Create Flowcharts Using Annotate Facility

Priya Saradha & Gurubaran Veeravel
PPD, Inc., Hamilton, NJ

Abstract

With mounting significance to the graphical presentation of data in different forms in the pharmaceutical industry, as well as in other industries, the need to create non-traditional graphs (flowcharts, super-imposing different types of graphs, etc.,) has increased in recent times. This paper will lay a strong foundation for the creation of flowcharts with the help of the ANNOTATE facility by discussing the newly developed macro, FLOWAN. FLOWAN can create a highly-customized flowchart with multiple levels using ANNOTATE macros and functions. Using a dataset as input, FLOWAN can represent various populations in a study or any other type of data in the form of a flowchart. This macro uses very few parameters like, the number of tiers (levels), the number of boxes in each level, the variables to display and the link or predecessor for each of the boxes. The output from this macro will present the data in an innovative way. Recently some clinical studies have used the flowchart technique to represent patient disposition data. Apart from aiding the creation of traditional graphs like line plots, scatter plots, bar charts, etc., the ANNOTATE facility has been a very useful tool to add to the capabilities of SAS/GRAPH® procedures to produce newer type of graphs, such as flowcharts. This paper will discuss in detail the FLOWAN macro and the new technique of creating flowcharts using SAS/GRAPH® and ANNOTATE facility.

Introduction

The usage of SAS/GRAPH® has gained tremendous importance in recent times to develop non-traditional highly customized graphs using ANNOTATE facility to meet the needs of the industry. The ANNOTATE Facility allows the user to create customized modifications to the graphic output. These modifications can be either predetermined or may be data driven. The ANNOTATE facility can be used for tasks as simple as displaying a p-value to tasks as complicated as displaying standard error bars for every data point in a line plot or every single bar in a bar chart. Although fluency in ANNOTATE takes time to develop, it provides a powerful, intuitive, and flexible means of building a graph to virtually any specification. Familiarity with ANNOTATE also offers an effective 'safety net' against all the last-minute changes made to the graphical specifications/requirements, since it allows you to control every pixel of the graph.

In an effort to present data in a new dimension, we experimented with a new form of graph, flowchart (as shown in Fig-1), for representing Patient Disposition Data. When we realized that there are some other companies trying to do a similar graph, we went on to generalize the algorithm and developed a macro FLOWAN, which can be used with any kind of data. This approach gives us a better understanding of the ANNOTATE macros and also provides a gateway to present data in a new facet.
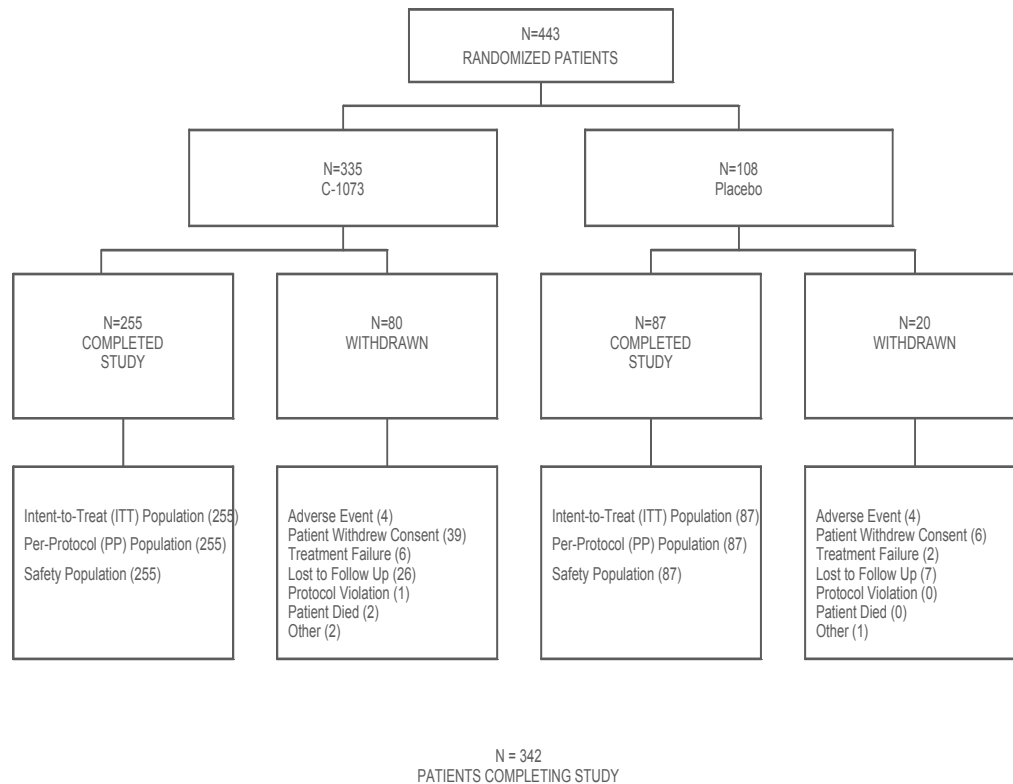
```
                        ┌─────────────────────┐
                        │       N=443         │
                        │ RANDOMIZED PATIENTS │
                        └─────────────────────┘
```

| N=335 C-1073 | N=108 Placebo |

| N=255 COMPLETED STUDY | N=80 WITHDRAWN | N=87 COMPLETED STUDY | N=20 WITHDRAWN |

Intent-to-Treat (ITT) Population (255)	Adverse Event (4)	Intent-to-Treat (ITT) Population (87)	Adverse Event (4)
Per-Protocol (PP) Population (255)	Patient Withdrew Consent (39)	Per-Protocol (PP) Population (87)	Patient Withdrew Consent (6)
Safety Population (255)	Treatment Failure (6)	Safety Population (87)	Treatment Failure (2)
	Lost to Follow Up (26)		Lost to Follow Up (7)
	Protocol Violation (1)		Protocol Violation (0)
	Patient Died (2)		Patient Died (0)
	Other (2)		Other (1)

N = 342
PATIENTS COMPLETING STUDY

Fig-1

ANNOTATE Functions and Macros

The ANNOTATE facility can be described simply as a means of using data step processing to create drawing instructions, which are then rendered as graphical output through SAS/GRAPH® procedures. The power of the ANNOTATE facility is accessed through the use of a specialized dataset. When using this dataset, ANNOTATE looks for variables with specific names and attributes, the values taken on by these variables in turn instruct ANNOTATE to your intentions. The dataset itself is fairly rigidly defined in terms of the names of the variables and their attributes. ANNOTATE functions and x, y coordinates describe the type and dimensions of each object. The most basic ANNOTATE functions include:

- MOVE (move to the new x,y coordinates)
- DRAW (draw a line from the last pair of x,y coordinates to the current ones)
- BAR   (draw a rectangle from the last pair of x,y coordinates to the current ones)
- LABEL (places text at the current x,y coordinates)

These functions are essentially qualified by the associated variable values present on the same record. For instance, if an attribute variable such as COLOR is retained, then its last initialization will determine the color value for each subsequent DRAW or BAR function.

ANNOTATE macros (i.e. %label, %line, %bar) are supplied as part of the ANNOTATE facility, and are typically stored at 'C:\Program Files\SAS Institute\SAS\V8 \core\sasmacro'. They can greatly reduce the code required to create an ANNOTATE data set, since they set attribute variables and capture repetitive MOVE and DRAW function calls. They effectively eliminate the

need for the programmer to keep track of the position of the 'drawing tool' before and after each instruction.

%FLOWAN Macro:

The fundamental principle of the FLOWAN macro is to use the ANNOTATE functions and macros to create an ANNOTATE dataset, which would be used to draw the flowchart using GPLOT Procedure. The FLOWAN macro creates an ANNOTATE dataset based on input macro parameters. This ANNOTATE data set, simply put, contains observations where each observation represents a command to draw a graphical element or to perform a graphical action. The observations use a set of predefined ANNOTATE variables. The values of these variables in the observation determine the action and the methodology of executing the graphical command. Few of these variables like color, thickness, x and y coordinates, axes, and graphical coordinate systems are assigned within the macro. Internally, a do loop is executed for each level and variable to be displayed. This loop creates observations to be written into the ANNOTATE dataset. Finally a Proc GPLOT prints out the graph by reading the ANNOTATE dataset. The macro also takes care of assigning 1 for thickness, black for color, 2 for x and y co-ordinate systems and also to suppress the axes and data values to be plotted on the graph.

The FLOWAN macro can be invoked as follows:

```
%FLOWAN
  (
    data    = ,                              /**** Input Dataset *****/
    tiers   = ,                         /**** Max. Number of levels ****/
    maxvars = ,                      /*Max. No. of variables in a level*/
    levelx = %str(n x-1=var1 x-1=var2 x-1=var3) /* Level x variables */
    ... repeat for other levels ...
  );
```

The input parameters:

DATA =   input dataset.
TIERS =   the number of tiers in the flowchart. This parameter takes care of centering the
                flowchart vertically
MAXVARS =   Takes the maximum number of variables in the longest level. This parameter is
                used to center the flowchart horizontally.
LEVELx =   Takes the variables names that need to be displayed in level x.

Of the above parameters, LEVEL parameters require the input values to be in a specific format as shown below.

Each input value can be divided into 3 parts.
- The first part is the number of variables to be displayed in this level.
- The second part is the variable number or box number in the previous level that would be the predecessor for this variable.
- The third part is the variable to be displayed.

levelx = %str(n x-1=var1 x-1=var2 x-1=var3)

- o x is the tier number
- o n is the number of variables to be displayed in the xth tier of the flowchart

3

o The x-1 in x-1=var1 is the node number or variable in the preceding level.
o The value var1 in x-1=var1 is the variable to be displayed.

The macro call is constructed in such a simple manner, that it becomes inevitable for you to think of the requirements of the input dataset. Of course, the input dataset needs to be in a specific structure as shown below (Fig-2).

You should create an input dataset (which contains values to be displayed in the flowchart) with 3 variables named as label, variable, and value. The variable "Label" holds the string to be displayed in the box, "Variable" is the variable name being passed as a value through the macro variables level1, level2, etc., and the "Value" is the variable containing the number of patients or events that would be displayed underneath the label in the box.

Example - Implementing the macro %FLOWAN

To illustrate to macro %FLOWAN in action, data collected from a clinical trial with over 445 patients was used. The dataset DISP (Fig-2) contains the patient disposition information to be displayed in the graph.

	label	variable	value
1	RANDOMIZED	rand	443
2	COMPLETED STUDY	comp	341
3	ABC	tmtr1	332
4	Placebo	tmtr2	111
5	ITT	itt1	259
6	ITT	itt2	82
7	Safety	saf1	259
8	Safety	saf2	82
9	Per-protocol	pp1	178
10	Per-protocol	pp2	62

Fig-2

The macro %FLOWAN will be called using the following structure:

```
%flowan(      data    = disp,
              tiers   = 3,
              maxvars = 3,
              level1  = %str(1 0=rand),
              level2  = %str(2 1=tmtr1 1=tmtr2),
              level3  = %str(3 1=itt1 1=saf1 2=itt2)
              );
```

Fig-3 is the flowchart produced from the above macro call. Data from the dataset DISP (Fig-2) is presented in 3 tiers and the maximum number of boxes in any level is 3 (MAXVARS=3). The LEVEL1 variable reads the value satisfying the condition "variable = rand" and displays the corresponding data from the variable "label" as the text inside the box and the data from the variable "value" as the population size. In a similar fashion, tier 2 displays 2 variables, TMTR1 and TMTR2 corresponding to the values ABC and Placebo respectively. The final tier has 3

boxes each representing the individual populations ITT and Safety for each of the treatment groups described above.
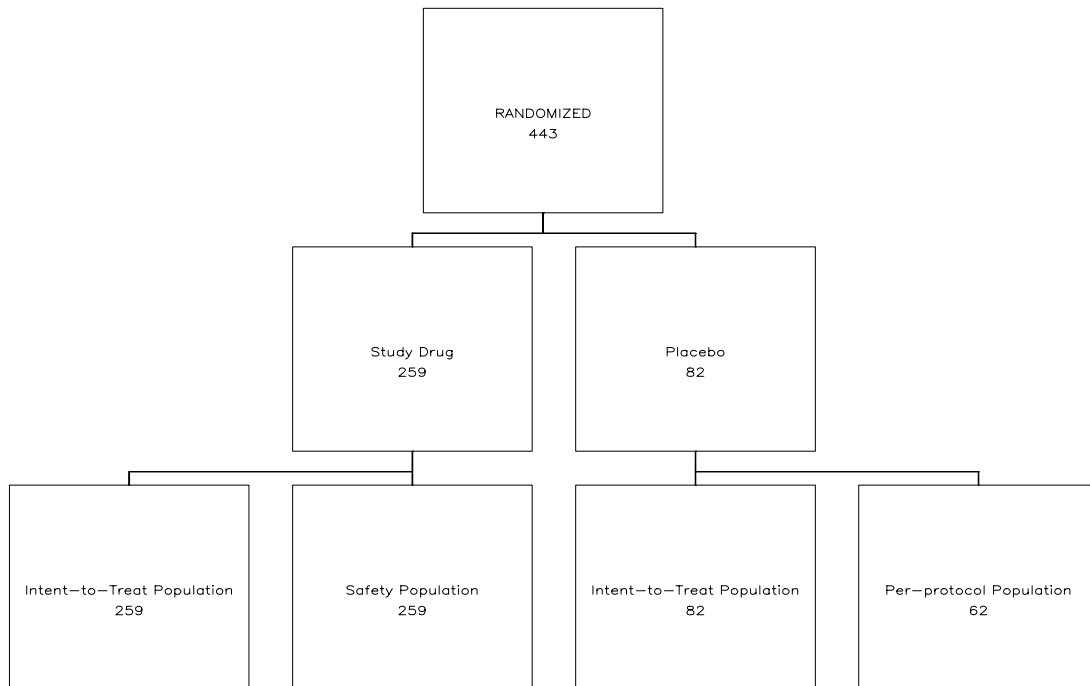

Fig-3

Conclusion:

As you can see, this paper has laid a strong foundation for the creation of flowcharts with the help of the ANNOTATE facility. This macro has brought out the power of the ANNOTATE facility in creating flowcharts. SAS$^®$ users should be able to develop various challenging graphs by combining ANNOTATE facility with graphical procedures. You can modify this macro to display other type of graphs like a bar chart, pie chart or a line plot as nodes of the flowchart. You can also have this macro as a starter; reduce the limitations of this macro, so that you can use this to display more levels and more variables in each level.

The FLOWAN macro can create a highly-customized flowchart with multiple levels with limits such as,

➢ Currently, this macro cannot handle more than 4 levels and/or more than 4 variables in the same level.
➢ You cannot display more than one label and a count within a single box.
➢ FLOWAN macro is specific to create a flow chart in a vertical fashion, you cannot move horizontally.

Some sections of the macro are presented below:

```
/***** Prepare the ANNOTATE macros ****/
%annomac;
/****** Initializing the ANNOTATE Dataset with default values ****/
data anno;
   length function style $8 text $80 label $200 col $50 value $20;
   retain function '' color 'black' xsys '2' ysys '2' when 'a' style ''
          boxno 0 levelno 0 x1 x2 y1 y2 x y line size 0 text '' col ''
          value '' pred 0 section 0 angle 0 rotate 0 position '0';
run;


/**** Below is the code to generate the intermediate datasets which
finally are set   together to form the ANNOTATE dataset ******/
data _vars&i&j;
   length function style $8 text $80 label $200 col $50 value $20;
   retain color 'black' xsys '2' ysys '2' when 'a' style 'solid'
          x1 x2 y1 y2 0 value ' ';
   set all;
   where compress(lowcase(col)) eq compress(lowcase("&&_vars&i&j"));

/**** Drawing and labeling nodes ******/
section=1;
 %bar(x1,y1,x2,y2,black,0,1);
 %LABEL((x2+x1)/2, (y2+y1)/2, label, black, 0, 0, 0.8, courier, 0);
 %LABEL((x2+x1)/2, (y2+y1)/2 - 3, value, black, 0, 0, 0.8, courier, 0);
/**** Drawing the extender lines from each node *****/
section=2;
%if %eval(&i) ne %eval(&tiers) %then %do;
  %line((x1+x2)/2,y1,(x1+x2)/2,y1-%eval(&_yblank/2)-1,black,0,1);
%end;
section=3;
%if %eval(&i) ne 0  and &&__pred&j ne 0  %then %do;
  %line((x1+x2)/2,y2,(x1+x2)/2,y2+%eval(&_yblank/2),black,0,1);
%end;
run;

proc append base=anno data=_vars&i&j force;
run;

/***** Code to display the graph using Proc GPLOT *****/
proc gplot data=anno;
  plot y*x / annotate=anno haxis=axis1 vaxis=axis2;
run;
quit;
```

References

Arthur L. Carpenter – (Paper) -- Using ANNOTATE MACROS as Shortcuts
Brian Fairfield-Carter, PRA International, Victoria, BC – (2004) -- A Stand-Alone SAS® Annotate
System for Figure Generation

Contact Information (header 1)

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

Author Name   Gurubaran Veeravel and Priya Saradha

Company   PPD, Inc.

Address   3575 Quakerbridge Road

City state ZIP Hamilton, NJ

Work Phone: 609-528-8121

Fax: -

Email: Guru.Veeravel@lawrence.ppdi.com / Priya.saradha@lawrence.ppdi.com

Web: www.ppdi.com