

Creating a Customized Graph for Adverse Event Incidence and Duration

Sanjiv Ramalingam, Octagon Research Solutions Inc.

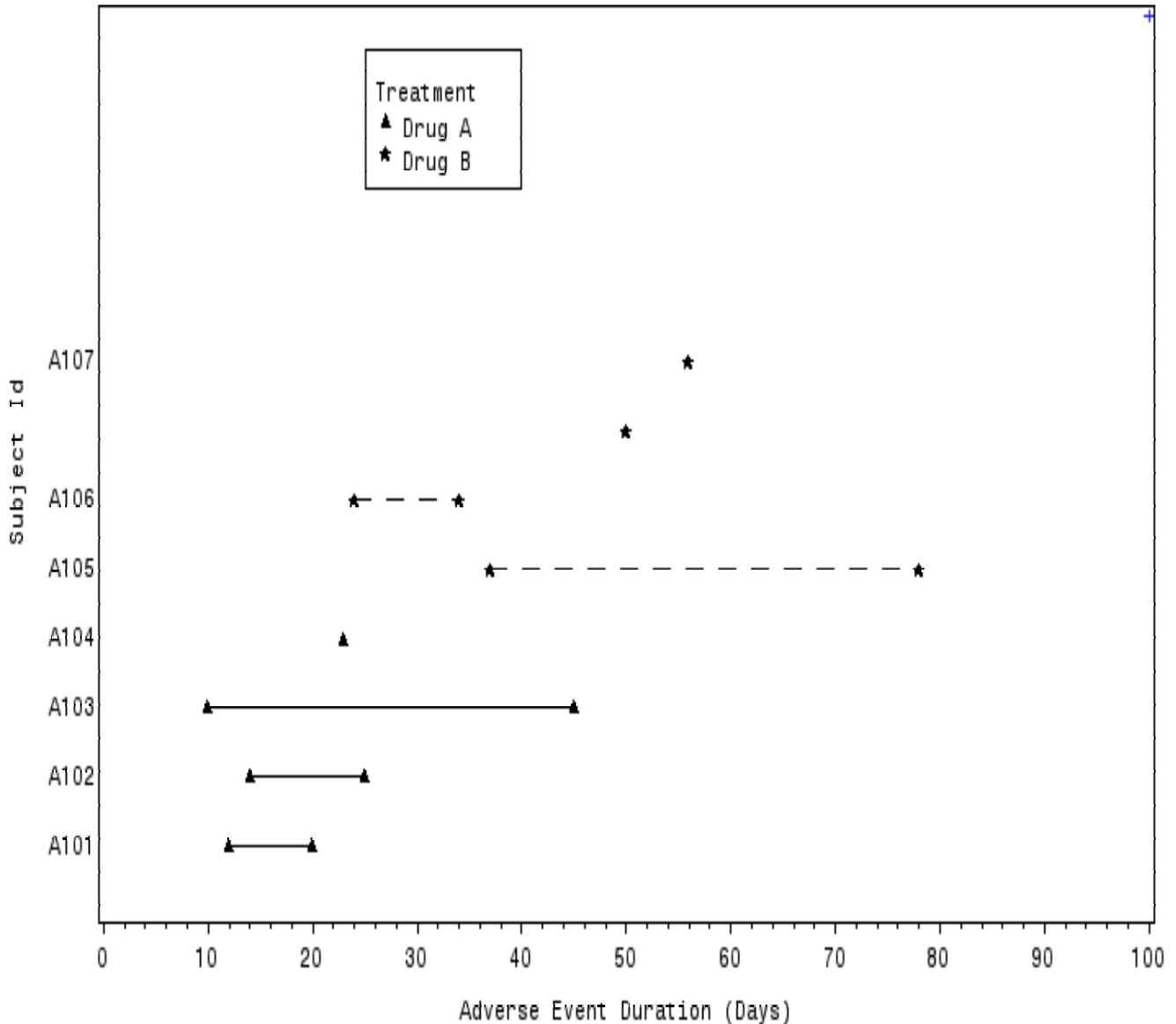
Abstract

Adverse event (AE) analysis is a critical part of clinical trial data analysis, and is commonly handled through conventional summary tables and listings. However, presenting adverse event information graphically at the individual subject level may help the clinician visualize the assessment of safety, emphasize the adverse events of special interest, and compare the adverse events between treatments and subjects. A very robust technique to create such a graph using the annotate feature is discussed. The technique also involves dynamically creating SAS® statements using PROC SQL and SAS Macro.

Introduction

The graph illustrates all the subjects that had a particular adverse event. Subjects identifiers are listed on the vertical axis with days on study drug in the horizontal axis. The distance between two connected symbols determines the duration of the AE. Dashed lines indicate that the AE was not treatment emergent.

The graph of interest is illustrated below:



Core Technique

One of the techniques incorporated involves the use of dynamically created formats for the subject identifiers illustrated on the vertical axis. The implementation is done through the use of PROC SQL and SAS macro. It is assumed that the reader has fundamental knowledge of the annotate feature, like the different co-ordinate systems and the know how to draw simple lines using the annotate feature.

Two major parts are involved. The first part involves two steps. In the first step, a text/string of the required code is created. This can be implemented using the concatenation operator.

E.g. To create a statement X='First Label'

```
fmtl=strip(start)||strip("=")||compress("'"||strip(label)||"'");
```

In the second step the text string (code) is converted to a macro variable using PROC SQL. Macro variables can be created using PROC SQL in two different ways depending on the requirement. If the requirement is to create a macro variable for each record of the variable of interest the following code structure will be used:

```
proc sql noprint;
  select varint                                /* Variable of interest */
  into :var1 -:var100                          /* Macro variables to be created */
  from srcds                                   /* Source dataset */
  ;
quit;
```

If the requirement is to make all the records of the variable of interest into one macro variable, the following code structure will be used:

```
proc sql;
  select varint                                /* Variable of interest */
  into :mvar separated by ' '                 /* Macro variable to be created */
  from srcds;                                 /* Source dataset */
quit;
```

The second part involves invoking the macro variable. This is implemented using the ampersand operator where required in the program. E.g. :

```
proc format ;
  &mvar;
run;
```

Further applications of use of dynamically generating SAS statements has been discussed by the author previously¹.

Implementation

```
/* create dummy data */  a1
data final;
  input usubjid $ pt $ s_dur e_dur trt $ teaeftl $ ;
  datalines;
  A101 backpain 12 20 A N
  A102 backpain 14 25 A N
  A103 backpain 10 45 A N
  A104 backpain 23 . A N
  A105 backpain 37 78 B Y
  A106 backpain 24 34 B Y
  A107 backpain 50 . B N
  A107 backpain 56 . B N
  ;
run;

/* Formation of template */  a2
data templ;
  retain k t l u;
  set final end=eof;
  if missing(e_dur) then e_dur=0;
  if _n_ eq 1 then do;
    t=e_dur;
    u=s_dur;
  end;
  k=max(e_dur,t);
  l=min(s_dur,u);
  t=k;
  u=l;
  if eof then do;
    maxh=k+20;k_str=put(maxh,best.); lnth=length(k_str);
    t1=input(substr(k_str,lnth,1),best.);
    t2=10-t1;maxh=maxh+t2;
    call symput('ntot',put(_n_+5,best12.));
    call symput('maxhx',put(maxh,best.));
    y=_n_+5;
    x=maxh;
    output;
  end;
  keep x y;
run;

/* actual plotting of data using annotate */  a3
data annol;
  length function $8 text $100 style $20;
  set final;
  if teaeftl eq 'N' then line=1;
  else if teaeftl eq "Y" then line=20; }  a4
  if _n_ eq 1 then i=1;
  else if _n_ ne 1 then i+1;
  retain line 1 xsys ysys '2' line 1 color 'black' when 'a';
  if not missing(e_dur) then do;
    function="symbol";
    if trt eq "A" then do;
      style="marker"; text="C";size=0.75;
    end;
    else do;
      style="marker"; text="V";size=1;
    end;
  end;
  x=s_dur; y=i;
  output;
  function='move'; x=s_dur;y=i;
  output;function='draw';x=e_dur;y=i;
  output;
```

```

        function="symbol";
        if trt eq "A" then do;
            style="marker"; text="C";size=0.75;
        end;
        else do;
            style="marker"; text="V";size=1;
        end;
        x=e_dur; y=i;
        output;
    end;
    if missing(e_dur) then do;
        function="symbol";
        if trt eq "A" then do;
            style="marker"; text="C";size=0.75;
        end;
        else do;
            style="marker"; text="V";size=1;
        end;
        x=s_dur; y=i;
        output;
    end;
end;
run;

/* macro variables for proc format */
proc sort data=final;
    by usubjid;
run;

/* macro variable for proc format ;value */
data temp1;
    length labl $100;
    set final end=eof;
    keep usubjid labl;
    labl=put(_n_,best12.)||'='||' " '||usubjid||' " ' ; → a5
    labl=compress(labl,' ');
    output;
    if eof then do;
        labl="other=[otherid.]";
        usubjid="zyx";
        output;
    end;
end;
run;

/* macro variable for proc format ;picture */ → a6
data temp2;
    length lablo $35;
    set temp1 end=eof;
    by usubjid;
    if first.usubjid and not last.usubjid and not eof then do;
        lablo=strip(put(_n_,best12.)||'='||' " '||' " '||' " ');
        labl=" ";
    end;
    if not first.usubjid and not last.usubjid and not eof then do;
        lablo=strip(put(_n_,best12.)||'='||' " '||' " '||' " ');
        labl=" ";
    end;
    else if first.usubjid and not last.usubjid and eof then
        lablo=strip(put(_n_,best12.)||'='||' " '||' " ');
    if usubjid eq "zyx" then do;
        lablo=strip(put(_n_,best12.)||'='||' " '||' " ');
    end;
end;

```

```

output;
/*insert dummy ;for proc format */ → a7
if _n_ eq 1 then do;
    lablo='0='||' ' " "';usubjid= " ";labl=" ";
    output;
    lablo=strip(put (&ntot-3,best12.))||'='||' ' " "';usubjid= " ";labl=" ";
    output;
    lablo=strip(put (&ntot-2,best12.))||'='||' ' " "';usubjid= " ";labl=" ";
    output;
    lablo=strip(put (&ntot-1,best12.))||'='||' ' " "';usubjid= " ";labl=" ";
    output;
    lablo=strip(put (&ntot,best12.))||'='||' ' " "';usubjid= " ";labl=" ";
    output;
end;
run;

/* macro variable for variable ;format value */
proc sql;
    select labl
    into :labl_m separated by ' '
    from temp2;
quit;

/* macro variable for variable;format picture*/
proc sql;
    select lablo
    into :lablo_m separated by ' '
    from temp2;
quit; → a8

proc format ;
    value ylabel
    (default=20)
    &labl_m
    ;
    picture otherid
    (default=20)
    &lablo_m
    ;
run;

data anno2; → a9
    length function $8 text $100 style $20;
    retain line 1 xsys ysys '2' line 1 color 'black' when 'a' style ' ' text ' ';
    function='label'; text="Treatment"; x=31;y=&ntot-1;color='black';size=1; output;
    function="symbol";style="marker";text="C";size=0.75 ;color="black"; x=27;
    y=&ntot-1.5;output;
    function='label';text="Drug A"; style=" ";x=32;y=&ntot-1.5; color='black';size=1.5;
    output;
    function="symbol";style="marker"; text="V";size=1 ;color="black"; x=27;
    y=&ntot-2;
    output;
    function='label'; text="Drug B"; style=" "; x=32;
    y=&ntot-2;color='black';size=1.5;
    output;
    /* draw frame around legend */ → a10
    function='move'; x=25; y=&ntot-2.5; style=' ' ; text=' ' ;output;
    function='draw'; x=40; y=&ntot-2.5; line=1;style=' ' ; text=' ' ;output;
    function='draw'; x=40; y=&ntot-0.5; line=1;style=' ' ; text=' ' ;output;
    function='draw'; x=25; y=&ntot-0.5; line=1;style=' ' ; text=' ' ;output;
    function='draw'; x=25; y=&ntot-2.5; line=1;style=' ' ; text=' ' ;output;
run;

```

```

data annof;
    set anno1 anno2;
run;

/* Figure setup */
title;
footnote;
options nodate nonumber orientation=landscape;
GOPTIONS reset=all device=png target=png xmax=7.5in ymax=4.5in xpixels=1500 ypixels=900
FTEXT="Courier" FTITLE="Courier/bold" FBy="Courier" ;
ods rtf file="C:\Documents and Settings\sramalingam\Desktop\test.rtf"
style=TStyleRTF nogtitle nogfootnote;

/* combine annotate and plot */
proc gplot data=templ;
    axis1 label=(angle=90 h=1.5 "Subject Id")
        order=(0 to &ntot by 1) value=( h=1.5) minor=none major=none width=1;
    axis2 split="," label=(h=1.5 'Days on Study Drug') order=(0 to &maxhx by 10)
        value=( h=1.5) offset=(1,1) ;
    plot y*x/anno=annof
        haxis=axis2
        vaxis=axis1
        ;
    format y ylabel.;
run;
quit;

ods rtf close ;

```

A simple dataset (FINAL) has been created to illustrate the technique [a1]. The variables of interest are the subject identifiers (USBJID), preferred term (PT), start duration (S_DUR), end duration (E_DUR), treatment (TRT) and the treatment emergent adverse event flag (TEAEFL). A template (TEMPL) that defines the structure of the graph (X co-ordinate and Y co-ordinate) is created [a2]. The X co-ordinate is assigned the maximum value of the end duration (E_DUR) plus a constant (20). A macro variable, MAXHX, of the same value is created. The constant (20) is added to allow for sufficient space on the X-axis. The Y co-ordinate is assigned to the total number of observations (macro variable, NTOT) plus a constant (e.g. 5 to allow for sufficient space to insert the legend).

An annotate dataset (ANNO1) is created to draw the symbols along with duration. Different symbols are then used to define each of the two treatments, A and B [a3]. The `_N_` variable is used as the identifier for each of the subject records and serves as the Y axis value. Since only the data area is used, the co-ordinate system used is XSYS=2 and YSYS=2². A dashed line is used to represent adverse events that were treatment emergent [a4]. Different line styles to be used are available in the SAS online documentation³.

Now that the data values have all been defined, the next step is to create formats for the subject identifiers as `_N_` was previously used as the identifier. Two formats are used. One is the regular, 'value' format that associates each of the identifier values with that of the subject identifiers [a5]. The formats are dynamically generated using the core technique described previously. A dummy additional record is created so that the last 'value' format statement assigns values not defined previously to the other format being used.

The other format used is the 'picture' format [a6]. This format ensures that for subjects having multiple entries of the same preferred term (e.g. subject A107), only the latest subject record is displayed on the graph and 'A107' is not displayed on the vertical axis twice. This format also ensures that the dummy records (as in the 5 additional records that were created to insert the legend) will be assigned a blank [a7]. The macro variables [a8] are created and the logic follows the same core technique discussed previously.

Another annotate dataset (ANNO2) is created to create the legend [a9] and draw the frame around the legend [a10]. The two annotate datasets are then set together to create the final annotate dataset, ANNOF [a11]. The figure is then plotted using the standard GPLOT procedure [a12].

Conclusion

A technique to plot a graph for adverse event incidence and duration has been shown. Since only so many adverse events can be listed on a single page, the source dataset can be subset to display the graph on multiple pages if the need arises.

References

1. Ramalingam, Sanjiv. Automating the Pooling of Variables Across Multiple Datasets Using Proc SQL and SAS Macro. PharmaSUG 2009, Portland, OR.
2. Carpenter, Art. 1999. *Annotate: Simply the Basics*. Cary, NC: SAS Institute Inc.
3. SAS online documentation for version 9. SAS Institute Inc. Cary, NC. [Cited; Available from: <http://support.sas.com/rnd/base/topics/statgraph/v91StatGraphStyles.htm>]

Acknowledgements

I would like to thank Allan Glaser, my manager for always reviewing my papers and offering constructive feedback. My acknowledgement would be incomplete without thanking Art Carpenter for his wonderful book on annotate that helped me get started and use this very powerful feature.

Contact Information

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

Sanjiv Ramalingam
Octagon Research Solutions Inc.
585 E Swedesforde Road
Wayne, PA 19087
Email : sramalingam@octagonresearch.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.