

Help Your Fellow Colleagues: Customized SAS Graphs Could Provide a Peek into Subjects' Progress

Arun Lal, Grunenthal Gmbh, Aachen, Germany
Jari Ahvenainen, Grunenthal Gmbh, Aachen, Germany

Abstract

Many agree to the benefits of having graphical representations of a subject's profile in a clinical trial. SAS® provides a rich toolset to efficiently generate such visual reports. The new or old procedures like GPLOT, SGPLOT, SGPanel, etc. make such tasks easier even for those with little knowledge of SAS. Still the adoption of these procedures by non-SAS programmers has been low, and yet they remain highly interested in the results. Naturally, other companies have filled the gap. Many vendors provide with Patient Profile Viewer with varying degree of user friendliness. While the cost of buying such a tool might not be an issue for a bigger firm, it is a problem for the smaller firms. SAS programmers can step in to create highly customizable and specific modules that might not be very user-friendly but still serves as convenient tools for letting medical writers, CRAs, or physicians have a peek at a subject's progress.

Requests

Programmers are often responsible for producing tables, listings, and figures (TLFs) that are part of clinical trial reports (CTRs). In addition, they receive requests for interim analyses that mostly concern summary statistics. However, there are stakeholders for whom subject-level information is quite important. The clinicians use patients' profile reports to forecast adverse effects, adjust dosages etc. Others might use these for planning further trials or keeping abreast with the current status. The programmers at Grunenthal receive requests to generate subject-level reports from such stakeholders that often concern the following topics:

- A subject's exposure to the drug
- Reported pain scores
- Reported AEs with severity & intensity
- Demographic information
- Other endpoints' progression
- Specific vitals parameters

Solution

General Characteristics

The repeated nature of the requests prompted us to develop a SAS program that:

- produces graphical outputs in panels on a page
- is simple and can be managed (update/modify) with little SAS knowledge
- has the flexibility to adapt to a variety of clinical trials

- is modular in structure for easier modifications and extensions
- could be quickly adapted if new parameters are included later for similar analysis.

In-built Tools

SAS 9.2 comes with many pre-existing tools that make programming in SAS/GRAPH much easier. We often used `%DCLANNO`, `%SYSTEM`, `%LINE`, and `%LABEL` for the annotation datasets. These in-built macros had obvious benefits, like fewer lines of code, overall more understandable programs, etc. to list a few.

Custom Tools

On top of the in-built macro functions, we created a few new ones that could be invoked either independently or within the modules to plot certain items in a graph.

- `%TITLETXT` (TXT=, XIN=, YIN=) [For titles and subtitles]
- `%YLINE` [vertical lines indicating different phases in a trial]
- `%INFOBOX` (WHERE=, XIN=, YIN=) [General information e.g. demographics etc.]
- `%YXPLOT` (INDAT=, YVAR=, YVAR2=, XVAR=, VAX=, COND=, CATEG=, H=, V=, NAME=, USYMB=) [main plots, Y vs. X, Y1 & Y2 vs. X]
- `%XAXIS` (H=, V=) [adjustable x-axis]

Custom Modules

While the custom tools provide the flexibility to plot a few graphical items anywhere on the page and within any panel, we still needed a set of program modules, each encompassing a few parameters required on a panel. For example, the module for a patient's WOMAC scores focused on displaying the data observed from the three subscales and the calculated total along the time period, whereas the module for displaying the pain intensity included exposure information as well along the same time period. The customization requirements (symbols, interpolation, height, width, etc.) could be handled within these custom modules. It might be necessary to do some data processing beforehand in some cases. We developed the following modules based on the requirements we received from our colleagues:

- `%PLOT_AE` (WHERE=, H=, V=) [AE occurrences along the x-axis]
- `%PLOT_WOMAC` (SUBJ=) [plot WOMAC score for a subject]
- `%PLOT_NRSEXP` (SUBJ=) [plot NRS pain intensity score for a subject]
- `%PLOT_VITALS` (SUBJ=) [display chosen vital statistics parameters for a subject]

Discussion

The paneled display extensively used the coordinate and output sizing facility to place various graphical outputs on a single page. The use of a few global variables for positioning and sizing ensured that any change in one graphical item would cascade rather smoothly to the other items. The built-in macro, `%LABEL`, was quite instrumental in program debugging during the annotation. On the other hand, SAS sometimes generates insufficient ERROR messages upon encountering exceptions. These messages are usually not helpful in pinpointing the coding issues. A part of the program is included here in the appendix.

Advantages

In general, we achieved the characteristics that we set out to have in our SAS program. The users of the graphical reports found them very informative. For the programming department at Grunenthal, it is a good tool because it is simple and anyone from the team can rather quickly get used to it.

Next Steps

The program was developed at Grunenthal sometime ago. Since then the emphasis and awareness on the new SAS graphical procedures has considerably increased. It would be interesting to explore the new procedures and test if the same or even better results could be obtained with them. Moreover, the program currently takes long to generate the outputs per trial. Using quicker graphics options and/or modifying the program flow, etc could improve this.

Acknowledgement

We express our sincere thanks to Grunenthal Management for providing us with the time and resources to do this project.

Arun Lal (arun.lal@grunenthal.com)

Jari Ahvenainen (Jari.Ahvenainen@grunenthal.com)

Appendix

```

*****;
*      A macro defined for presenting demographic infobox in the left-middle *;
*      corner of the plot. *;
*      Limitation: 1. Demographic attributes on display are fixed. It could be *;
*                  easily modified. Only 50 percent of the allocated graphic *;
*                  area is used. *;
*                  2. The display could be moved either way, *;
*                  right-up-down by changing the XY coordinates, but is *;
*                  not advisable. It might result in graph overflow. *;
*                  *;
*      where = where condition to subset the KDEMOG dataset one subject every time*;
*      xin = graph origin in inches from left to right *;
*      yin = graph origin in inches from bottom up *;
*****;
%MACRO INFOBOX (WHERE=, XIN=0, YIN=3.5);

    DATA PATINFO_;
        SET KDEMOG;
        WHERE &WHERE;
        KEEP STUDYID1 SUBJID TRTGRP WEIGHTKG AGE
            AGEUNIT SEX RACE COUNTRY;

    RUN;

    DATA NRSSUM_;
        SET KNRSSUM_ (KEEP=SUBJID CHANGE PARAMC WINDOWC);
        WHERE &WHERE;
        KEEP SUBJID CHANGE WINDOWC;

    RUN;

    PROC TRANSPOSE DATA=NRSSUM_ OUT=NRS9899 PREFIX=WIN;
        BY SUBJID;
        ID WINDOWC;
        VAR CHANGE;

    RUN;

    DATA CGI_;
        SET KCGI_ (KEEP=SUBJID CGCHG CGCHG8 CGCHG10 CGCHG12);
        WHERE &WHERE;

    RUN;

    DATA PATINFO_;
        MERGE PATINFO_ NRS9899 CGI_;
        BY SUBJID;

    RUN;

    DATA TXT_;
        LENGTH TEXT $40;
        SET PATINFO_;

        %*initializes annotation variables;
        %DCLANNO;

        %*set the coordinate system to percentages of graphics area for *;
        %*x & y and to data values for h;
        %SYSTEM(3,3,2);

        %*display demographic information;
        %LABEL(0,98,"STUDY ID: "||LEFT(STUDYID1),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,92,"SUBJECT: "||LEFT(SUBJID),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,86,"TREATMENT: "||LEFT(TRTGRP),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,80,"SEX: "||LEFT(SEX),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,74,"RACE: "||LEFT(RACE),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,68,"AGE: "||LEFT(PUT(AGE,3.)||" YEARS"),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,62,"WEIGHT: "||LEFT(PUT(WEIGHTKG,3.)||" KG"),BLACK,0,0,4,SIMPLEX,6);
        %LABEL(0,56,"COUNTRY: "||LEFT(COUNTRY),BLACK,0,0,4,SIMPLEX,6);

```

PhUSE 2013

Paper PP05

```

%LABEL(0,50,"AVG PAIN INTENSITY(LOCF)W12:
"||LEFT(PUT(WIN98,5.2)),BLACK,0,0,4,SIMPLEX,6);
%LABEL(0,44,"AVG PAIN INTENSITY(LOCF)OVERALL:
"||LEFT(PUT(WIN99,5.2)),BLACK,0,0,4,SIMPLEX,6);
%LABEL(0,38,"PGIC at WEEK 4: "||LEFT(CGCHG8),BLACK,0,0,4,SIMPLEX,6);
%LABEL(0,32,"PGIC at WEEK 8: "||LEFT(CGCHG10),BLACK,0,0,4,SIMPLEX,6);
%LABEL(0,26,"PGIC at WEEK 12: "||LEFT(CGCHG12),BLACK,0,0,4,SIMPLEX,6);
%LABEL(0,20,"PGIC at ENDPOINT: "||LEFT(CGCHG),BLACK,0,0,4,SIMPLEX,6);

DROP STUDYID SUBJID TRTGRP WEIGHTKG AGE AGEUNIT SEX RACE COUNTRY;

RUN;

%*set the xy position where the plotting would start;
GOPTIONS HORIGIN=&XIN. IN VORIGIN=&YIN. IN;

%*plot the graph;
PROC GANNO ANNOTATE=TXT_ DATASYS;
RUN;
QUIT;

%MEND INFOBOX;

*****;
*      The main plots are drawn here. Two types of plots are supported: i) Y vs. X      *;
*      ii) Y1 & Y2 vs. X. The plotting starts at the specified XY origin.              *;
*                                                                                       *;
*      Limitation:      1. For Y vs. X plot, the axes could be different, but for Y1/Y2  *;
*                        vs. X, the Y2 axis is fixed.                                  *;
*                                                                                       *;
*      indat = input dataset with variables to be plotted                             *;
*      yvar = Y-axis variable                                                           *;
*      yvar2 = Right-side Y-axis variable                                              *;
*      xvar = X-axis variable                                                           *;
*      hax   = horizontal axis scheme by axis statement                              *;
*      vax   = vertical axis scheme by axis statement                                *;
*      condn = dataset subset condition#1                                              *;
*                        Right now, it is designed to be subsetting for a subject      *;
*      categ = dataset subset condition#2                                              *;
*      h = horizontal origin in inches from bottom-left                               *;
*      v = vertical origin in inches from left-bottom                                 *;
*      usymb = user-defined symbols definition                                        *;
*      name = vertical axis label                                                       *;
*****;
%MACRO YXPLOT (INDAT=, YVAR=, YVAR2=, XVAR=, VAX=AXIS2, CONDN=1,
CATEG=1, H=, V=, NAME=, USYMB='NO');

%*default x-axis;
AXIS1 COLOR=BLACK STYLE=1 OFFSET=(0,0) ORDER=(&XMING. TO &XMAXG. BY &XMAJG.)
MAJOR=NONE MINOR=NONE VALUE=NONE LABEL=NONE;

%*default y-axis;
AXIS2 COLOR=BLACK STYLE=1 ORDER=(1 TO 4 BY 1)
MINOR=NONE LABEL=(A=90 H=0.5 "&NAME.") VALUE=(h=0.3);

%*depending upon the treatment group select the right-side y-axis;
DATA _NULL_;
SET &INDAT (OBS=1); *NEED THE FIRST RECORD FOR IDENTIFYING THE TREATMENT GORUP;
WHERE &CONDN AND &CATEG;
BY SUBJID;

IF UPCASE(TRTGRP) EQ "OXYCODONE CR" THEN
CALL EXECUTE("AXIS22 COLOR=BLACK STYLE=1 ORDER=(0 TO 160 BY 40) VALUE=(h=0.5)
MINOR=NONE LABEL=(A=90 H=.5 'TOTAL DAILY DOSE ('||LEFT(TRIM(TRTGRP))||")'
);");
ELSE
CALL EXECUTE("AXIS22 COLOR=BLACK STYLE=1 ORDER=(0 TO 800 BY 100) VALUE=(h=0.5)
MINOR=NONE LABEL=(A=90 H=.5 'TOTAL DAILY DOSE ('||LEFT(TRIM(TRTGRP))||")'
);");

RUN;

%IF %LEFT(%TRIM(&USYMB)) EQ 'NO' %THEN %DO;

```

PhUSE 2013

Paper PP05

```

    %*JOIN THE DATA POINTS BY A LINE, DIFFERENT SYMBOLS AND COLORS IF TWO PLOTS;
    SYMBOL1 I=JOIN CI=BLUE VALUE=CIRCLE H=0.3;
    SYMBOL2 I=JOIN CI=RED VALUE=TRIANGLE LINE=8 H=1;

%END;

%*draw lines parallel to y-axis to indicate different timepoints in the trial;
%YLINE;

%*select a position for the plot;
GOPTIONS HORIGIN=&H. IN VORIGIN=&V. IN;

%*plot the graph;
PROC GPLOT DATA=&INDAT ;
    WHERE &CONDN AND &CATEG;
    %*plot with one y-axis or two y-axis depending upon the request;
    %IF NOT %LENGTH(&YVAR2) %THEN %DO;
        PLOT &YVAR * &XVAR / HAXIS=AXIS1 VAXIS=&VAX ANNO=LINE_;
    %END;
    %ELSE %IF %LENGTH(&YVAR2) > 0 %THEN %DO;
        %*legend included only if there are two different plots in one graph;
        LEGEND1 POSITION=(BOTTOM LEFT INSIDE) MODE=SHARE NOFRAME ACROSS=2
        LABEL=NONE VALUE=(H=.5);
        LEGEND2 POSITION=(BOTTOM RIGHT INSIDE) MODE=SHARE NOFRAME ACROSS=2
        LABEL=NONE VALUE=(H=.5);

        PLOT &YVAR * &XVAR / HAXIS=AXIS1 VAXIS=&VAX LEGEND=LEGEND1 ANNO=LINE_;
        PLOT2 &YVAR2 * &XVAR / VAXIS=AXIS22 LEGEND=LEGEND2;;
    %END;

RUN;
QUIT;

GOPTIONS RESET=(AXIS, LEGEND, PATTERN, SYMBOL, TITLE, FOOTNOTE) ROTATE=NOROTATE;

%MEND YXPLOT;

*****;
*      A macro defined for calling yxplot and xaxis for plotting vital parameters *;
*      Depending upon the number of parameters to be plotted, number of calls are *;
*      made for yxplot macro. Each macro call plots a set of graphs on a single *;
*      page for one subject. *;
*      *;
*      subj=subject number *;
*****;
%MACRO PLOT_VITALS (SUBJ=);

    %*until absolutely neede plot all graphs on the same page;
    ODS PDF STARTPAGE=NO;
    %*workaround. the graphs overflow to next page otherwise;
    ODS PDF TEXT=" ";

    %*area of the first sentence that is to be the title1;
    GOPTIONS HSIZE=6IN VSIZE=.15IN;
    %LET TTL1=%STR(SUBJECT SAFETY, EXPOSURE AND EFFICACY PROFILES (3));
    %*draw the title1;
    %TITLETEXT(TXT=&TTL1);

    %*area the second sentence that is to be the title2;
    GOPTIONS HSIZE=6IN VSIZE=.1IN;
    %LET TTL1=%STR(VITAL STATISTICS);
    %*draw the title;
    %TITLETEXT(TXT=&TTL1, YIN=6.1);

    %*area of the infobox with demographic information;
    GOPTIONS HSIZE=2IN VSIZE=2.4IN;
    %*put to demographic information for a subject;
    %INFOBOX(WHERE=SUBJID EQ "&SUBJ.");

    %IF &VITALPN. EQ 1 %THEN %DO;

DATA NRSSUM98 (RENAME=(CHANGE=AVGPI_W12 WINDOW=WINDOW12)) NRSSUM99 (RENAME=(CHANGE=AVGPI_OV
WINDOW=WINDOWOV));
```

PhUSE 2013

Paper PP05

```
SET KNRSSUM_ (KEEP=SUBJID CHANGE PARAMC WINDOW WINDOWC);
IF WINDOWC=98 THEN OUTPUT NRSSUM98;
ELSE IF WINDOWC=99 THEN OUTPUT NRSSUM99;
KEEP SUBJID CHANGE WINDOW ;

RUN;

DATA VITALXT1;
MERGE VITAL_W(IN=A) KDEMOG(IN=B) KCGI_(IN=C KEEP=SUBJID CGCHG)
      NRSSUM98(IN=D) NRSSUM99(IN=E);
BY SUBJID;
IF A;
RENAME CGCHG=PGIC ;

RUN;

%END;

AXIS5 COLOR=BLACK STYLE=1 ORDER=(20 TO 120 BY 40) MINOR=NONE VALUE=(h=0.35) LABEL=(A=90
H=0.5 "PR(BPM)");
SYMBOL1 CI=GREEN VALUE=CIRCLE H=1;
%*yx plot area. this area remains set for the next graph and further as well unless
changed;
GOPTIONS HSIZE=6.25IN VSIZE=0.8IN;
%YXPLOT( INDAT=VITAL_W, YVAR=VALUE, XVAR=VSRELDY, CONDN=SUBJID EQ
"&SUBJ.", VAX=AXIS5, CATEG=%STR(PARAM EQ 'PULSE RATE (bpm)'), H=2.95, V=3.1, NAME=%STR(PR),
USYMB='YES' );

AXIS6 COLOR=BLACK STYLE=1 ORDER=(50 TO 250 BY 50) MINOR=NONE VALUE=(h=0.35) LABEL=(A=90
H=0.5 "SBP(MMHG)");
SYMBOL1 CI=RED VALUE=STAR H=1;
%YXPLOT( INDAT=VITAL_W, YVAR=VALUE, XVAR=VSRELDY, CONDN=SUBJID EQ "&SUBJ.", VAX=AXIS6,
CATEG=%STR(PARAM EQ 'SYSTOLIC BLOOD PRESSURE (mmHg)'), H=2.95, V=3.92, NAME=%STR(SBP),
USYMB='YES' );

AXIS6 COLOR=BLACK STYLE=1 ORDER=(20 TO 140 BY 40) MINOR=NONE VALUE=(h=0.35) LABEL=(A=90
H=0.5 "DBP(MMHG)");
SYMBOL1 CI=BLUE VALUE=TRIANGLE H=1;
%YXPLOT( INDAT=VITAL_W, YVAR=VALUE, XVAR=VSRELDY, CONDN=SUBJID EQ "&SUBJ.", VAX=AXIS6,
CATEG=%STR(PARAM EQ 'DIASTOLIC BLOOD PRESSURE (mmHg)'), H=2.95, V=4.74, NAME=%STR(DBP),
USYMB='YES' );

AXIS8 COLOR=BLACK STYLE=1 ORDER=(5 TO 45 BY 15) MINOR=NONE VALUE=(h=0.35) LABEL=(A=90
H=0.5 "RR(/MIN)");
SYMBOL1 CI=CYAN VALUE=CIRCLE H=1;
GOPTIONS HSIZE=6.23IN VSIZE=0.8IN;
%YXPLOT( INDAT=VITAL_W, YVAR=VALUE, XVAR=VSRELDY, CONDN=SUBJID EQ "&SUBJ.", VAX=AXIS8,
CATEG=%STR(PARAM EQ 'RESPIRATION RATE (/min)'), H=2.97, V=5.56, NAME=%STR(RR),
USYMB='YES' );

%*reset the graphics area again for the x-axis;
GOPTIONS HSIZE=7.17IN VSIZE=.5IN;
%*XAXIS(H=2.73, V=2.5);

%IF &VITALPN. EQ 1 %THEN %DO;

DATA VITAL_OXY VITAL_TAP VITAL_PLA;
SET VITALXT1;
WHERE VALUE NE . AND VSRELDY NE .
      AND PARAMC IN (1,2,3,4);
IF TRTGRP EQ 'OXYCODONE CR' THEN OUTPUT VITAL_OXY;
ELSE IF TRTGRP EQ 'TAPENTADOL ER' THEN OUTPUT VITAL_TAP;
ELSE IF TRTGRP EQ 'PLACEBO' THEN OUTPUT VITAL_PLA;

KEEP SUBJID TRTGRP STUDYID WEIGHTKG AGE AGEUNIT SEX RACE COUNTRY
PARAM VALUE VSRELDY PGIC WINDOW12 AVGPI_W12 WINDOWOV AVGPI_OV;

RUN;

%LET VITALPN=0;

%END;

%*after a set of graphs have been plotted on a page, move to another page;
ODS PDF STARTPAGE=NOW;
```

PhUSE 2013

Paper PP05

```
%MEND PLOT_VITALS;

*define the x-axis;
%LET XMING=-30;
%LET XMAXG=130;
%LET XMAJG=10;
%LET XLAB=DAYS;

*set the PDF page orientation. The graphs with current settings won't display;
*correctly if presented on a potrait orientation;
*reset the graphics options;

OPTIONS ORIENTATION=LANDSCAPE LEFTMARGIN=0.5in
RIGHTMARGIN=0.1in
TOPMARGIN=0.1in
BOTTOMMARGIN=0.1in
;
GOPTIONS DEVICE=PDFC FTEXT="SIMPLEX" HTEXT=8PT NOPROMPT ;

*data _null_ has been used to execute the macro for each subject;
DATA _NULL_;
    SET UNIQSUBJ END=EOF;
    RETAIN TAG 0;
    IF TAG EQ 0 THEN DO;
        *set the output pdf files into multiple parts;
        CALL EXECUTE("ODS PDF FILE='PLT\ppfplt02_"||TRIM(LEFT(_N_))||"_.PDF'
            BOOKMARKLIST=HIDE; " ) ;
        TAG=1;
    END;
    CALL EXECUTE("ODS PROCLABEL='SUBJECT: " || LEFT(TRIM(SUBJID)) || "';");
    CALL EXECUTE('%PLOT_NRSEXP(SUBJ=' || LEFT(TRIM(SUBJID)) || ')');
    CALL EXECUTE('%PLOT_VITALS(SUBJ=' || LEFT(TRIM(SUBJID)) || ')');

    *process 25 subjects and then create the pdf for access;
    *change the value if each pdf should contain more than 25 subjects;
    IF (MOD(_N_,25) EQ 0) OR EOF THEN DO;
        CALL EXECUTE("ODS PDF CLOSE;");
        TAG=0;
    END;

RUN;
```