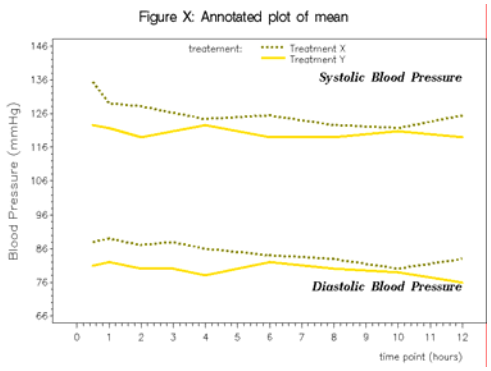
How to annotate graphics

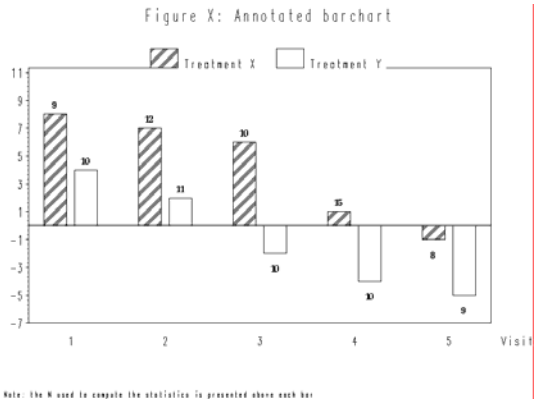
Sandrine STEPIEN, Quintiles, Strasbourg, France

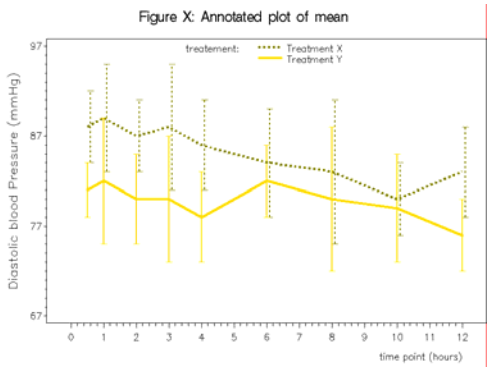
ABSTRACT
Graphs can be annotated using different functions that add graphics elements to the output. Amongst other things, the Annotate facility enables you to label points (plots), bars (horizontal and vertical bar charts), or to generate custom graphics. All information regarding the Annotate is contained in a data set called the Annotate data set that follows a specific structure. To build this data set of graphics commands, values can be assigned either explicitly with data step or implicitly with Annotate macros. A SAS/GRAPH® procedure can be then submitted to produce the graphics output (Annotate option).

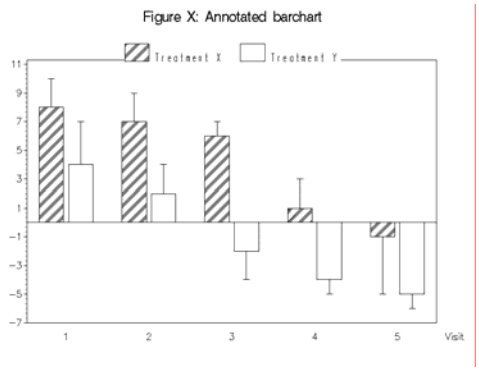
In clinical studies, the Annotate facility can be used to:

- Label and distinguish two types of curves on the same graph (e.g. diastolic and systolic blood pressure),
- Give the sample size at each point on a graph presenting mean over time,
- Create shifted vertical lines to show the variability of data (standard error, standard error of the mean, confidence interval) on a plot of means.









INTRODUCTION

Programming graphs using SAS/GRAPH® is not really easy and user-friendly but with some experience and thanks to the annotate facilities it is possible to program some types of graphs which don't seem possible. The Annotate facility gives a large number of possibilities to customize your graph. Graphics elements as lines, text, and polygons can be added inside your graph in order to enhance graphics information or to make it clearer. This paper will give you some technical clues and some concrete examples in order to give you an overview of how to annotate graphics.

ANNOTATE DATA SET

The Annotate data set contains all information concerning the graphics elements to be added to the graphics output. In an Annotate data set, each observation represents a command to draw a graphics element or to perform an action. The Annotate variables have predefined names. The Annotate facility looks only for variables with those names. Other variables can be present, but they are ignored.

There are three types of variables:

- An action variable tells **what** to do. The only action variable is FUNCTION, which specifies what graphics element to draw or what action to take.
- Positioning variables tells **where** to do it. The positioning variables specify the point at which to draw the graphics element.
- Attribute variables tells **how** to do it. The attribute variables specify the characteristics of the graphics element (for example, color, size, line style, text font, position).

Summary of Annotate Variables

Task Group	Variable	Description
Variable that defines an action	FUNCTION	specifies a drawing or programming action; Summary of Graphics Tasks Performed by Annotate Functions describes these actions.
Positioning variables that determine coordinate values	GROUP	uses the value of the GCHART GROUP= option in place of X or Y
	MIDPOINT	uses the value of the GCHART MIDPOINT= option in place of X or Y
	SUBGROUP	uses the value of the GCHART SUBGROUP= option in place of X or Y
	X	specifies a numeric horizontal coordinate
	Y	specifies a numeric vertical coordinate
	Z	specifies a numeric third dimensional coordinate; used with G3D procedure only
	XC	specifies a horizontal character coordinate; only used with data coordinate systems 1, 2, 7, 8
	YC	specifies a vertical character coordinate; only used with data coordinate systems 1, 2, 7, 8
Positioning variables that contain internal coordinates	XLAST, YLAST	contain the X and Y coordinates of the last nontext function
	XLSTT, YLSTT	contain the X and Y coordinates of the last text function
Positioning variables that	HSYS	specifies type of units for the SIZE variable

specify coordinate systems		
	XSYS	specifies coordinate system for X or XC coordinates
	YSYS	specifies coordinate system for Y or YC coordinates
	ZSYS	specifies coordinate system for Z coordinate (G3D procedure only)
Attribute variables	ANGLE	angle of text label or starting angle of a pie slice
	CBORDER	colored border around text or symbol
	CBOX	colored box behind text or symbol
	COLOR	color of a graphics primitive
	LINE	line type to use in drawing or special control over pies and bars
	POSITION	placement and alignment for text strings
	ROTATE	angle at which to place individual characters in a text string or the delta angle (sweep) of a pie slice
	SIZE	size of an aspect of a graphics primitive; depends on FUNCTION variable (for TEXT, height of characters; for PIE, pie slice radius; for DRAW, line thickness; and so on)
	STYLE	font or pattern for a graphics element, depends on the FUNCTION variable
	TEXT	text to use in a label, symbol, or comment
	WHEN	whether a graphics element is drawn before or after procedure graphics output
Web variable	HTML	specifies link information for a drill-down graph

FUNCTION VARIABLE

The different actions that can be performed on a graphic are the following:

Annotate functions	Action
LABEL	Draw text
MOVE	Move to the specified point (X,Y)
DRAW	Draw a line from the current (X,Y) position
POINT	Draw a point
BAR	Draw a rectangle from the current (X,Y) position; optionally fill with a pattern

There are also other functions (‘FRAME, ‘PIE, ‘POLY’, ‘POLYCONT’, ‘SYMBOL’). For more details please check *http://v8doc.sas.com/sashtml/gref/zaboutds.htm#znctable* .

COORDINATE SYSTEM VARIABLES

The first action to perform before creating the annotate actions is to define the data coordinate system for the Annotate. It determines how coordinates will be interpreted depending on:

- the **area** where annotate elements will be inserted. It could be '*data*', '*graphics output area*', '*procedure output area*',
- the **unit** for the coordinate system (values, %, cells)
- the **placement** (absolute or relative).

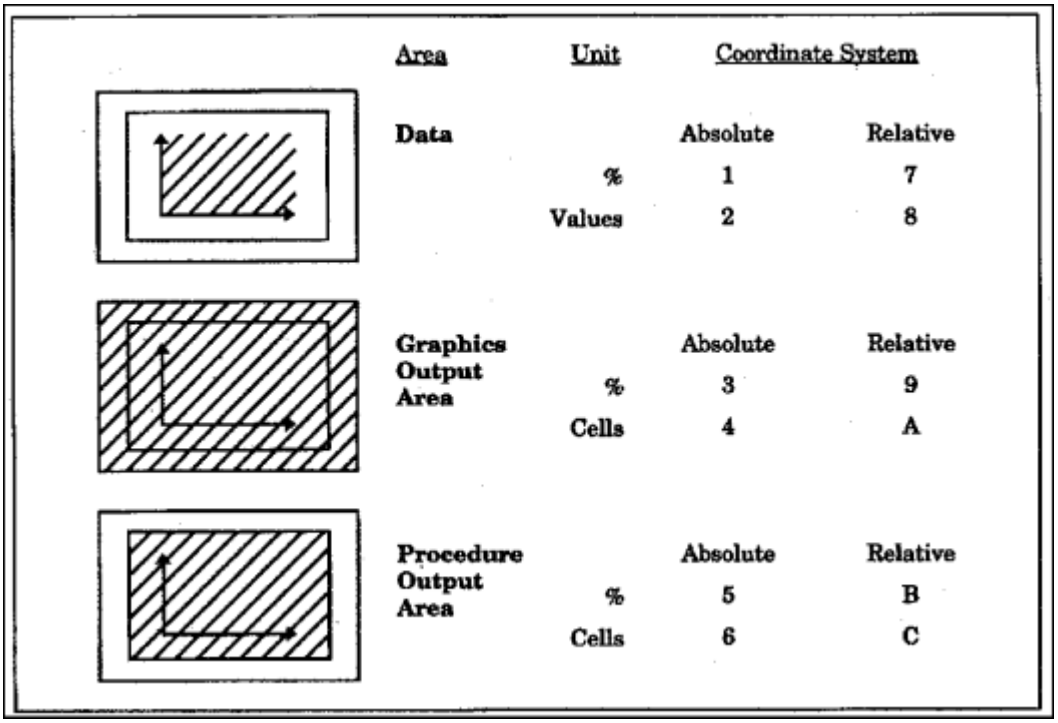
The data coordinate system is coded by the following variables XSYS, YSYS, ZSYS (for 3-D graphs), HSYS (specifies the type of units for the SIZE variable).

To add information inside the graph itself (area delimited by the graph axis), and to use plotted value coordinates to draw annotate elements, the method to be used is:

- Area defined as 'Data'
- Unit defined as 'Values'
- Placement defined as 'Absolute'

Which corresponds to **XSYS='2'** and **YSYS='2'**.

Areas and Their Coordinate Systems



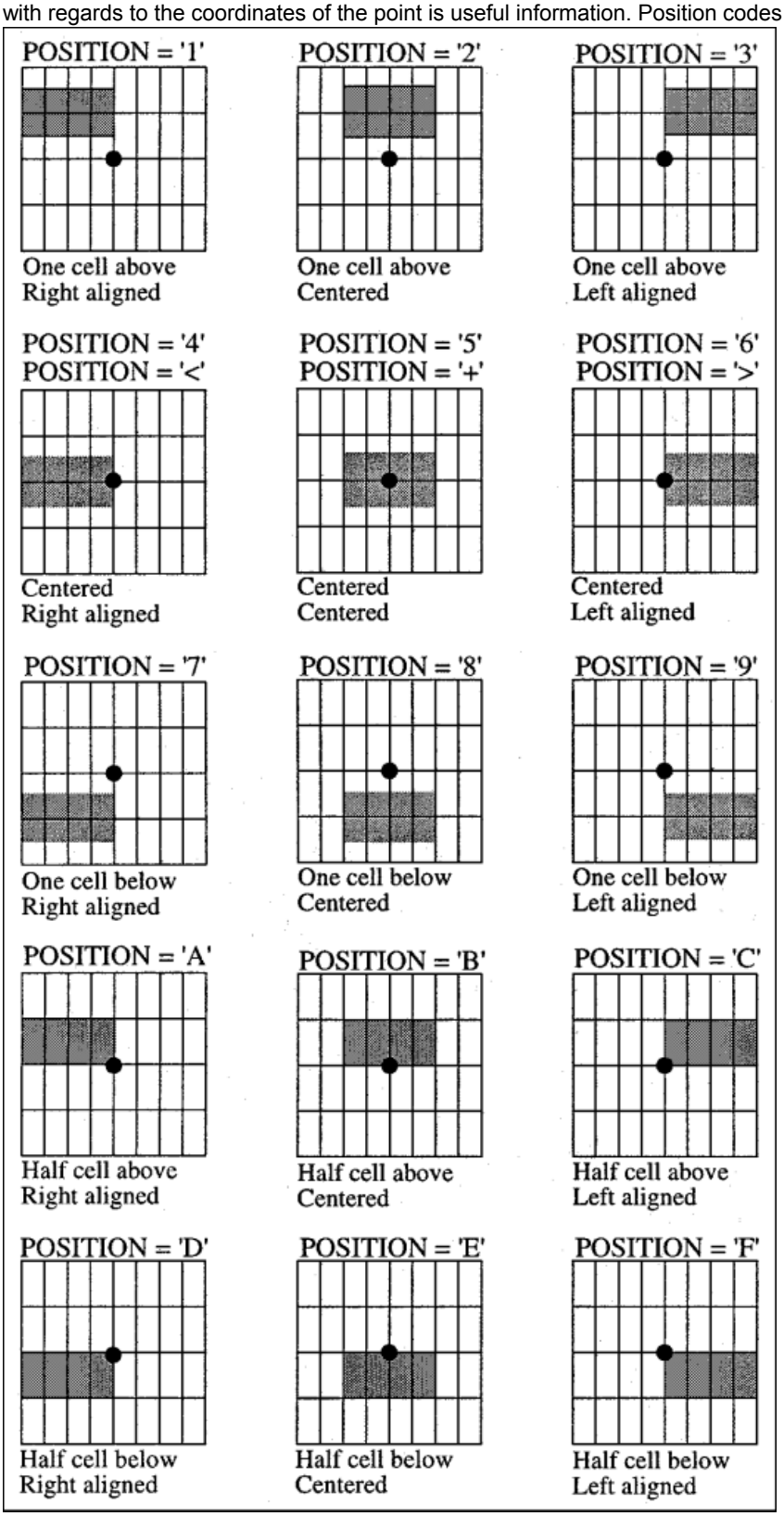
POSITIONING VARIABLES

The positioning variables used to locate graphics elements have to be chosen according to the type of graph you want to annotate. See the table below:

Positioning variables	Action
X	Specifies a numeric horizontal coordinate
Y	Specifies a numeric vertical coordinate
Z	Specifies a numeric third dimensional coordinate, used with the G3D procedure only
XC	Specifies a horizontal character coordinate; only used with data coordinate systems 1, 2, 7, 8
YC	Specifies a vertical character coordinate; only used with data coordinate systems 1, 2, 7, 8
GROUP	To annotate output from the GCHART procedure. Use these variables to specify coordinates for horizontal or vertical bar charts
MIDPOINT	
SUBGROUP	

ATTRIBUTE VARIABLES

They determine how graphics elements will be output. For example, the code which defines the position of the label with regards to the coordinates of the point is useful information. Position codes are given in the figure below.



Color, font, style have also to be defined if required by the function variable. Information can be found in the SAS® online doc.

ANNOTATE MACROS

A set of Annotate macros is provided in the SAS® sample library. This is the implicit way of creating Annotate data sets and it can be used within a SAS® DATA step to simplify the process of creating Annotate observations. With a macro, functions and variable values are specified and assigned in one step without having to write explicit variable assignment statements. Macro calls and assignment statements can be mixed in the same DATA step.

The ANNOMAC macro must be run before any other Annotate macros are used in a SAS® session. This enables you to access macro facilities to create your Annotate data set. A message in the SAS® log shows how to get help for using the macros.

Tasks with Annotate Macros	
If you want to...	**Use this macro...**
assign values of (XLSTT,YLSTT) to (XLAST,YLAST)	%TXT2CNTL;
begin drawing a polygon	%POLY(**x, y, color, style, line**);
continue drawing a polygon	%POLYCONT(**x, y, color**);
copy (XLAST,YLAST) to (XLSTT,YLSTT)	%CNTL2TXT;
declare all variables	%DCLANNO;
draw a bar	%BAR(**x**1, **y**1, **x**2, **y**2, **color, line, style**);
draw a circle	%CIRCLE(**x, y, size, color**);
draw a frame	%FRAME(**color, line, size, style**);
draw a line from (XLAST,YLAST) to (XLSTT,YLSTT)	%DRAW2TXT(**color, line, size**);
draw a line from previous point	%DRAW(**x, y, color, line, size**);
draw a line	%LINE(**x**1, **y**1, **x**2, **y**2, **color, line, size**);
draw a pie slice or arc	%SLICE(**x, y, angle, rotate, size, color, style, line**);
draw a rectangle	%RECT(**x** 1,**y** 1,**x** 2,**y** 2, **color, line, size**);
draw text	%LABEL(**x, y, text, color, angle, rotate, size, style, position**);
exchange the values of (XLAST,YLAST) and (XLSTT,YLSTT)	%SWAP;
move to a point near a pie slice	%PIEXY(**angle, size**);
move to a point without drawing	%MOVE(**x, y**);

put values into a stack	%PUSH;
retrieve values from a stack	%POP;
scale and move input	%SCALET(**ptx, pty, x**0, **y**0, **x**1, **y**1, **x**0, **vy**0, **vx**1, **vy**1);
scale input	%SCALE(**ptx, pty, x**0, **y**0, **x**1, **y**1, **x**0, **vy**0, **vx**1, **vy**1);
set the coordinate system for the observation	%SYSTEM(**xsys, ysys, hsys**);
set when to draw an observation	%SEQUENCE(**when**);
write a comment to the data set	%COMMENT(**text**);

EXAMPLE 1

This example shows how to get help for the Annotate macros

THE PROGRAM

```
%HELPANO(ALL);
%annomac;
```

THE LOG MESSAGE

```
For further information on ANNOTATE macros, enter,
    %HELPANO(macroname), (for specific macros)
    %HELPANO(ALL), (for information on all macros)
 or %HELPANO (for a list of macro names)

1791  %HELPANO(ALL);
                ANNOTATE MACROS

 The ANNOTATE macros are designed to lessen the burden
 of coding ANNOTATE observations. They are to be used
 within the confines of a SAS data step. Each of the
 available macros is described briefly below. Consult
 the ANNOTATE documentation for command parameters.

 BAR(x1, y1, x2, y2, color, bar_type, pattern)
   draws fillable rectangle with opposing corners at
   (x1, y1) and (x2, y2)
   valid bar_type values are :
       0   draw all edges          1   vertical adjust
       2   horizontal adjust       3   draw no edges
   color of asterisk(*) denotes previous assignment

 CIRCLE(x1, y1, rad, color)
   draws empty circle with center at (x1, y1)
…
1792  %annomac;

*** ANNOTATE macros are now available ***
```

EXAMPLE 2

Example 2 shows you how to define the Annotate data set. When you use the SAS® DATA step with assignment statements, each statement provides a value for an annotate variable. After assigning all the variable values for an observation, the OUTPUT statement has to be added to write the observation to the data set. It is easy to compare assignment statements with the Annotate macro calls that simplify the program. The MPRINT option gives you the details in the log of the explicit variable statements corresponding to the Annotate macros.

THE PROGRAM

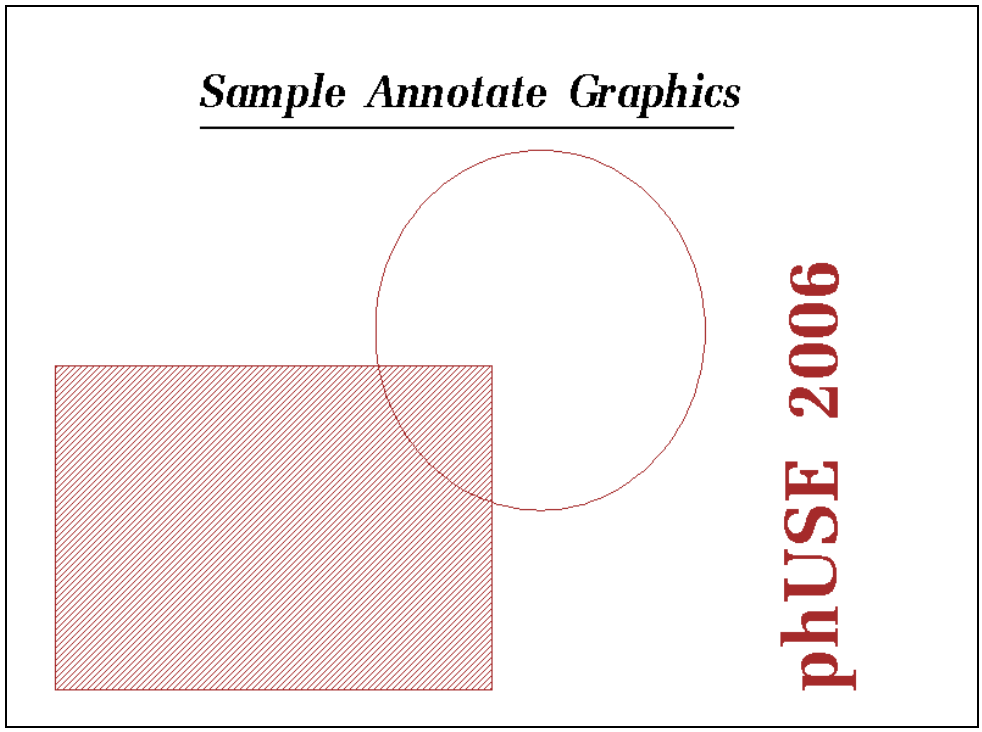
```
data test;
  *declare variables;
  ******************;
  length function style color $8 text $50;

  *coordinate system using absolute % for the whole graph area;
  ************************************************************;
  retain hsys xsys ysys '3';

    *create observation to draw the title;
    ************************************;
    function='LABEL';
    X=50;Y=90;
    position='5';               /*equal to X=23;Y=90;position='6';*/
    text='Sample Annotate Graphics';
    style='zapfbi';
    color='black';
    size=6;
    OUTPUT;

    *create observations to draw the line below the title;
    ****************************************************;
    function='MOVE';
    X=23;Y=83;
    OUTPUT;

    function='DRAW';
    X=78;Y=83;
    size=.5;
    color='black';
    line=1;
    OUTPUT;

    *create graphics element using the Annotate macros;
    *************************************************;
    %annomac;
    %bar(5,5,50,50, brown, 0, r2);
    %circle(55,55,25,brown);
    %label(90,35,"phUSE 2006",brown,90,0,10,zapfb,2);
run;

proc ganno annotate=test;
run;quit;
```

THE ANNOTATE DATA SET

ixqf1&	Vw|oh	froru	wh{w	kv|v2	[v|v2	\v|v	[	\	Srv-	vl}h	olqh	Dqjoh	urwdwh
ODEHO	}dsiel	eodfn	Vdpsoh Dqqrwdwh Judsklfv	6	6	6	83	<3	8	913	1	1	1
PRYH				6	6	6	56	;6		1	1	1	1
GUDZ				6	6	6	:;	;6		318	4	1	1
PRYH		eurzq		6	6	6	8	8				1	1
EDU	u5	eurzq		6	6	6	83	83		313	3	1	1
SLH	HPSW\	eurzq		6	6	6	88	88		5813	3	3	693
ODEHO	}dsie	eurzq	skXVH 5339	6	6	6	<3	68	5	4313		<3	3

function variable
* position variable

THE GRAPHICS OUTPUT



EXAMPLE 3

You can add text anywhere in the graphics output to identify some points or add some information in order to make your graphics clearer. Here is an example of labeling curves of blood pressure when systolic and diastolic BP are on the same graph. Coordinates for the text have been chosen knowing that diastolic blood pressures are lower than systolic blood pressures.

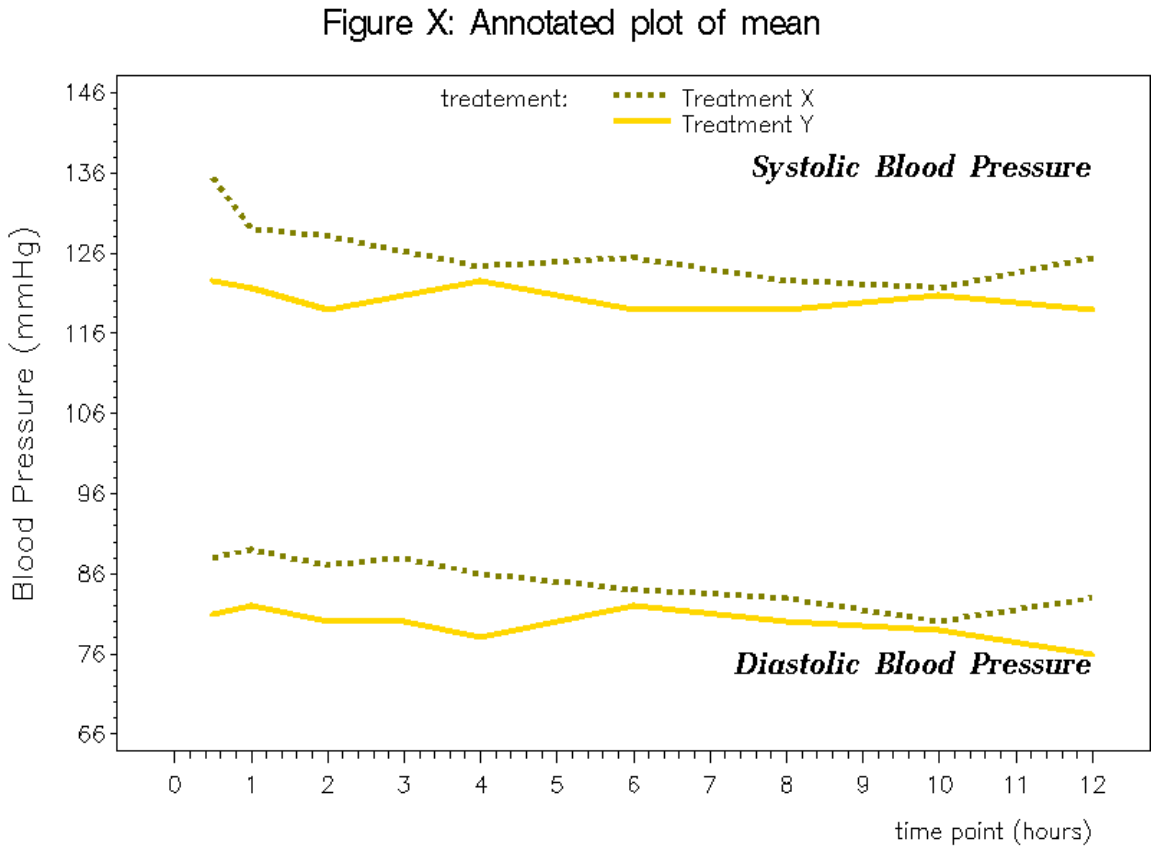
THE PROGRAM

```
*enter descriptive statistics data;
*--------------------------------;
data mean;
  length bp $9;
  input trt time dbp sbp;
  cards;
1 0.5 88 138
1 1 89 131
1 2 87 130
1 3 88 128
1 4 86 126
1 6 84 127
1 8 83 124
```

```
1 10 80 123
1 12 83 127
2 0.5 81 124
2 1 82 123
2 2 80 120
2 3 80 122
2 4 78 124
2 6 82 120
2 8 80 120
2 10 79 122
2 12 76 120
;
run;

*create min and max values to annotate/label curves below or above the line curves;
*-------------------------------------------------------------------------------;
proc sql noprint;
  create table MIN as select min(dbp) as Y,max(time) as X from mean;
  create table MAX as select max(sbp) as Y,max(time) as X from mean;
  select min(dbp-10) into:minval from mean;
  select max(SBP+10) into:maxval from mean;
quit;

*create the annotate data set using DATA STEP;
*--------------------------------------------;
data ANNO1;
  length function style color $8 text $50;
  set MIN(in=min) MAX(in=max);
    *coordinate system;
    ******************;
    XSYS='2';
    YSYS='2';
    *add text on the graph area;
    ***************************;
    if min then do;
           function='LABEL';
           text='Diastolic Blood Pressure';
           position='4';  *(right align);
           style='zapfbi';
           color='black';
           size=1.5;
    end;
    if max then do;
           function='LABEL';
           text='Systolic Blood Pressure';
           position='4';
           style='zapfbi';
           color='black';
           size=1.5;
    end;
run;
proc format;
  value TRT 1='Treatment X'
            2='Treatment Y';
run;

*generate the graph;
*------------------;
goptions reset=all
         display
         cback=white
         ftext=simplex
         ftitle=swiss
         colors=(black)
```

```
          gunit=pct
          ctext=black
          htext=3
          htitle=2;


axis1      offset=(5,5) label=(h=3 j=r "time point (hours)" )
           value=(h=3);
axis2      offset=(2,2) label=(h=4 angle=90 "Blood Pressure (mmHg)")
           value=(h=3) order=(&minval to &maxval by 10);
axis3      offset=(5,5) label=none value=none major=none minor=none
           order=(&minval to &maxval by 10);

legend1 label=(h=3 'treatement:') position=(top center inside)
          mode=protect value=(h=3 j=left) across=1;

symbol1 i=join value=none line=2 color=olive width=4 h=2;
symbol2 i=join value=none line=1 color=gold width=4 h=2;

symbol3 i=join value=none line=2 color=olive width=4 h=2;
symbol4 i=join value=none line=1 color=gold width=4 h=2;

title;footnote;
title1 j=c h=4 "Figure X: Annotated plot of mean";

proc gplot data=mean gout=graf;
   plot DBP*time=trt/   vaxis=axis2 haxis=axis1 legend=legend1 annotate=ANNO1;
   plot2 SBP*time=trt/  vaxis=axis3 haxis=axis1 nolegend;
   format trt trt.;
run;quit;
```

THE GRAPHICS OUTPUT



EXAMPLE 4

Bars/lines representing the variability of data could be added using Annotate. This example will show you how to create a plot of mean +/- SEM with shifted bar to avoid overlapping.

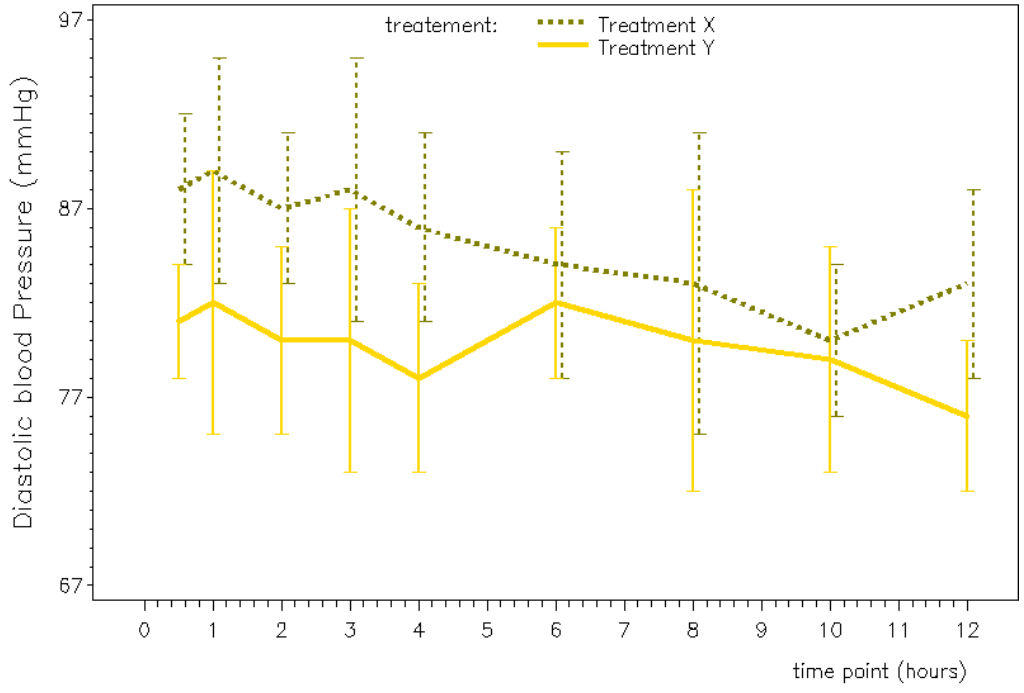
THE PROGRAM

```
*enter descriptive statistics data;
*--------------------------------;
data mean;
  input trt time dbp sem n;
  cards;
1 0.5 88 4 10
1 1 89 6 9
1 2 87 4 12
1 3 88 7 10
1 4 86 5 10
1 6 84 6 11
1 8 83 8 15
1 10 80 4 8
1 12 83 5 8
2 0.5 81 3 13
2 1 82 7 10
2 2 80 5 11
2 3 80 7 12
2 4 78 5 13
2 6 82 4 10
2 8 80 8 10
2 10 79 6 9
2 12 76 4 8
;
run;
proc sql noprint;
  select min(dbp-sem-5) into:minval from mean;
  select max(dbp+sem+5) into:maxval from mean;
quit;

*create the annotate data set using Annotate macros;
*--------------------------------------------------;
%annomac;
data ANNO1;
  length function style color $8 text $50;
  set mean;
    tiret='_';
    %annomac;
    %system(2,2);
    *--color--*;
    if trt=1 then do;
           %line(time+0.1,dbp-sem,time+0.1,dbp+sem,olive,2,2);
           %label(time+0.1,dbp-sem,tiret,olive,0,0,1,simplex,B);
           %label(time+0.1,dbp+sem,tiret,olive,0,0,1,simplex,B);
    end;
    if trt=2 then do;
           %line(time,dbp-sem,time,dbp+sem,gold,1,2);
           %label(time,dbp-sem,tiret,gold,0,0,1,simplex,B);
           %label(time,dbp+sem,tiret,gold,0,0,1,simplex,B);
    end;
run;
proc format;
  value TRT 1='Treatment X'
            2='Treatment Y';
run;
```

```
*generate the graph;
*------------------;
goptions reset=all
         display
         cback=white
         ftext=simplex
         ftitle=swiss
         colors=(black)
         gunit=pct
         ctext=black
         htext=3
         htitle=2;


axis1      offset=(5,5) label=(h=3 j=r "time point (hours)" ) value=(h=3);
axis2      offset=(2,2) label=(h=4 angle=90 "Diastolic blood Pressure (mmHg)")
           value=(h=3) order=(&minval to &maxval by 10);

legend1 label=(h=3 'treatement:') position=(top center inside)
          mode=protect value=(h=3 j=left) across=1;

symbol1 i=join value=none line=2 color=olive width=4 h=2;
symbol2 i=join value=none line=1 color=gold width=4 h=2;

title;footnote;
title1 j=c h=4 "Figure X: Annotated plot of mean";

proc gplot data=mean gout=graf;
 plot DBP*time=trt/     vaxis=axis2 haxis=axis1
                        legend=legend1 annotate=ANNO1;
  format trt trt.;
run;quit;
```

THE GRAPHICS OUTPUT

Figure X: Annotated plot of mean



EXAMPLE 5

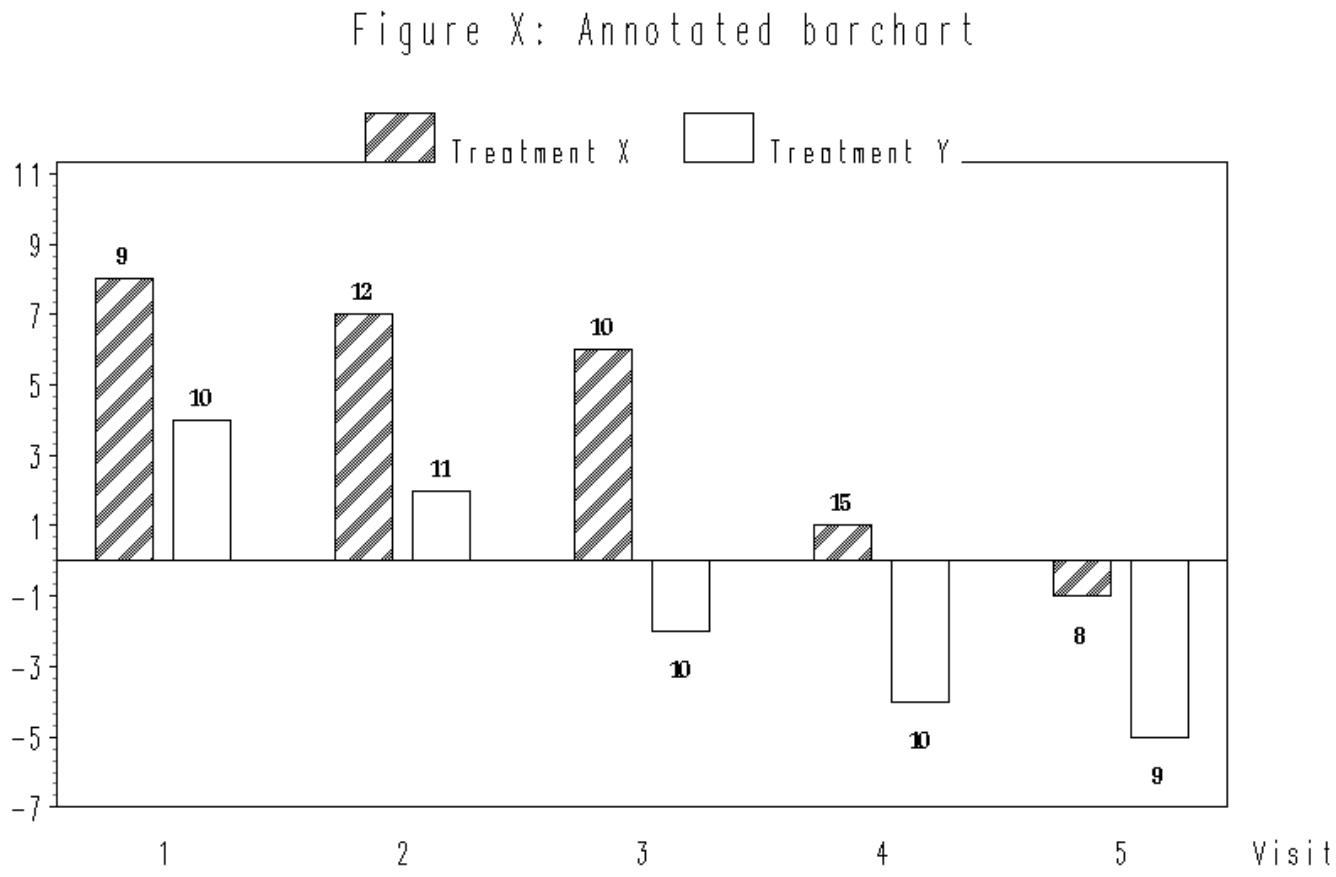
The Annotate facility can also be used with the GCHART procedure. The coordinates are not X or Y but GROUP and MIDPOINT. The position of the text depends on the bar values (positive or negative) and is positioned using the LABEL macro from the Annotate macro facility.

THE PROGRAM

```
data mean;
  input trt visit mean sem n;
  cards;
1 1 8 2 9
1 2 7 2 12
1 3 6 1 10
1 4 1 2 15
1 5 -1 4 8
2 1 4 3 10
2 2 2 2 11
2 3 -2 2 10
2 4 -4 1 10
2 5 -5 1 9
;
run;
proc sql noprint;
  select min(mean-sem-1) into:minval from mean;
  select max(mean+sem+1) into:maxval from mean;
quit;
%annomac;
data ANNO1;
  length function style color $8 text $50;
  set mean;
    %annomac;
    %system(2,2);
    Nchar=put(n,2.);
    *--label the bar with the number of observation per bar--*;
    if mean>=0 then do;
            %label(trt,mean,Nchar,black,0,0,1,zapfb,2);
    end;
    if mean<0 then do;
            %label(trt,mean,Nchar,black,0,0,1,zapfb,8);
    end;
    rename visit=group trt=subgroup;
run;
proc format;
  value TRT 1='Treatment X'
            2='Treatment Y';
run;
goptions reset=all
         display
         cback=white
         ftext=simplex
         ftitle=swiss
         colors=(black)
         gunit=pct
         ctext=black
         htext=3
         htitle=2;

axis1 label=(h=3 j=right 'Visit') value=(h=3);
axis2 label=none value=(h=3) order=(&minval to &maxval by 2) minor=(n=5 color=gray);
axis3 label=none value=none offset=(5,5);

pattern1 value=r5 color=black;
pattern2 value=E color=black;
```

```
legend1 shape=bar(5,2)CELLS label=none position=(top center outside)
          mode=protect value=(j=left h=3 font=HWCGM003);

title;footnote;
title1 j=c h=4 "Figure X: Annotated barchart";

proc gchart data=mean  gout=graf;
  vbar trt /  discrete
              sumvar=mean
              width=4
              space=1
              gspace=1
              group=visit
              subgroup=trt
              maxis=axis3
              raxis=axis2
              gaxis=axis1
              annotate=anno1
              legend=legend1;
      format  trt trt. ;
run;
quit;
```

THE GRAPHICS OUTPUT



Note: the N used to compute the statistics is presented above each bar

EXAMPLE 6

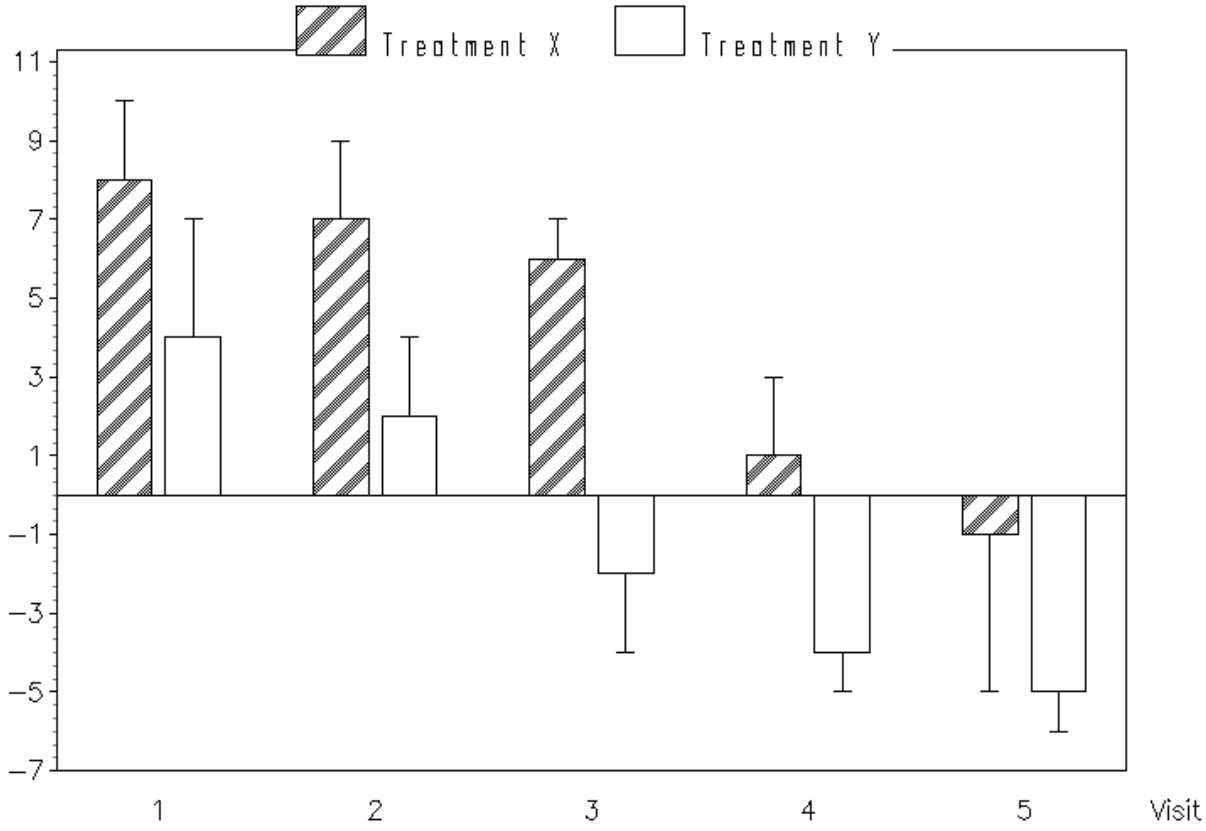
Instead of adding the number of observation by bar, a indicator of the distribution dispersion could be visualized. The program will be the same as example 5 but the ANNO1 data step should be replace by the following program:

THE PROGRAM

```
data ANNO1;
  length function style color $8 text $50;
  set mean;
    tiret='_';
    %annomac;
    %system(2,2);
    *--add SEM bar--*;
    if mean>=0 then do;
            %line(trt,mean+sem,trt,mean,black,1,1);
            %label(trt,mean+sem,'_',black,0,0,1,simplex,B);
    end;
    if mean<0 then do;
            %line(trt,mean-sem,trt,mean,black,1,1);
            %label(trt,mean-sem,'_',black,0,0,1,simplex,B);
    end;
    rename visit=group trt=subgroup;
run;
```

THE GRAPHICS OUTPUT

Figure X: Annotated barchart



EXAMPLE 7

For survival plots, the censored events and the number of patients censored could be added as graphics element:

THE PROGRAM

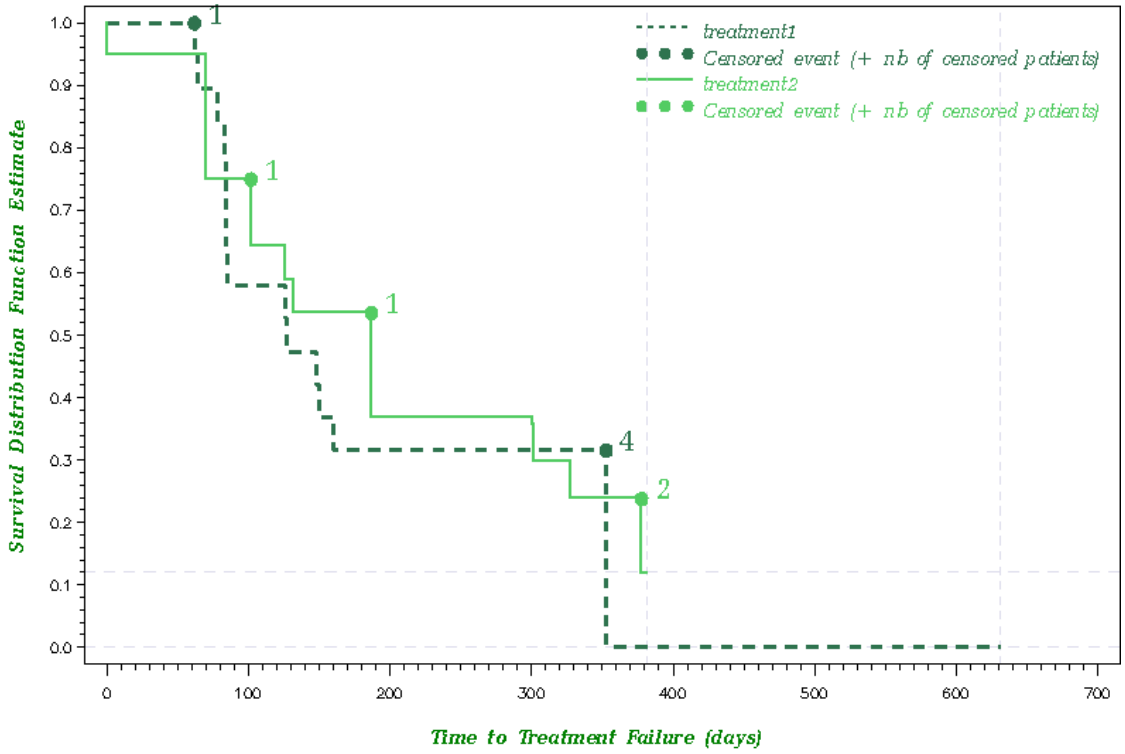
```
data _survival;
  input doselev $1-10 timefailure _censor_ survival;
  cards;
treatment1          0            0        1.00000
treatment1         62            1        1.00000
treatment1         64            0        0.94737
treatment1         78            0        0.89474
treatment1         83            0        0.84211
treatment1         84            0        0.78947
treatment1         85            0        0.63158
treatment1        126            0        0.57895
treatment1        127            0        0.52632
treatment1        148            0        0.47368
treatment1        150            0        0.42105
treatment1        160            0        0.36842
treatment1        353            0        0.31579
treatment1        353            1        0.31579
treatment1        353            1        0.31579
treatment1        353            1        0.31579
treatment1        353            1        0.31579
treatment1        631            0        0.00000
treatment2          0            0        1.00000
treatment2         70            0        0.95000
treatment2        102            0        0.75000
treatment2        102            1        0.75000
treatment2        126            0        0.64286
treatment2        132            0        0.58929
treatment2        187            0        0.53571
treatment2        187            1        0.53571
treatment2        301            0        0.36810
treatment2        302            0        0.35714
treatment2        328            0        0.29762
treatment2        378            0        0.23810
treatment2        378            1        0.23810
treatment2        378            1        0.23810
treatment2        382            0        0.11905
treatment2        397            1

;
run;
proc sort data=_survival;
  by  timefailure doselev ;
run;
data _censorP1    (keep=timefailure survival rename=(survival=C1))
     _censorP2    (keep=timefailure survival rename=(survival=C2))
     _surv1             (keep=timefailure survival rename=(survival=S1))
     _surv2             (keep=timefailure survival rename=(survival=S2));
  set _survival;
  by timefailure;
    if doselev='treatment1' and _censor_=1 then output _censorP1;
    if doselev='treatment2' and _censor_=1 then output _censorP2;
    if doselev='treatment1' then output _surv1;
    if doselev='treatment2' then output _surv2;
run;
```

```
*--create annotate---*;
*====================*;
proc sql;
  *-count number of censored patient by timefailure--*;
  create table _censorP1b as select put(count(*),2.) as nb_txt,*
    from _censorP1
    group by timefailure,C1;
  create table _censorP2b as select put(count(*),2.) as nb_txt,*
    from _censorP2
    group by timefailure,C2;
quit;
*--assign annotate macro facilities---*;
%annomac;

data anno1;
  length color $8;
  set _censorP1b _censorP2b;
  *--coordinate system: absolute values area--*;
  %system(2,2);
  *--label the point with the number of censored patients--*;
    if C1 ne . then do;
           %label(timefailure,C1,nb_txt,stg,0,0,1.5,zapf,C);end;
    if C2 ne . then do;
           %label(timefailure,C2,nb_txt,biyg,0,0,1.5,zapf,C);end;
run;

*--create dataset for the proc Gplot (contains all required variables)
    for curves and dot plot---*;
data _pre_plot;
  merge _censorP1 _censorP2 _surv1 _surv2;
  by timefailure;
run;
proc sort data=_pre_plot nodupkey;
  by timefailure C1 C2 S1 S2;
run;

goptions reset=all
         display
         cback=white
         ftext=simplex
         ftitle=swiss
         colors=(black)
         gunit=pct
         ctext=black
         htext=3
         htitle=2;

axis1      offset=(2,2) label=(h=2.5 c=green f=zapfbi "Time to Treatment Failure
(days) " )
           value=(h=2);
axis2      offset=(0,0) label=(h=2.5 angle=90 c=green f=zapfbi
           "Survival Distribution Function Estimate") value=(h=2) offset=(2,2);

legend1 label=none position=(top right inside) mode=share across=1
           value=(h=2.5 f=zapfi c=stg 'treatment1' 'Censored event (+ nb of censored
patients)' c=biyg 'treatment2' 'Censored event (+ nb of censored patients)') ;

symbol1 i=stepjr value=none line=2 color=stg width=2 h=2;
symbol2 i=none value=dot color=stg width=3 h=3;
symbol3 i=stepjr value=none line=1 color=biyg width=2 h=3;
symbol4 i=none value=dot color=biyg width=3 h=3;

title1 j=c h=4 c=green "Figure X: Survival plot";
```

```
proc gplot data=_pre_plot gout=graf;
 plot S1*timefailure C1*timefailure
      S2*timefailure C2*timefailure
                 /   overlay vaxis=axis2 haxis=axis1
                        legend=legend1
                        vref=(&vref2,&vref1) cv=bwh lv=20
                        href=(&href2,&href1) ch=bwh lh=20
                        annotate=anno1;
run;
quit;
```

THE GRAPHICS OUTPUT

Figure X: Survival plot



EXAMPLE 8

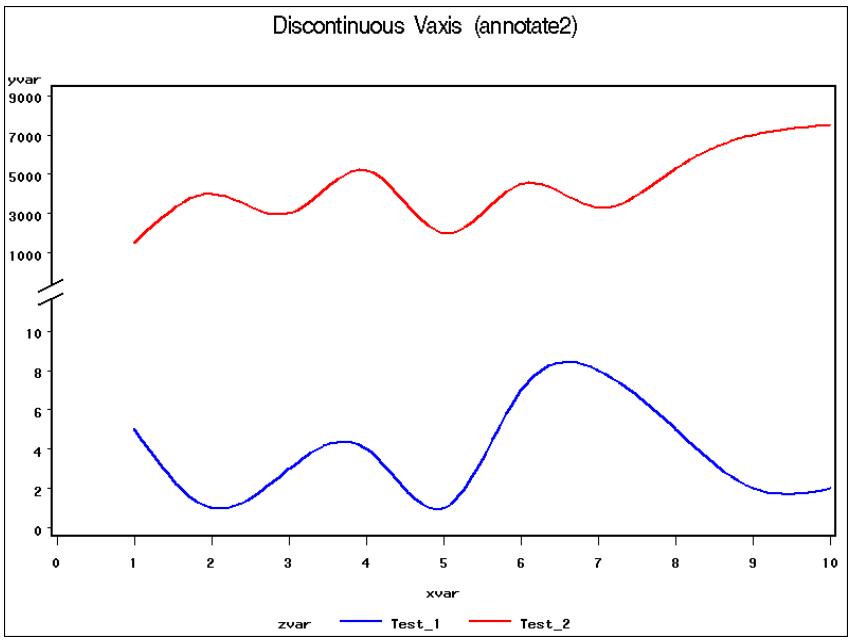
Another possibility with the Annotate facility is to visualize a discontinuation in the axis X or Y. The solution is to create a white solid box on the axis and then add a draw that could symbolize the discontinuation of the axis. Here is an example.

THE PROGRAM

```
data a;
input xvar yvar zvar $;
cards;
1 5 Test_1
2 1 Test_1
3 3 Test_1
4 4 Test_1
5 1 Test_1
6 7 Test_1
7 8 Test_1
8 5 Test_1
9 2 Test_1
```

```
10 2 Test_1
1 1500 Test_2
2 4000 Test_2
3 3000 Test_2
4 5200 Test_2
5 2000 Test_2
6 4500 Test_2
7 3300 Test_2
8 5300 Test_2
9 7000 Test_2
10 7500 Test_2
;
run;
data anno2;
    length function style color $8;
    *--define when to draw annotate: A='After graph' B='before graph'--;
    retain  when 'a';
    *--Draw a solid white box to cover the vaxis area where the
       break occurs.*;
    %annomac;
    %system(5,2);
    %bar(1,42,8,200,white,0,solid);output;  *change into white*;
    * Now draw the jagged line where the vaxis breaks.  *;
    %*system(B,B);
    %line(4,36.5,7,49,black,1,2);output; *change into black*;
    %line(4,50,7,350,black,1,2);output; *change into black*;
run;

axis1 order=(0 to 10 by 2, 50, 1000 to 10000 by 2000)
      minor=none width=2;
* Produce the plot with annotate2--;
title1 h=1.5 'Discontinuous Vaxis (annotate2)';

proc gplot data=a;
plot yvar*xvar=zvar / vaxis=axis1 haxis=axis2 anno=anno2;
run;
quit;
```

THE GRAPHICS OUTPUT



CONCLUSION

Creating graphics output using SAS® will always be time consuming but thanks to the Annotate facilities, it gives you tools to generate lots of types of graphics. With a little bit of experience, your programming efficiency will considerably increase and SAS/GRAPH® won't be such a complicated topic.

REFERENCES

SAS online doc V8

CONTACT INFORMATION (HEADER 1)

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

Sandrine STEPIEN
QUINTILES
4 route de la rivière
67380 LINGOLSHEIM
Phone: +33(0)3 88 77 44 13
Email: Sandrine.stepien@quintiles.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.