

SESUG Paper 063-2017
**Behind the Scenes: from Data to Customized Swimmer Plots
 Using SAS® Graphical Template Language (GTL)**

Rita Tsang, ICON Clinical Research
 Hima Bhatia, ICON Clinical Research

ABSTRACT

A Swimmer plot is a useful visual tool to assess treatment efficacy for individual subjects in oncology clinical trials. These efficacy endpoints may include tumor response evaluation over time as measured by Response Evaluation Criteria in Solid Tumors (RECIST), or percent change from baseline in target lesions for measurable disease as defined by the presence of at least one measurable lesion. These endpoints are always juxtaposed with other valuable information, such as treatment doses, duration of treatment, primary tumor type, and treatment status to create a compelling visual presentation.

There are many industry publications describing how to create swimmer plots in SAS. They provide a useful framework for using either SGPLOT or GTL to generate swimmer plots. This paper will build on the flexibility of SAS GTL to go further behind the scenes to illustrate how to prepare the data structure and variables for a customized swimmer plot. Focus will be placed on using dynamic programming for addressing missing data. The SGRENDER procedure will be used to create the graph based on the prepared data set and the graph template created in PROC TEMPLATE.

KEY WORDS

Swimmer Plot, Oncology Clinical Trial, SAS GTL.

INTRODUCTION

A swimmer plot may display individual patient data by dose group and primary tumor type (below). The bars represent the duration of treatment in weeks. Annotation can be added for the following information: tumor response assessment (such as CR, PR, SD, PD) over time, percent change at maximum reduction from baseline in target lesions, and treatment status. This paper will show how to prepare the data structure and variables referenced by the graph template and the SGRENDER procedure.

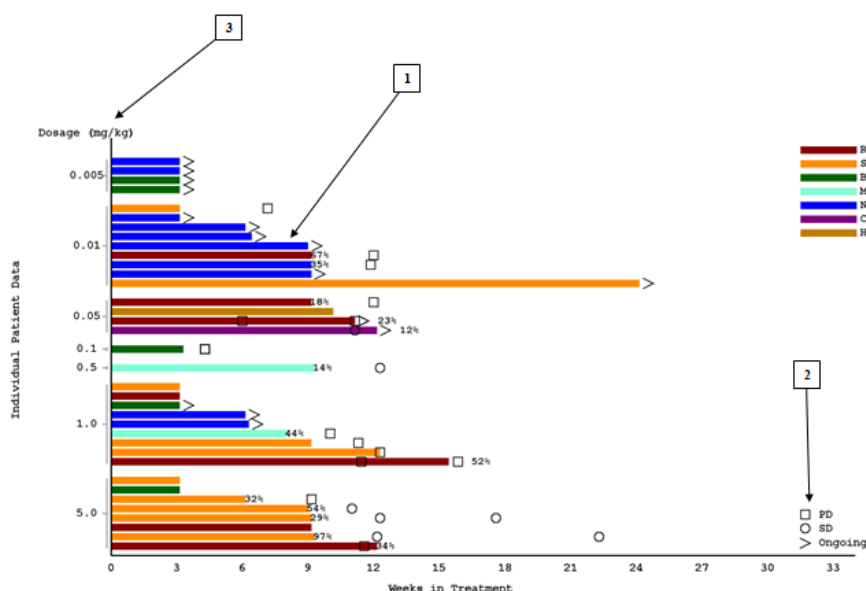


Figure 1. A Swimmer Plot

There are three steps to put together the swimmer plot:

Step 1: Create bars for treatment duration by dose group and primary tumor type for individual subjects. A blank line is added between each dose group. The VECTORPLOT and DISCRETELEGEND statements will be used in PROC TEMPLATE.

Step 2: Annotate the tumor response over time, treatment status, and the percent change at maximum reduction from baseline in target lesion for measurable disease. The SCATTERPLOT and DISCRETELEGEND statements in PROC TEMPLATE will be used for the annotation.

Step 3: Annotate the dose group values and label on the Y-axis. In addition, lines will be drawn for grouping each dose group and for the X and Y axes. The SCATTERPLOT and VECTORPLOT statements will be used.

GTL

Here is the skeleton of the GTL code used for creating the swimmer plot. The style template used by ODS RTF is included in **APPENDIX 2**.

```
ODS RTF file="C:\Users\tsangc\paper2017\step1.rtf" style=PR0128pt1;

ods graphics / reset=all border=off width=10in height=6.5in ;

proc template;
  define statgraph sgplot;
    begingraph /;

    ..... more steps here .....

  endgraph;
end;
run;

proc sgrender data=step3 template=sgplot;
run;

ods rtf close;
```

This is our style template which controls the graph attributes, such as font size, color, and margin.

The graph template SGPLOT is created in PROC TEMPLATE and is then called in PROC SGRENDER.

Note that the structure starts with the DEFINE STATGRAPH/END and BEGINGRAPH/ENDGRAPH statements.

This is our data set called by PROC SGRENDER.

Display 2. GTL

STEP 1:

Create bars for treatment duration by dose group and primary tumor type for individual subjects. A blank line is added between each dose group.

We have the following variables in our step 1 data:

USUBJID - unique subject identification number

TRT01P - planned treatment in text

TRT01PN – planned treatment in numeric format with values from 1 to 7

PRITMTYP - primary tumor type

TRTDURW - treatment duration in week

SCSTAT - treatment status with values of blank or "ONGOING"

AVALC - percent change at maximum reduction from baseline in target lesion (unique value per subject)

ADW - tumor response assessment time from date of first dose in weeks over time, e.g. subject 004 has assessments at week 12 and week 22.

OVRLRESP – the overall tumor response assessment using RECISIT over time, e.g. subject 006 has a SD assessment at week 12 and another SD assessment at week 22.

	USUBJID	TRT01P	TRT01PN	PRITMTYP	TRTDURW	SCSTAT	AVALC	ADW	OVRLRESP	ID
1	502	5.0	7	R	13		34%	12	PD	1
2	505	5.0	7	R	10		83%	.		2
3	004	5.0	7	S	10		97%	12	SD	3
4	004	5.0	7	S	10		97%	22	SD	3
5	006	5.0	7	S	10		29%	12	SD	4
6	006	5.0	7	S	10		29%	18	SD	4
7	007	5.0	7	S	10		54%	11	SD	5
8	403	5.0	7	S	7		32%	9	PD	6
9	405	5.0	7	B	4		70%	.		7
10	008	5.0	7	S	4		56%	.		8
11			.	S	.			.		9
12	504	1.0	6	R	16		52%	11	PD	10
13	504	1.0	6	R	16		52%	16	PD	10
14	413	1.0	6	S	13		74%	12	PD	11
15	401	1.0	6	S	10		68%	11	PD	12

Display 2. Snapshot of the STEP1 data

Note that the data set is sorted by descending treatment group (TRT01PN) and treatment duration (TRTDURW). An additional record is added between each dose group. A new record identifier variable (ID) is added based on the sorting order. The ID variable is used for the Y-axis display. Please reference Appendix 1 for the code for generating the STEP1 data.

PROC SGRENDER will call the STEP1 data and the SGPLOT graph template. In PROC TEMPLATE, the VECTORPLOT statement will be used to create the horizontal bars based on treatment duration (TRTDURW) for the X axis and the record identifier (ID) for the Y axis. These values are also grouped by primary tumor type (PRITMTYP). Note that the top right legend for the grouped variable is created by the DISCRETELEGEND statement.

```
VectorPlot X=trtdurw Y=id XOrigin=0 YOrigin=id/group=pritmtyt Lineattrs=(Pattern=1
Thickness=7px) ARROWHEADS=FALSE NAME="type";
```

```
DiscreteLegend "type" / Location=inside across=1 autoAlign=(topright)
Title=" " Border=false valueattrs=(SIZE=8pt);
```

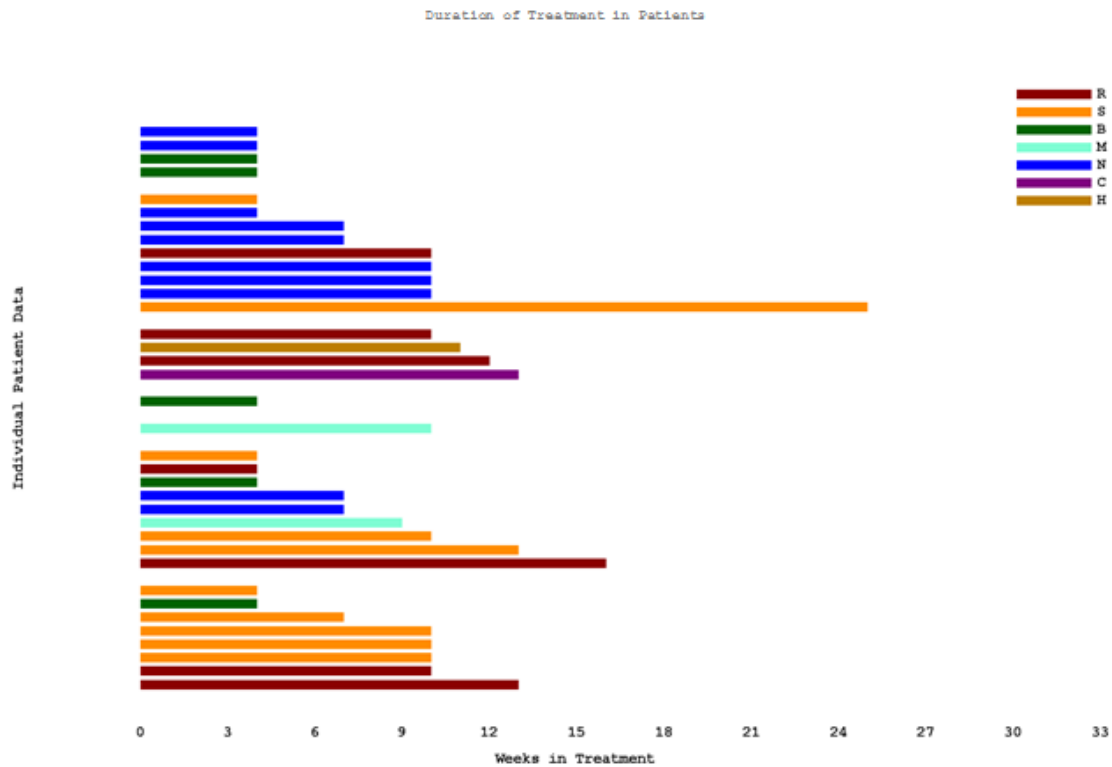


Figure 2. Step 1 Graph

STEP 2:

Annotate the tumor response over time, treatment status, and the percent change at maximum reduction from baseline in target lesion for measurable disease.

step2

	USUBJID	TRT01P	TRT01PN	PRITMTYP	TRTDURW	SCSTAT	AVALC	ADW	OVRLRESP	ID	PD	SD	MAXX	X1	XE
1	502	5.0	7	R	13		34%	12	PD	1	10	.	13	13.4	.
2	505	5.0	7	R	10		83%	.		2	.	.	10	10.4	.
3	004	5.0	7	S	10			12	SD	3	.	12	10	.	.
4	004	5.0	7	S	10		97%	22	SD	3	.	22	10	10.4	.
5	006	5.0	7	S	10			12	SD	4	.	12	10	.	.
6	006	5.0	7	S	10		29%	18	SD	4	.	18	10	10.4	.
7	007	5.0	7	S	10		54%	11	SD	5	.	11	10	10.4	.
8	403	5.0	7	S	7		32%	9	PD	6	9	.	7	7.4	.
9	405	5.0	7	B	4		70%	.		7	.	.	4	4.4	.
10	008	5.0	7	S	4		56%	.		8	.	.	4	4.4	.
11			.	S	.			.		9	.	.	.	.	.
12	504	1.0	6	R	16			11	PD	10	11	.	16	.	.
13	504	1.0	6	R	16		52%	16	PD	10	16	.	16.5	16.9	.
14	413	1.0	6	S	13		74%	12	PD	11	12	.	13	13.4	.
15	401	1.0	6	S	10		68%	11	PD	12	11	.	10	10.4	.
16	301	1.0	6	M	9		44%	10	PD	13	10	.	9	9.4	.
17	105	1.0	6	N	7	ONGOING	58%	.		14	.	.	7	7.4	8
18	115	1.0	6	N	7	ONGOING	62%	.		15	.	.	7	7.4	8
19	114	1.0	6	B	4	ONGOING	61%	.		16	.	.	4	4.4	5
20	414	1.0	6	R	4		75%	.		17	.	.	4	4.4	.
21	503	1.0	6	S	4		81%	.		18	.	.	4	4.4	.
22			.	S	.			.		19	.	.	.	.	.
23	003	0.5	5	M	10		14%	12	SD	20	.	12	10	10.4	.
24			.	M	.			.		21	.	.	.	.	.

Display 3. Snapshot of the STEP2 data

Here are the additional derived variables in STEP2:

PD – lesion assessment in week for PD (ADW when OVRLRESP =PD)

SD – lesion assessment in week for SD (ADW when OVRLRESP =SD)

MAXX – is equivalent to treatment duration (TRTDURW), an adjustment will be made for the overlapping time by adding a value of 0.5 (TRTDURW+0.5). For example, subject 504 has 16 weeks of treatment (TRTDURW=16) and a PD assessment at week 16 (PD=16). The MAXX value for the subject is adjusted to 16.5 to avoid overlapping of the annotation. MAXX will be used for deriving the time on the X axis for percent change and treatment status.

X1 – is the time for percent change at maximum reduction from baseline (AVALC). It is derived as MAXX + 0.4 when AVALC value is available.

XE – is the time for treatment status (SCSTAT). It is derived as MAXX + 1 when SCSTAT value is “ONGOING”.

Missing data for tumor response is a common occurrence. We have built in dynamic code using PROC SQL and macro do-loop as shown below to ensure that accomodation can be made for any missing response data. Variables for lesion duration (CR , PR, SD, and PD) will only be created based on the availability of the response assessment data.

```
proc sql noprint;
  select distinct ovrlresp into: resp separated by " " from step1;
quit;

%put &resp;

%let r=1;
data step2;
  set step1;
  by usubjid;
  if not (first.usubjid and last.usubjid) then do;
    if first.usubjid then avalc='';
  end;
  maxx=trtdurw;
  %do %while (%scan(&resp,&r) ne );
    if ovrlresp="%scan(&resp,&r)" then %scan(&resp,&r)=adw;
    if maxx>. and %scan(&resp,&r)=maxx then maxx=maxx+0.5;
    %let r=&r+1;
  %end;
  if avalc>' ' then x1=maxx+0.4;
  if scstat='ONGOING' then xe=maxx+1;
run;
%let r=1;
```

In PROC TEMPLATE, the SCATTERPLOT statements are used to create annotation for the tumor response over time (SD and PD), treatment status (XE), and the percent change at maximum reduction from baseline (X1). The DISCRETELEGEND statement will be used to create the legend for these endpoints. Note that dynamic programming is also put in place in PROC TEMPLATE for the potential missing tumor response data as shown below.

```
ScatterPlot X=x1 Y=id / MarkerCharacter=avalc MARKERCHARACTERATTRS
=(family='Courier New' size=7pt weight=bold) NAME="SCATTER";
%do %while (%scan(&resp,&r) ne );
  %if %scan(&resp,&r)=CR %then %do;
    ScatterPlot X=cr Y=id / Markerattrs=(color=black Symbol=diamond
    Size=9px) LegendLabel="CR" NAME="cr";
```

```

%end;
%if %scan(&resp,&r)=PR %then %do;
    ScatterPlot X=pr Y=id / Markerattrs=(color=black Symbol=triangle
    Size=9px) LegendLabel="PR" NAME="pr";
%end;
%if %scan(&resp,&r)=SD %then %do;
    ScatterPlot X=sd Y=id / Markerattrs=(color=black Symbol=square
    Size=9px) LegendLabel="SD" NAME="sd";
%end;
%if %scan(&resp,&r)=PD %then %do;
    ScatterPlot X=pd Y=id / Markerattrs=(color=black Symbol=circle
    Size=9px) LegendLabel="PD" NAME="pd";
%end;
%let r=%eval(&r+1);
%end;
%let r=1;
ScatterPlot X=xe Y=id / Markerattrs=(color=black Symbol=greaterthan
Size=10px) LegendLabel="Ongoing" NAME="on";
DiscreteLegend
%do %while (%scan(&resp,&r) ne );
    %if %scan(&resp,&r)=CR %then %do;
        "cr"
    %end;
    %if %scan(&resp,&r)=PR %then %do;
        "pr"
    %end;
    %if %scan(&resp,&r)=SD %then %do;
        "sd"
    %end;
    %if %scan(&resp,&r)=PD %then %do;
        "pd"
    %end;
    %let r=%eval(&r+1);
%end;
"on" / Location=inside across=1 autoAlign=(bottomright) Title=" "
Border=false valueattrs=(SIZE=8pt);

```

Please reference **APPENDIX 1** for the complete PROC TEMPLATE code.

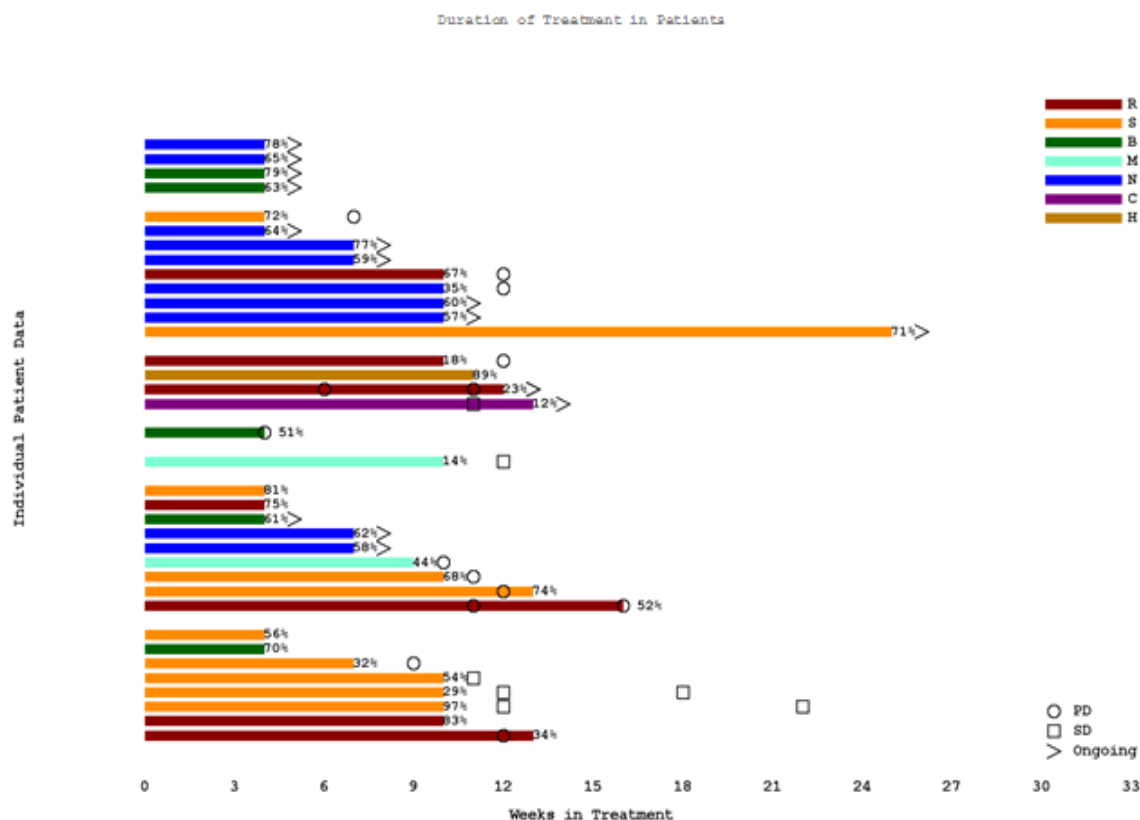


Figure 3. Step 2 Graph

STEP 3:

Annotate the dose group values and label on the Y-axis. In addition, lines will be drawn for grouping each dose group and for the X and Y axes.

	USUBJID	TRT01P	TRT01P N	ID	TRT2	XPT	YMAX	XMAX	Y0	X0	_XLS	_XLE	_YLS1	_YLE1	_Y1
33	407	0.01	2	29		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
34	104	0.01	2	30		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
35	113	0.01	2	31		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
36	509	0.01	2	32		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
37	506	0.01	2	33		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
38	106	0.01	2	34		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
39	419	0.01	2	35		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
40	123	0.01	2	36		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
41	409	0.01	2	37		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
42			.	38		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
43	121	0.005	1	39		-1.1	44.5	34	-1	0	-0.2	-0.4	38.8	.	.
44	421	0.005	1	40		-1.1	44.5	34	-1	0	-0.2	-0.4	38.8	.	.
45	126	0.005	1	41		-1.1	44.5	34	-1	0	-0.2	-0.4	38.8	.	.
46	420	0.005	1	42		-1.1	44.5	34	-1	0	-0.2	-0.4	38.8	42.2	40.5
47			.	43		-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.
48			.	45	Dosage (mg/kg)	-1.1	44.5	34	-1	0	-0.2	-0.4	.	.	.

Display 4. Snapshot of the STEP3 data

Additional variables for annotating the dose group label and values:

TRT2 – set to “Dosage (mg/kg)” for the additional record. The ID value is set to the maximum ID+2, i.e. 43 + 2 =45.

XPT – set to an X value of -1.1. A negative value is used because it is smaller than 0 on the X axis.
 Note that the dose values will be annotated at the X value of XPT and the midpoint ($_Yi$) of each dose group on the Y axis.

Additional variables for drawing the X and Y axes:
 YMAX – set to the maximum ID value, i.e. $43 + 1.5 = 44.5$.
 XMAX – set to the maximum X value, i.e. $33 + 1 = 34$.
 Y0 – set to a Y value of -1.
 X0 – set to an X value of 0.

Additional variables for drawing a short horizontal line at the midpoint ($_Yi$) of each dose group:
 $_XLS$ – set to an X value of -0.2.
 $_XLE$ – set to an X value -0.4.

Additional variables for drawing a vertical line for each dose group:
 $_YLS1$ – the minimum Y value for TRT01PN=1, derived as the first ID value – 0.2, i.e. $39 - 0.2 = 38.8$.
 $_YLE1$ – the maximum Y value for TRT01PN=1, derived as the last ID value + 0.2, i.e. $42 + 0.2 = 42.2$.
 $_Y1$ – the mid-point of the Y value range for TRT01PN=1, derived as the first ID value + (the last ID value – the first ID value)/2, i.e. $39 + (42 - 39)/2 = 40.5$.

Note that $_YLSi$, $_YLEi$, and $_Yi$ where $i=1$ to 7 (maximum planned treatment value), will also be derived based on the total number of treatment groups in a dynamic fashion as show below:

```
proc sql noprint;
  select max(trt01pn) into: t from step3;
quit;

data step3;
  set step3;
  by trt01pn id;
  %do i=1 %to &t;
    if trt01pn=&i then do;
      if first.trt01pn and last.trt01pn then do;
        _yls&i=id-0.2;
        _yle&i=id+0.2;
        _y&i=id;
      end;
      if not (first.trt01pn and last.trt01pn) then do;
        if first.trt01pn then do;
          retain _yls&i .;
          _yls&i=id-0.2;
        end;
        if last.trt01pn then do;
          _yle&i=id+0.2;
          _y&i=_yls&i+ (_yle&i-_yls&i)/2;
        end;
      end;
    end;
  %end;
run;
```

We will use the SCATTERPLOT statements to annotate the dose group label (TRT2) and the planned treatment group values (TRT01P) in PROC TEMPLATE. Using treatment group 1 as an example, the dose value will be annotated at the midpoint value of treatment group 1 ($_Y1$).

ScatterPlot X=xpt Y=id / MarkerCharacter=trt2 MARKERCHARACTERATTRS=
 (family='Courier New' size=8pt weight=bold);

```
ScatterPlot X=xpt Y=_y1 / MarkerCharacter=trt01p MARKERCHARACTERATTRS=
(family='Courier New' size=8pt weight=bold);
```

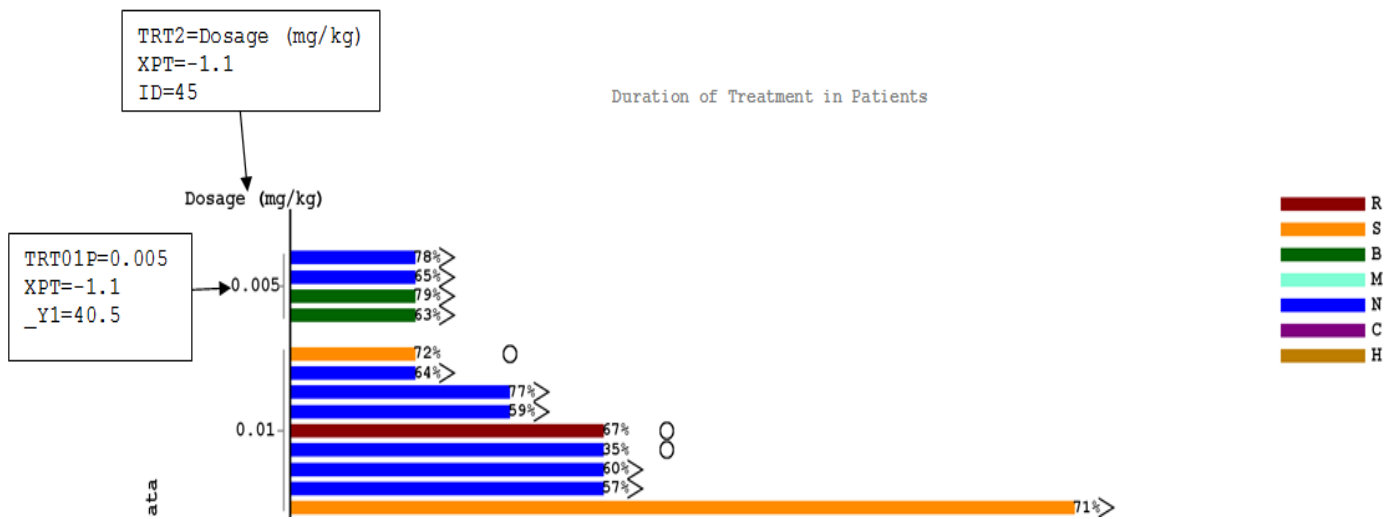


Figure 4. Annotating the Dosing Labels and Values

In addition to creating bars, the VECTORPLOT statements will be also used to draw lines by adjusting the thickness (THICKNESS=). Lines in grey will be added for grouping the planned treatment on the Y axis. Using treatment group 1 as an example, the first VECTORPLOT statement will draw a vertical line in grey from two set of X, Y coordinates (_xls, _yls1) to (_xls, _yle1), i.e. (-0.2, 38.8) to (-0.2, 42.2). The second statement will draw a short horizontal line from _xls (-0.2) to _xle (-0.4) at the midpoint value (_y1=40.5).

```
VectorPlot X=_xls Y=_yle1 XOrigin=_xls YOrigin=_yls1 / Lineattrs=(Color=grey
Pattern=1 Thickness=1px) Arrowheads=false;
```

```
VectorPlot X=_xle Y=_y1 XOrigin=_xls YOrigin=_y1 / Lineattrs=(Color=grey Pattern=1
Thickness=1px) Arrowheads=false;
```

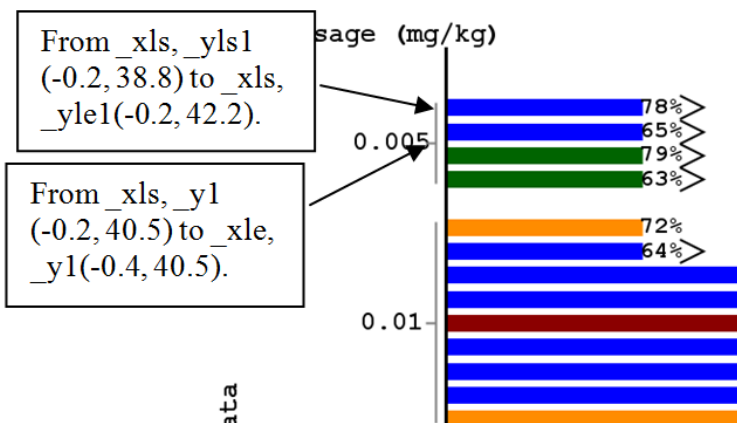


Figure 5. Drawing the Grey Lines for Grouping Purpose

Furthermore, the VECTORPLOT statements will be used for drawing a line in black for each axis. The first statement will draw a line for the Y axis, from x0, y0 (0, -1) to x0, ymax (0, 44.5). The second statement will draw a line for the X axis, from x0, y0 (0, -1) to xmax, y0 (34, -1).

```
VectorPlot X=x0 Y=ymax XOrigin=x0 YOrigin=y0 / Lineattrs=(Color=black Pattern=1
Thickness=1px) Arrowheads=false ;
```

```
VectorPlot X=xmax Y=y0 XOrigin=x0 YOrigin=y0 / Lineattrs=(Color=black Pattern=1
Thickness=1px) Arrowheads=false ;
```

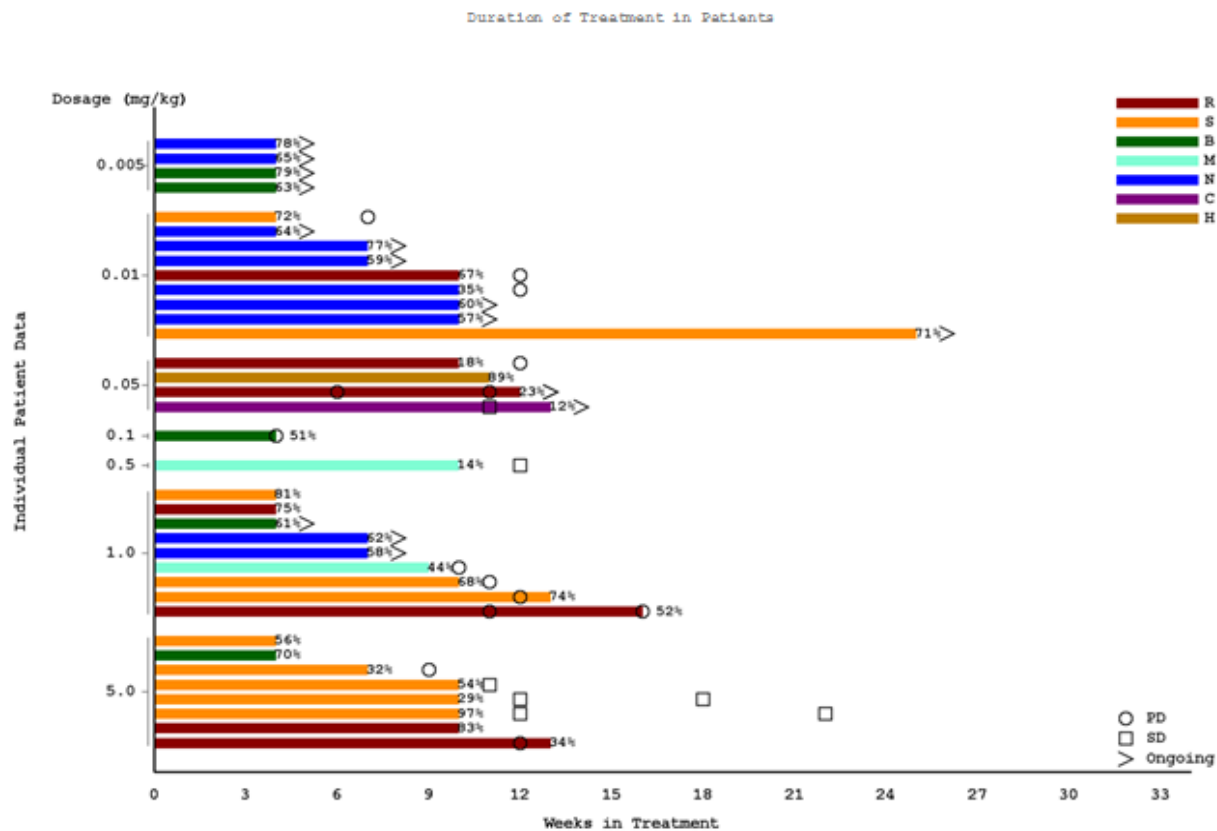


Figure 6. The Complete Swimmer Plot

CONCLUSION

This paper shows how to create a customized swimmer plot in three simple steps using SAS Graph Template Language. Focus has been placed on how to prepare the data structure and variables using dynamic programming to accommodate for missing data in oncology trial. A well-prepared data set is an important first step to create a customized swimmer plot using PROC TEMPLATE and PROC SGRENDER. SAS GTL is a versatile tool for generating any professional graphics if one is willing to go behind the scenes to explore and prepare the data.

REFERENCE

- Phillips, S. (2014), "Swimmer Plot: Tell a Graphical Story of Your Time to Response Data Using PROC SGPLOT", PharmaSUG 2014.
- Huang, J. (2016), "A Different Approach to Create Swimmer Plot Using Proc Template and SGRENDER", PharmaSUG 2016.
- Mantange, S. (2016), "Clinical Graphs Using SAS®", PhUSE 2016.
- SAS Institute Inc. (2014), "SAS 9.4 Graph Template Language: User's Guide, Third Edition", Cary, NC, USA
- Wang, B. (2016), "Swimmer Plot by Graphic Template Language (GTL)", PharmaSUG 2016.

APPENDIX 1

Here is the complete code for creating our swimmer plot.

```
%inc "C:\Users\tsangc\paper2017\figtemplate1.sas";
libname libanal "C:\Users\tsangc\paper2017";
title1 "Duration of Treatment in Patients";

%macro step3;
  proc sort data=libanal.adsl out=adsl;
    by descending trt01pn descending trtdurw pritmtyp usubjid;
  run;

  data adsl;
    set adsl;
    by descending trt01pn descending trtdurw pritmtyp ;
    output;
    if last.trt01pn then do;
      avalc=' '; usubjid=' '; trt01pn=.; trt01p=''; scstat=' ';
      adw=.; trtdurw=.; output;
    end;
  run;

  data adsl;
    set adsl;
    id=_n_;
  run;

  proc sort data=adsl;
    by usubjid;
  run;

  data step1;
    merge adsl (in=a) libanal.adrs ;
    by usubjid;
    if a;
  run;

  proc sql noprint;
    select distinct ovrresp into: resp separated by " " from step1;
  quit;
  %put &resp;

  %let r=1;
  data step2;
    set step1;
    by usubjid;
    if not (first.usubjid and last.usubjid) then do;
      if first.usubjid then avalc='';
    end;
    maxx=trtdurw;
    %do %while (%scan(&resp,&r) ne );
      if ovrresp="%scan(&resp,&r)" then %scan(&resp,&r)=adw;
      if maxx>. and %scan(&resp,&r)=maxx then maxx=maxx+0.5;
      %let r=&r+1;
    %end;
    if avalc>' ' then x1=maxx+0.4;
    if scstat='ONGOING' then xe=maxx+1;
  run;
```

Step 1 Data

Step 2 Data

```

%let r=1;

proc sql noprint;
  select max(id) into: maxy from step2 where id>.;
  select max(round(max(trtdurw, adw)))- mod(max(round(max(trtdurw, adw))), 3) + 9
  into: maxx from step2 where trtdurw>.;
quit;

data _null_;
  length list list1 $100;
  do i=1 to %eval(&maxy+1);
    list=strip(list)||' '||strip(put(i,best.));
  end;
  do i=3 to %eval(&maxx) by 3;
    list1=strip(list1)||' '||strip(put(i,best.));
  end;
  call symput('ylist',strip(list));
  call symput('xlist',strip(list1));
run;

data step3;
  set step2 end=eof;
  if eof then do;
    call symput('pt',pritmtyp);
  end;
run;

data filler;
  id=%eval(&maxy+2);
  trt2='Dosage (mg/kg)';
  pritmtyp="&pt";
run;

data step3;
  set step3 filler;
  ymax=%sysevalf(&maxy+1.5);
  xpt=-1.1;
  x0=0;
  y0=-1;
  xmax=%eval(&maxx+1);
  _xle=-0.4;
  _xls=-0.2;
run;

proc sort data=step3;
  by trt01pn id;
run;

proc sql noprint;
  select max(trt01pn) into: t from step3;
quit;

data step3;
  set step3;
  by trt01pn id;
  %do i=1 %to &t;
    if trt01pn=&i then do;
      if first.trt01pn and last.trt01pn then do;
        _yls&i=id-0.2;
      end;
    end;
  %end;

```

Step 3 Data

```

        _yle&i=id+0.2;
        _y&i=id;
    end;
    if not (first.trt01pn and last.trt01pn) then do;
        if first.trt01pn then do;
            retain _yls&i .;
            _yls&i=id-0.2;
        end;
        if last.trt01pn then do;
            _yle&i=id+0.2;
            _y&i=_yls&i+ (_yle&i-_yls&i)/2;
        end;
    end;
end;
%end;
run;

```

Step 3 Data

```

data step3;
    set step3;
    by trt01pn id;
    %do i=1 %to &t;
        if _yls&i>. and trt01pn ne &i then _yls&i=.;
    %end;
run;

```

```

proc sort data=step3;
    by id;
run;

```

```

option orientation=landscape;

```

```

ODS RTF file="C:\Users\tsangc\paper2017\final.rtf" style=PR0128pt1;
ods graphics / reset=all border=off width=10in height=6.5in ;

```

```

proc template;
    define statgraph sgplot;
        begingraph /;
            layout overlay /xaxisopts=(label='Weeks in Treatment'
                labelattrs=(size=8pt) display=(tickvalues label)
                offsetmin=0.1 type=linear
                linearopts=( tickvaluelist=( 0 &xlist)
                viewmin=0 viewmax=&maxx )
                TICKVALUEATTRS=(SIZE=3PX))

                yaxisopts=( label='Individual Patient Data'
                labelattrs=(size=8pt) display=(label) type=linear
                linearopts=(viewmin=0 viewmax=%eval(&maxy+3)));

```

Step 1

```

        VectorPlot X=trtdurw Y=id XOrigin=0 YOrigin=id/group=pritmtyp
        Lineattrs=(Pattern=1 Thickness=7px) ARROWHEADS=FALSE NAME="type";

        DiscreteLegend "type" / Location=inside across=1 autoAlign=(topright)
        Title=" " Border=false valueattrs=(SIZE=8pt);

```

Step 2

```

        %do %while (%scan(&resp,&r) ne );
            %if %scan(&resp,&r)=CR %then %do;
                ScatterPlot X=cr Y=id / Markerattrs=(color=black Symbol=diamond
                Size=9px) LegendLabel="CR" NAME="cr";
            %end;
        %end;

```

```

    %if %scan(&resp,&r)=PR %then %do;
        ScatterPlot X=pr Y=id / Markerattrs=(color=black Symbol=triangle
            Size=9px)LegendLabel="PR" NAME="pr";
    %end;
    %if %scan(&resp,&r)=SD %then %do;
        ScatterPlot X=sd Y=id / Markerattrs=(color=black Symbol=square
            Size=9px)LegendLabel="SD" NAME="sd";
    %end;
    %if %scan(&resp,&r)=PD %then %do;
        ScatterPlot X=pd Y=id / Markerattrs=(color=black Symbol=circle
            Size=9px)LegendLabel="PD" NAME="pd";
    %end;
    %let r=%eval(&r+1);
%end;
%let r=1;

ScatterPlot X=x0 Y=id / Markerattrs=(color=black Symbol=
Greaterthan Size=10px) LegendLabel="Ongoing" NAME="on";

Step 2 ScatterPlot X=x1 Y=id / MarkerCharacter=avalc MARKERCHARACTERATTRS
=(family='Courier New' size=7pt weight=bold) NAME="SCATTER";

DiscreteLegend
%do %while (%scan(&resp,&r) ne );
    %if %scan(&resp,&r)=CR %then %do;
        "cr"
    %end;
    %if %scan(&resp,&r)=PR %then %do;
        "pr"
    %end;
    %if %scan(&resp,&r)=SD %then %do;
        "sd"
    %end;
    %if %scan(&resp,&r)=PD %then %do;
        "pd"
    %end;
    %let r=%eval(&r+1);
%end;
"on" / Location=inside across=1 autoAlign=(bottomright) Title=" " Border=
False valueattrs=(SIZE=8pt);

```

```

ScatterPlot X=xpt Y=id / MarkerCharacter=trt2 MARKERCHARACTERATTRS=
(family='Courier New' size=8pt weight=bold);

%do i=1 %to &t;
    ScatterPlot X=xpt Y=_y&i / MarkerCharacter=trt01p MARKERCHARACTERATTRS
    =(family='Courier New' size=8pt weight=bold);

    VectorPlot X=_xls Y=_yle&i XOrigin=_xls YOrigin=_yls&i /
    Lineattrs=(Color=grey Pattern=1 Thickness=1px) Arrowheads=false;

    VectorPlot X=_xle Y=_y&i XOrigin=_xls YOrigin=_y&i /
    Lineattrs=(Color=grey Pattern=1 Thickness=1px) Arrowheads=false;
%end;

VectorPlot X=x0 Y=ymax XOrigin=x0 YOrigin=y0 / Lineattrs=(Color=black
Pattern=1 Thickness=1px) Arrowheads=false ;
VectorPlot X=xmax Y=y0 XOrigin=x0 YOrigin=y0 / Lineattrs=(Color=black
Pattern=1 Thickness=1px) Arrowheads=false ;

```

```

        endlayout;
    endgraph;
end;
run;

proc sgrender data=step3 template=sgplot;
run;

ods rtf close;
%mend;

```

APPENDIX 2

Here is the code which creates the style template PR0128pt1.

```

%macro figtemplate1;

proc template;
    define style PR0128pt1;
        parent=styles.listing;

        class graphwalls / frameborder=off;

        style GraphData1 / ContrastColor=blue;
        style GraphData2 / ContrastColor=darkred;
        style GraphData3 / ContrastColor=darkgreen;
        style GraphData4 / ContrastColor=purple;
        style GraphData5 / ContrastColor=darkorange;
        style GraphData6 / ContrastColor=darkyellow;
        style GraphData7 / ContrastColor=AQUAMARINE;
        style GraphData8 / ContrastColor=darkbrown;

        style graphfonts from graphfonts /
            'GraphLabelFont' = ("Courier New",8pt,bold)
            'GraphValueFont' = ("Courier New",8pt,bold);

        replace fonts /
            'Titlefont'          = ("Courier New",8pt,Bold)
            'Titlefont2'         = ("Courier New",8pt,Bold)
            'Strongfont'         = ("Courier New",8pt)
            'EmphasisFont'       = ("Courier New",8pt)
            'FixedEmphasisFont'  = ("Courier New",8pt)
            'FixedStrongFont'    = ("Courier New",8pt)
            'FixedHeadingFont'   = ("Courier New",8pt)
            'BatchFixedFont'     = ("Courier New",8pt)
            'FixedFont'          = ("Courier New",8pt)
            'headingEmphasisFont' = ("Courier New",8pt,Bold Italic)
            'headingFont'        = ("Courier New",8pt,Bold)
            'docFont'            = ("Courier New",8pt,Bold)
            'FootnoteFont'       = ("Courier New",8pt,Bold);

        style table from table/
            rules=groups
            frame=void
            cellspacing=0.1mm
            cellpadding=0.1mm;
    enddefine;
end;

```

```

style PageNo from PageNo/
    font_size=0.1pt
    background=white
    foreground=white;

style BodyDate from BodyDate/
    foreground=white
    background=white
    font_size=0.1pt;

style Header from Header /
    Background = _undef_
    borderwidth = 0.25pt
    just = center
    frame = below
    rules = groups;

style SystemFooter from TitlesAndFooters
    'Controls system footer text.'/
    font = Fonts('FootnoteFont');

replace Body from Document
    "Controls the Body file. " /
    bottommargin = 0in
    topmargin    = 0.75in
    rightmargin  = 1in
    leftmargin   = 1in;
end;
run;

%mend;

```