PhUSE 2008

Paper PO13

Run Forest! Run! Create a flexible forest plot with SAS® Enterprise Guide

Emmanuel Bourmaud, Keyrus Biopharma, Paris, France
Adrien Vallee, Keyrus Biopharma, Paris, France
Gwenael Le Teuff, Keyrus Biopharma, Paris, France

ABSTRACT

To our knowledge, no SAS application dedicated to meta-analysis exists. Only home-made SAS macros have been developed to perform the different steps of the meta-analysis, separately. This paper presents the first part of a project to develop an application for meta-analysis. It consists of the creation of the classic representation for meta-analysis results i.e. the forest plot. We explain how to create a fully customized stored process displaying the plots and any further information defined by the users with SAS Enterprise Guide.

INTRODUCTION

The number of meta-analyses is increasing in medical sciences. The principle of a meta-analysis is to combine the results of independent but related studies. Meta-analysis consists of several steps including the selection of studies, the evaluation of the quality of the studies, the detection of publication bias, the choice of the effect size, the method of calculating the common effect size and the presentation of the results. Although software dedicated to meta-analysis exists [1, 2], the SAS-users can only use SAS macros to perform these steps, separately. For example, when creating a forest plot to display the results of a meta-analysis, Mitchell [3] and Foster [4] proposed a customized SAS macro using the GPLOT procedure. However, a review of the literature shows that both the information and the order in which the information is displayed in the forest plot can be different. In this context, we propose a SAS program for generating a flexible forest plot using SAS Enterprise Guide. This point allows the user to specify which information is to be displayed and in which order.

Firstly, we describe the structure of the input data. Secondly, we present the project with SAS Enterprise Guide and describe the macros. Thirdly, we present an application based on a simple meta-analysis.

INPUT DATA SOURCE

The input data source contains one observation for each study included in the meta-analysis. To illustrate, we have taken the first five observations of Table 2.1 and Table 2.2 from Sutton's example [2]. These tables represent the total mortality from random clinical trials investigating the effect of cholesterol lowering interventions. We have completed the data by calculating the information for the 'combined' study. ID represents the unique identification, STUDY the label of the study, NBPAT_GRPA the number of patients in the treatment arm, NBPAT_GRPB the number of patients in the control arm, TOTAL_DEATH the total number of death, OR odds ratio, OR_LOW and OR_HIGH the lower and upper limits of the 95% confidence interval. This dataset must be added to by the user. For example, a column containing the weight associated to each study can be added. The SAS code is provided below:

```
DATA FOREST_PLOT;
      INPUT  ID STUDY $ NBPAT_GRPA NBPAT_GRPB  TOTAL_DEATH   OR     OR_LOW OR_HIGH;
      CARDS;
             1  STUDY1   204        202         79           0.48   0.29   0.79
             2  STUDY2   285        147         108          0.93   0.59   1.47
             3  STUDY3   156        119         77           0.62   0.36   1.04
             4  STUDY4   88         30          5            0.23   0.04   1.22
             5  STUDY5   30         33          3            0.14   0.01   2.89
             6  COMBINED 763        531         272          0.64   0.49   0.84
RUN;
```

PROJECT WITH SAS ENTERPRISE GUIDE

We decide to use SAS Enterprise Guide because it allows the user to build a project step-by-step, to enter the values of the parameters via an interface and to create a stored process.
The first step of the project is the initialization of some parameters. The second step describes three SAS macros and the third step, allows the creation of a store process.

Step 1

In this step, we present the position parameters used to manage a flexible forest plot and the input parameters used to control the information the user wants to display.

Position parameters

To create a flexible forest plot, we suggest that the total area of the output is divided into three windows containing (i) the 'study identification', (ii) the 'statistics' and (iii) the 'graph'. Each of these windows is managed by a set of parameters whose signification is described in the following annotate forest plot (Figure 1). This figure is created using the dataset from the 'Input data source' paragraph.

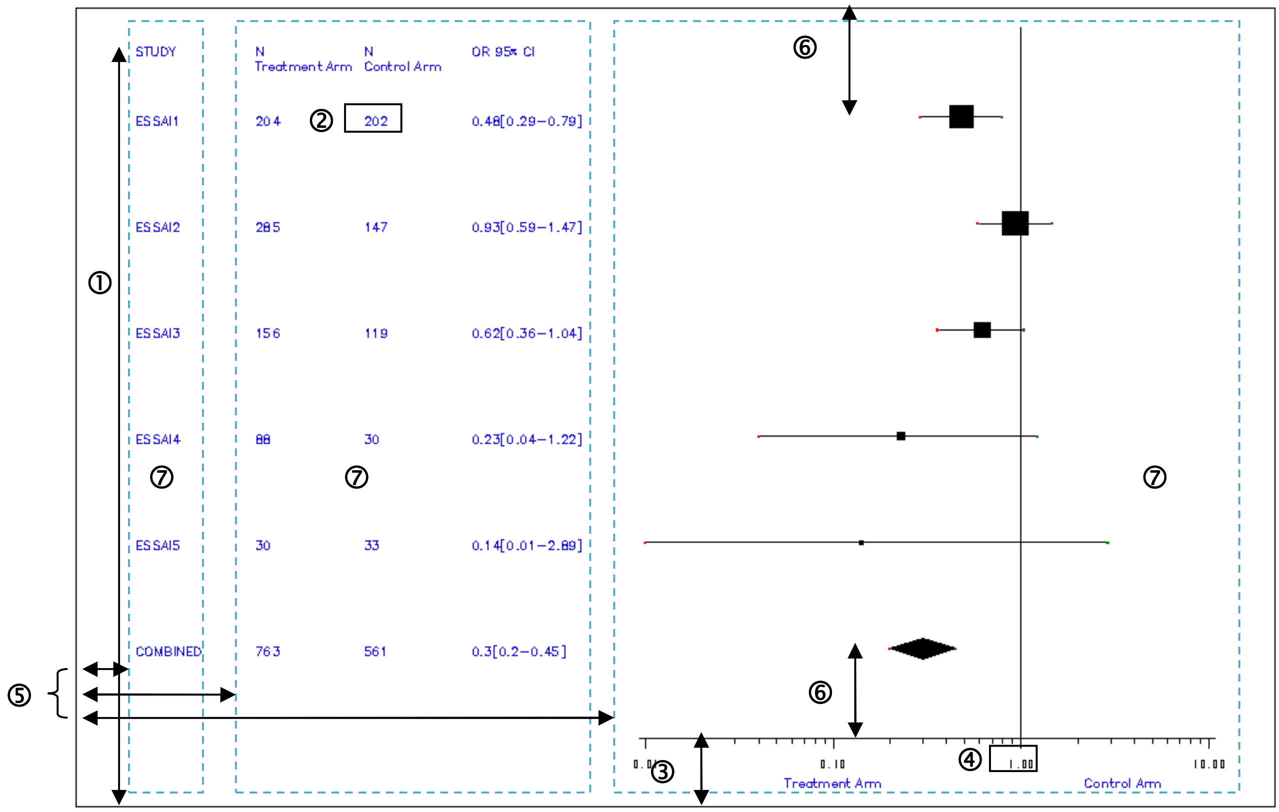


Figure 1: Position parameters used to perform a flexible forest plot

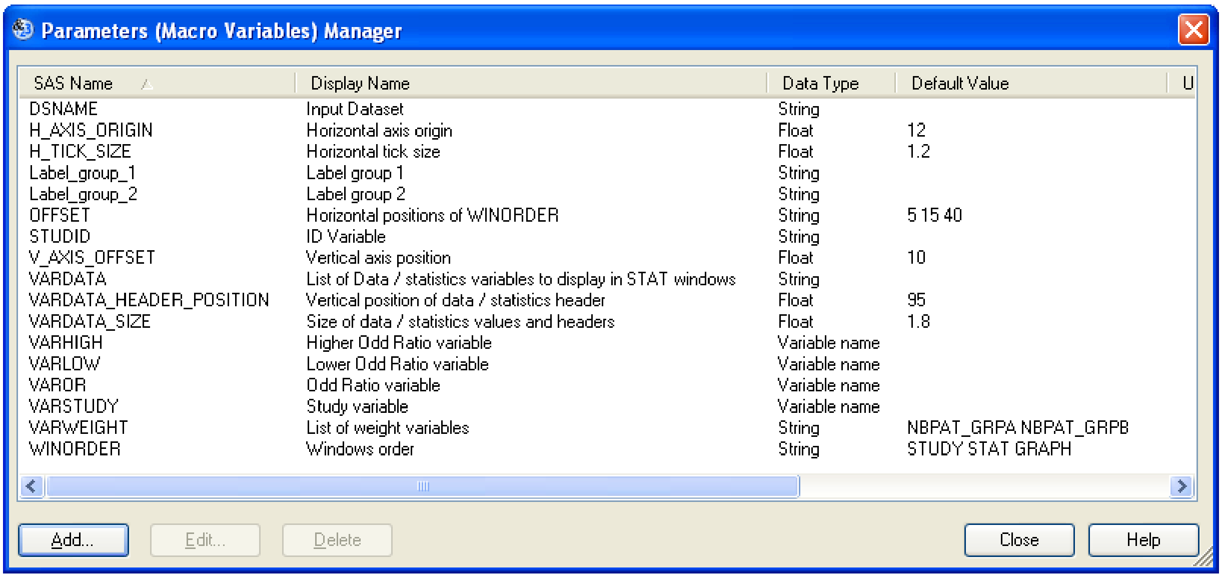
Keys:

① **Vardata_header_position** *(default=95)*: vertical position in percentage of the headers for the 'study identification' and 'statistics' windows
② **Vardata_size** *(default= 1.6)*: size in percentage of the headers, values and label groups
③ **H_axis_origin** *(default= 12)*: bottom margin
④ **H_tick_size** *(default= 1.2)*: size in percentage of the tick marks on the horizontal axis
⑤ **Offset** *(default=5 15 40)*: horizontal position in percentage for each window
⑥ **V_axis_offset** *(default= 10)*: specifies the distance in percentage from the first and last major tick marks to the ends of the vertical axis line
⑦ **Winorder** *(=STUDY STAT GRAPH)*: defines which window to display and the order. The possible values are STUDY for 'study identification', STAT for 'statistics' and GRAPH for 'graph'. These values are separated by a space.

Input parameters
These parameters contain the names of the input dataset and the names of the variables.

- **Dsname***(default=FOREST_PLOT)*: input dataset name
- **Studid***(default=ID)*: identification variable
- **Varstudy***(default=STUDY)*: study variable whose the values are displayed in 'study identification' window
- **Varor***(default=OR)*, **Varlow***(default=OR_LOW)*, **Varhigh***(default=OR_HIGH)*: variables represent odds ratio, lower and upper limits of the 95% confidence interval
- **Varweight***(default=NBPATGRPA NBPATGRPB)*: weight variable determines the size of squares. Here, the weight is based on the number of patients in the 2 groups.
- **Vardata***(default=NBPATGRPA NBPATGRPB OR_CONCAT)*: list of variables to display in 'statistics' window. OR_CONCAT is a new variable. It represents the concatenation of OR, OR_LOW and OR_HIGH variables.

With SAS Enterprise Guide, we set both position and input parameters by clicking on the 'tools' menu and 'parameters manager'. The following screen is then obtained (Figure 2).


Parameters (Macro Variables) Manager

SAS Name	Display Name	Data Type	Default Value	U
DSNAME	Input Dataset	String		
H_AXIS_ORIGIN	Horizontal axis origin	Float	12	
H_TICK_SIZE	Horizontal tick size	Float	1.2	
Label_group_1	Label group 1	String		
Label_group_2	Label group 2	String		
OFFSET	Horizontal positions of WINORDER	String	5 15 40	
STUDID	ID Variable	String		
V_AXIS_OFFSET	Vertical axis position	Float	10	
VARDATA	List of Data / statistics variables to display in STAT windows	String		
VARDATA_HEADER_POSITION	Vertical position of data / statistics header	Float	95	
VARDATA_SIZE	Size of data / statistics values and headers	Float	1.8	
VARHIGH	Higher Odd Ratio variable	Variable name		
VARLOW	Lower Odd Ratio variable	Variable name		
VAROR	Odd Ratio variable	Variable name		
VARSTUDY	Study variable	Variable name		
VARWEIGHT	List of weight variables	String	NBPAT_GRPA NBPAT_GRPB	
WINORDER	Windows order	String	STUDY STAT GRAPH	

Add... Edit... Delete Close Help

Figure 2: Parameters screen (position and input) in SAS Enterprise Guide

Step 2
In the second step, a project is created with SAS Enterprise Guide (Figure 3). This project includes three main SAS macros. The first macro INIT_FOREST_PLOTS initializes the macro variables for graphical and output management. The second macro ANNOTATE_FOREST_PLOTS defines, based on the previous step, the annotate dataset in order to specify what to draw or write, and where. The third macro DRAW_FOREST_PLOTS defines the axis specifications, draw plots and uses the annotate dataset in a GPLOT procedure.

The assignment of the position and input parameters (see step1) to the 3 SAS macros is performed by right-clicking on the selected macro. Then select 'properties' and add all parameters in the 'parameters' menu.

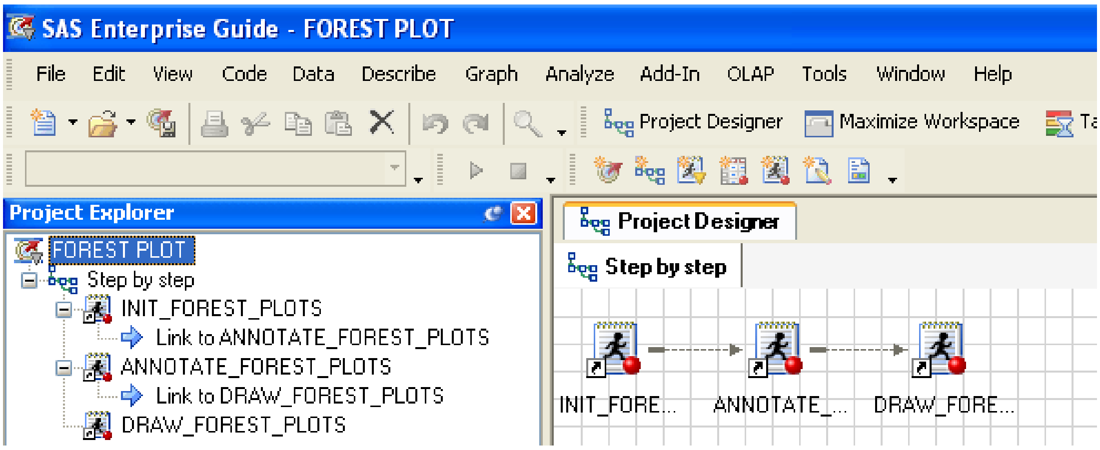


Figure 3: Screen to create a project with SAS Enterprise Guide

INIT_FOREST_PLOTS: INITIALISE PARAMETERS AND MACRO VARIABLES

This macro creates additional macro variables to:

- Identify the names of variables from input dataset
- Store the values of these variables
- Count the number of studies
- Set the offset and compute the length of windows
- Compute the weight that determines the size of squares

CREATE DERIVATED MACRO VARIABLES

The macro LIST2MACRO creates additional macro variables from the following position and input parameters OFFSET, WINORDER, VARWEIGHT and VARDATA defined in Step 1. For example, if WINORDER=(STUDY STAT GRAPH) then the macro creates three macro variables such as WINORDER1=STUDY, WINORDER2=STAT and WINORDER3=GRAPH. In this way, we facilitate the management of each element of these macro variables.

```
%MACRO LIST2MACRO(list=,macro_name=,count_name=);
       %LOCAL cpt;
       %let cpt=1;

       %do %while(%scan(&list,&cpt) ne );
              %GLOBAL &macro_name&cpt;
              %let &macro_name&cpt = %scan(&list,&cpt);
              %let cpt=%eval(&cpt+1);
       %end;
       %GLOBAL &count_name;
       %let &count_name = %eval(&cpt-1);
%MEND;

%LIST2MACRO(list=&offset,macro_name=offset,count_name=nb_offset);
%LIST2MACRO(list=&winorder,macro_name=winorder,count_name=nb_winorder);
%LIST2MACRO(list=&varweight,macro_name=varweight,count_name=nb_varweight);
%LIST2MACRO(list=&vardata,macro_name=vardata,count_name=nb_vardata);
```

CREATE MACRO VARIABLES CONTAINING THE VALUES OF VARIABLES

In order to create the annotate dataset and draw the forest plot with the GPLOT procedure, we store the values of each variable into macro variables.

```
/* read all vardata values and create macro-variables */
%do i=1 %to &nb_vardata;
       data _NULL_;
              set &dsname;
              call symput("&&vardata&i"||compress(_N_),trim(left(&&vardata&i)));
              call symput("nb_&&vardata&i",compress(_N_));
       run;
%end;

/* read all varweight values and create macro-variables */
%do i=1 %to &nb_varweight;
       data _NULL_;
              set &dsname;
              call
       symput("&&varweight&i"||compress(_N_),trim(left(&&varweight&i)));
       call symput("nb_&&varweight&i",compress(_N_));
       run;
%end;

/* read all studies, varlow, varhigh and varor */
data _NULL_;
       set &dsname;
       call symput("&varstudy"||compress(_N_),trim(left(&varstudy)));
       call symput("&varlow"||compress(_N_),trim(left(&varlow)));
       call symput("&varhigh"||compress(_N_),trim(left(&varhigh)));
       call symput("&varor"||compress(_N_),trim(left(&varor)));
run;
```

COUNT NUMBER OF STUDIES

STUDID_MAX count the total number of observations in the input dataset, STUDID_NB contains the number of individual studies.

```
proc sql noprint;
       select max(id)
       into :studid_max
       from &dsname;
       select count(id)
       into :studid_nb
       from  &dsname
       where &varstudy ne 'COMBINED';
quit;
%let studid_max=%trim(%left(&studid_max));
%let studid_nb=%trim(%left(&studid_nb));
```

DETERMINE OFFSET AND LENGTH OF EACH WINORDER'S ELEMENT

We define the offset and the length of each element of the WINORDER position parameter. When WINORDER contains GRAPH, STAT and STUDY, we create the following macro variables: GRAPH_OFFSET, GRAPH_LENGTH, STAT_OFFSET, STAT_LENGTH, STUDY_OFFSET and STUDY_LENGTH.

```
%do i=1 %to &nb_winorder;
       %let &&winorder&i.._offset=&&offset&i;
       %if &i ne &nb_winorder %then %do;
              %let tmp=%eval(&i+1);
              %let &&winorder&i.._length=%sysevalf(&&offset&tmp-&&offset&i);
       %end;
       %else %let &&winorder&i.._length=%sysevalf(100-&&offset&i);
%end;
```

COMPUTE THE WEIGHT AND CREATE WEIGHT MACRO VARIABLES

Here, the weight of each study is represented by the sum of patients in the treatment arm and in the control arm divided by the grand total number of patients in both groups. It defines the size of squares in the forest plot. The user can easily redefine the weight of each study by the VARWEIGHT macro variable.

```
data _NULL_;
       set &dsname(where=(&varstudy ne 'COMBINED')) end=eof;
       ARRAY weight{&nb_varweight} &varweight;
       do i=1 to &nb_varweight;
              sum_weight=sum(sum_weight,weight(i));
              tot_weight+sum_weight;
       end;
       call symput("weight"||compress(_N_),trim(left(sum_weight)));
       if eof then call symput("tot_weight",trim(left(tot_weight)));
run;

%do i=1 %to &studid_nb;
       %let weight&i=%sysevalf(&&weight&i/&tot_weight);
%end;
```

ANNOTATE_FOREST_PLOTS: CREATE THE ANNOTATE DATASET

The annotate dataset allows to draw some features in a graph such as the lines, the rectangles, the circles, the pies… SAS provides macros to facilitate this work. These macros are %LABEL, %BAR, %MOVE, %SYSTEM, %LINE, %POLY and %POLYCONT. They are used to position and label the elements (lines, axis, squares, diamonds, group labels, horizontal axis, legend, …) in the 3 windows. Each window is managed in the annotate dataset, separately.

The coordinate systems are used to define a repository for positioning the objects i.e. lines, squares, …. It must be used with caution (absolute or relative system, values or percentage coordinate, choice of the output area). We use different parameters of %SYSTEM macro for setting the coordinate system as followed:

%SYSTEM(2,2,5):	We work with the values on X and Y axis. The object's height is defined in percentage of total output area.
%SYSTEM(B,B,5):	We work with the relative percentages of total output area compared to a point defined by %MOVE(X,Y) macro. The object's height is defined in percentage of total output area.
%SYSTEM(5,5,5):	We work with the percentages of total output area.
%SYSTEM(5,2,5):	We work with the values on Y axis. X axis and the object's height are defined in percentage of total output area.

For complete information on coordinate systems, see SAS documentation [5].

LINK THE PLOTS

```
%SYSTEM(2,2,5);
%do i=1 %to &studid_nb;
       %LINE(&&&&&VARLOW&i,&i,&&&&&VARHIGH&i,&i,CX000000,1,0.2);
%end;
```

DRAW SQUARES

In the 'graph' window, the size of the squares, which represents the weight of each study, is centered on the outcome and is adjusted by a coefficient in order to take the number of studies into account. This ensures that the squares aren't too small when there is only a few studies and that the squares don't overlap each other when the number of studies is large. The coefficient decreases with the number of studies.

```
%if &studid_nb < 10 %then %let sq_coef=%sysevalf(10-&studid_nb);
%else %let sq_coef =1;
```

For each study, we move on the odd ratio value i.e. the center of the square. Then, we set coordinate systems in relative percentage of the total area.
A move to the bottom left-hand part of the square must be performed. Due to output paper format (A4) and orientation (landscape), vertical coordinates are multiplied by 29.4/21=1.4 to adjust to the proper proportions. FILL_SQ is an additional parameter which can be either "solid" or "empty", indicating whether the inside part of the squares and diamonds are colored in black or left in blank. The %BAR macro function is used to draw the square.

```
%do i=1 %to &studid_nb;
       %SYSTEM(2,2,5);
       /* Move on OR value */
       %MOVE(&&&&&VAROR&i,&i);
/* First move to the lower left corner, 1.4 is the adjustment */
/* for a A4 paper (29.4/21)                                   */
       %SYSTEM(B,B,5);
       %MOVE(-%sysevalf(&&weight&i*&sq_coef.),
            -%sysevalf(&&weight&i*&sq_coef.*1.4)
            );

/* Then move to the upper right corner */
       %BAR( 0,0,
             %sysevalf(&&weight&i*2*&sq_coef.),
             %sysevalf(&&weight&i*2*&sq_coef.*1.4),
             CX000000, 0.2, &fill_sq.
           );
%end;
```

DRAW DIAMOND

The combined outcome is represented by a diamond. We link OR, OR_LOW, OR_HIGH and set a vertical offset to ±0.2. This value (0.2), defined arbitrarily, allows to control the shape of the diamond. We use %POLY and %POLYCONT macros.

```
%SYSTEM(2,2,5);

%POLY( x1=&&&&&VARLOW&studid_max,
       y1=&studid_max.,
       color=CX000000,
       lintyp=1,
       pattern=m&fill_sq.
     );

%POLYCONT( x1=&&&&&VAROR&studid_max,
           y1=%sysevalf(&studid_max.-0.2),
           colin=CX000000
         );

%POLYCONT( x1=&&&&&VARHIGH&studid_max,
           y1=&studid_max.,
           colin=CX000000
         );

%POLYCONT( x1=&&&&&VAROR&studid_max,
           y1=%sysevalf(&studid_max.+0.2),
           colin=CX000000
         );

%POLYCONT( x1=&&&&&VARLOW&studid_max,
           y1=&studid_max.,
           colin=CX000000
         );
```

WRITE STUDY HEADER AND VALUES

First, we test if the STUDY_OFFSET macro variable exists i.e. WINORDER position parameter contains the ‘study identification’ window. Then, for each study, we draw the studies header and values using %LABEL.

```
%if %SYMEXIST(STUDY_OFFSET) %then %do;
       /*Headers*/
       %SYSTEM(5,5,5);
       %LABEL (&STUDY_OFFSET,
               &vardata_header_position,
               "STUDY",
               CX1206C9,0,0,&vardata_size,SIMPLEX,&varpos
```

```
              );
       /*Values*/
       %SYSTEM(5,2,5);
       %do i=1 %to &studid_max.;
              %LABEL (&STUDY_OFFSET,
                     &i,
                     "&&&&&VARSTUDY&i",
                     CX1206C9,0,0,&vardata_size,SIMPLEX,&varpos
                     );
       %end;
%end;
```

WRITE STATISTICS HEADER AND VALUES

First, we test if the STAT_OFFSET macro variable exists i.e. WINORDER position parameter contains the 'statistics' window. For each study, we draw the statistics header and values using %LABEL. We divide the length of the 'statistics' window by the number of elements in VARDATA macro variable. Then, each column is positioned. The first column starts with STAT_OFFSET in a horizontal position.

VARPOS is a parameter that defines the label alignment and placement.
For complete information on label alignment and placement, see SAS documentation [6].

```
%if %SYMEXIST(STAT_OFFSET) %then %do;
       /*Headers*/
       %SYSTEM(5,5,5);
       %do i=1 %to &nb_vardata;
         %LABEL (%sysevalf(&STAT_OFFSET+%eval(&i-1)*(&STAT_LENGTH/&nb_vardata)),
                &vardata_header_position, "&&vardata&i", CX1206C9, 0, 0,
                &vardata_size,
                SIMPLEX,
                &varpos
                );
       %end;

       /*Values*/
       %SYSTEM(5,2,5);
       %do i=1 %to &nb_vardata;
         %do j=1 %to &&&&nb_&&vardata&i;
          %LABEL (%sysevalf(&STAT_OFFSET+%eval(&i-1)*(&STAT_LENGTH/&nb_vardata)),
                 &j,
                 "&&&&&&vardata&i..&j",CX1206C9,0,0,
                 &vardata_size,
                 SIMPLEX,
                 &varpos
                );
         %end;
       %end;
%end;
```

DRAW_FOREST_PLOTS: DRAW THE FOREST PLOT

The last step consists in the specification of the symbol and axis statements. According to what was defined earlier, the length and the origin of the horizontal axis can now be defined dynamically using GRAPH_OFFSET and GRAPH_LENGTH. Here, the horizontal axis is scaled in log base 10 because the example of meta-analysis is based on binary outcomes (odds ratio).

```
option ORIENTATION=LANDSCAPE ;
goption RESET=ALL GUNIT=PCT target=GIF;

/* Horizontal axis */
AXIS1  LOGBASE=10 LOGSTYLE=EXPAND LABEL=NONE
       ORIGIN=(%sysevalf(&GRAPH_OFFSET),&h_axis_origin)
       VALUE=(HEIGHT=&h_tick_size)
       LENGTH=%sysevalf(&GRAPH_LENGTH)
;

/* Vertical axis */
```

```
AXIS2  ORDER=(%eval(&studid_max) to 1 by -1)
       ORIGIN=(&GRAPH_OFFSET)
       OFFSET=(&v_axis_offset,&v_axis_offset)
       LABEL=NONE MAJOR=NONE MINIR=NONE VALUE=NONE STYLE=0;

SYMBOL1 VALUE=dot HEIGHT=0.2 REPEAT=3;
```

The PROC GPLOT is used with the specification of the annotate dataset.

```
TITLE1 "Example of flexible forest plot with SAS enterprise Guide";
TITLE2 " Application to a simple meta-analysis based on binary outcomes";
FOOTNOTE1 "The ‘statistics’ window is on the right of the ‘graph’ window";
FOOTNOTE2 "We add the total number of death in each study";

proc gplot data=&dsname annotate=ANNO_FOREST_PLOT;
       plot &studid*&varor &studid*&varlow &studid*&varhigh / overlay
       haxis=axis1 vaxis=axis2 href=1;
run;
quit;
```

Step 3
The final step is optional but permits a more general use of the SAS macros provided that we insert them in one program. With SAS Enterprise Guide, we can create a stored process by right clicking on the code and selecting ‘Create stored process…’ and with SAS add-in for Microsoft Office®, user can generate a forest plot directly in Excel® and Word®.

RUN FOREST! RUN!: AN EXAMPLE

In this example, we use the same dataset described in the ’Input data source’ paragraph with different parameters than those used in Figure 1. Here, we display the information related to the odds ratio and 95% confidence interval on the right of the graph and add the total number of deaths for each study in the ‘statistics’ window.

The general process is as follows:

(1) Initialize global parameters (position and input parameters)
(2) Compile the three main macros INIT_FOREST_PLOTS, ANNOTATE_FOREST_PLOTS and DRAW_FOREST_PLOTS
(3) Create a program which calls them and run the project by clicking on the ‘Run’ button

A screen is created (Figure 4). It displays the values of the parameters by default. We enter the values TOT_DEATH NBPAT_GRPA NBPAT_GRPB OR_CONCAT for the ‘List of data/statistics variables to display in STAT window’ and STUDY GRAPH STAT for the ‘Windows order’. Due to the screen capture, we cannot see all the parameters (first circle in figure 4). The macro variables related to this information are VARDATA and WINORDER.

(4) Click on the ‘Run’ button

A flexible forest plot is automatically created (Figure 5).

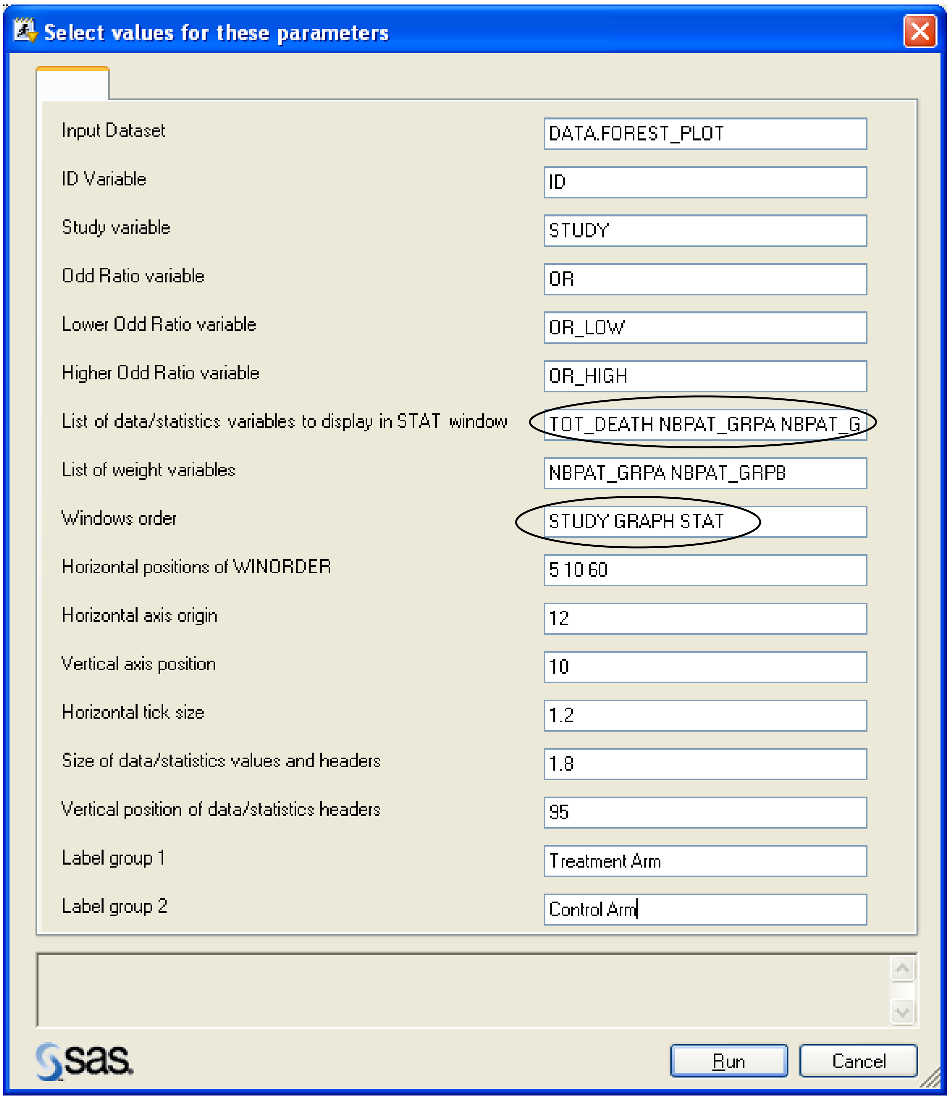


Figure 4: Parameter values screen created by the process

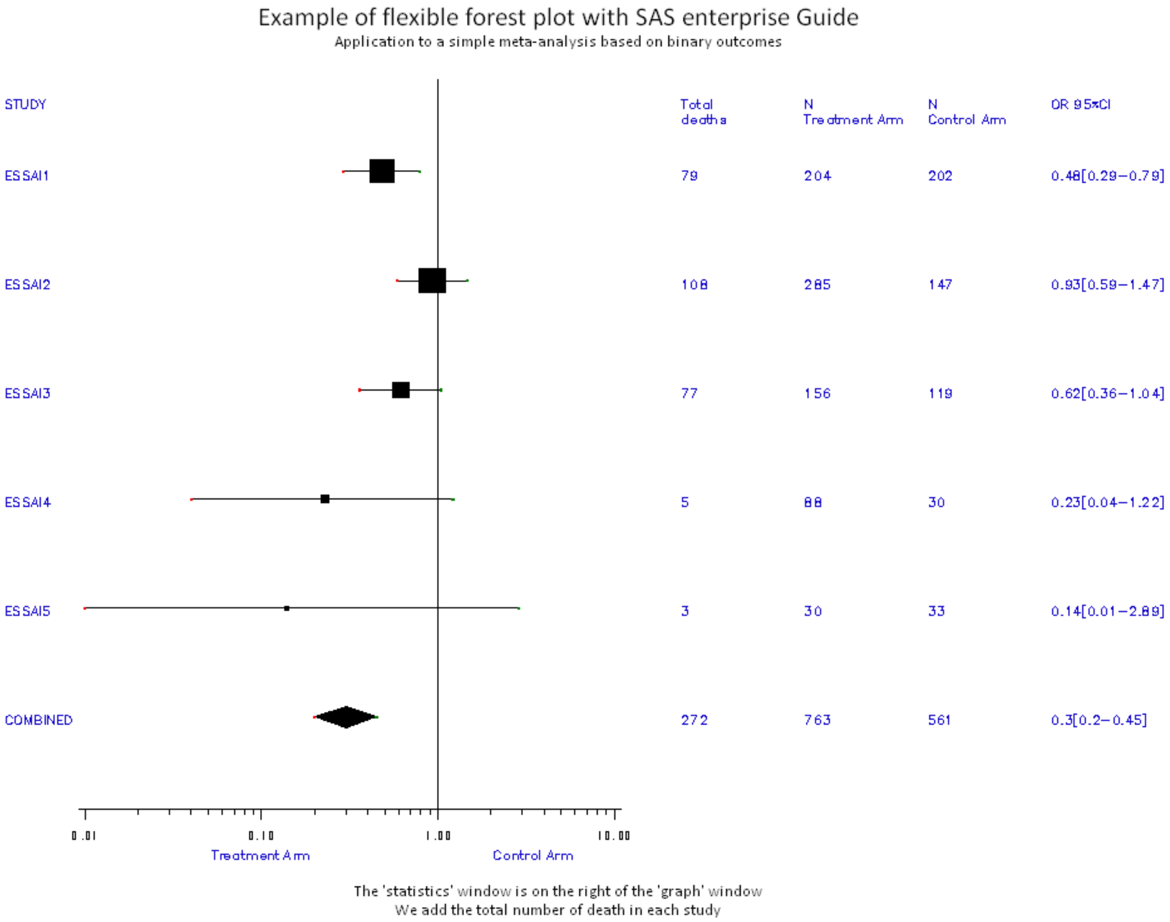


Figure 5: Flexible forest plot with SAS Enterprise Guide

CONCLUSION

In this paper, we presented the generation of a flexible forest plot with SAS Enterprise Guide based on the decomposition of the output in three windows and using SAS annotate macros. The information displayed can be easily extended by the addition of variables to the 'study identification' or 'statistics' windows by the user. The development with SAS Enterprise Guide permits via a stored process to extend its application in Windows applications. Further works are ongoing among which one is to create a new horizontal window to write additional information and another is to manage the display of the results of subgroups analyses.

ACKNOWLEDGMENTS

We thank Salah Ghabri and Kirsten Dumaz for reviewing this paper.

REFERENCES

[1] Egger M, Sterne J.A.C., Davey Smith G. Meta-analysis software. British Medical Journal, 316 (2000).

[2] Sutton AJ, Abrams KR, Jones DR, Sheldon TA, Song F. Methods for Meta-Analysis in Medical Research. Wiley.

[3] Mitchell RM. Forcing SAS/GRAPH® software to meet my statistical needs: A graphical presentation of odds ratios. SUGI 25, Paper 882-887 (2000).

[4] Gary A. Foster, Charles H. Goldsmith. Problems commonly associated with forest plots addressed using high resolution graphics in SAS. SUGI 31, Paper 139-31 (2006).

[5] http://support.sas.com/onlinedoc/913/getDoc/fr/graphref.hlp/annodata-creating-grelem.htm

[6] http://support.sas.com/onlinedoc/913/getDoc/fr/graphref.hlp/annotate_position.htm

CONTACT INFORMATION

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

Adrien VALLEE
KEYRUS BIOPHARMA
155 rue Anatole France
92593 Levallois-Perret Cedex
FRANCE
Phone: +33 (0)1 41 34 28 28
Fax: +33 (0)1 41 34 28 29
Email: adrien.vallee@keyrus.com
Web: www.keyrus.com
www.keyrusbiopharma.com