PharmaSUG 2015 - DV04

An Enhanced Forest Plot Macro Using SAS®

Janette Garner, Gilead Sciences, Inc., Foster City, CA

ABSTRACT

A forest plot allows for a quick visualization of results across multiple subgroups. Additional information such as the actual values of the forest plot or the response rates in each treatment group is usually not included in a basic forest plot. Building upon forest plot code written by Sanjay Matange, a platform-independent macro was developed to allow the user to create a forest plot with the additional information mentioned above. Moreover, the plot can be enhanced with a bar plot of response rates in each treatment group and numerical labels of the values for the forest plot. It utilizes SG procedures available in SAS 9.3 (in particular, the HIGHLOWPLOT statement) and a standardized input dataset. The macro can accommodate the data based on the differences between two treatment groups or the odds/risk ratio. Readability has also been improved by adding reference bands to the graph.

INTRODUCTION

Sanjay Matange provided code to generate a forest plot on his SAS Blog, "Graphically Speaking". The code was modified to include a bar plot and labels for the forest plot. To reduce the technical burden on the user, the code was written into a macro that works on any platform. This enhanced forest plot macro allows the user to not only create a forest plot, but also display a bar plot of response rates in each treatment group, show the numerical response rates (with percentages, if so desired), and display the numerical values of the difference (or odds/risk ratio) and confidence interval or any other statistical output (e.g. a p-value). Moreover, the user can specify labels for the forest plot and reference bands have been incorporated to enhance readability. The HIGHLOWPLOT statement was used extensively to draw the forest plot, the bar plot, and to control the labels on the side. The SCATTERPLOT statement with the datalabel option is used to plot the numeric values. Much of the customization is done in the PROC TEMPLATE code preceding the PROC SGRENDER code, which generates the final graph.

MOTIVATION

The following graph by Pozniak et al appeared in *Lancet Infectious Diseases*. It combines a forest plot and a bar plot to present the response rates for the two treatment groups as well as the differences and 95% confidence intervals for the overall results and ten subgroup analyses.

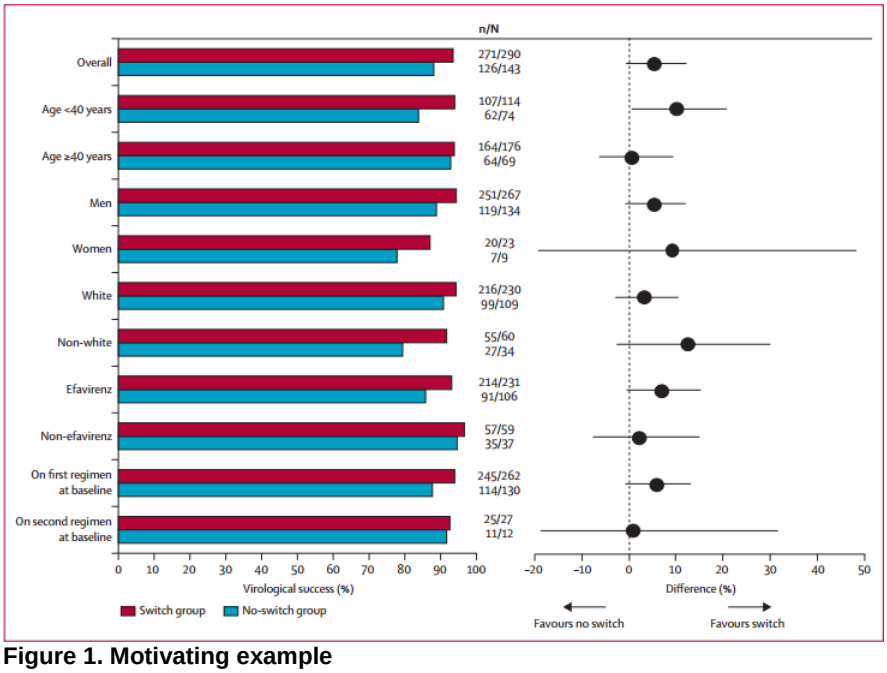

Figure 1. Motivating example

Here is the same figure produced in SAS 9.3 using the enhanced forest plot macro.

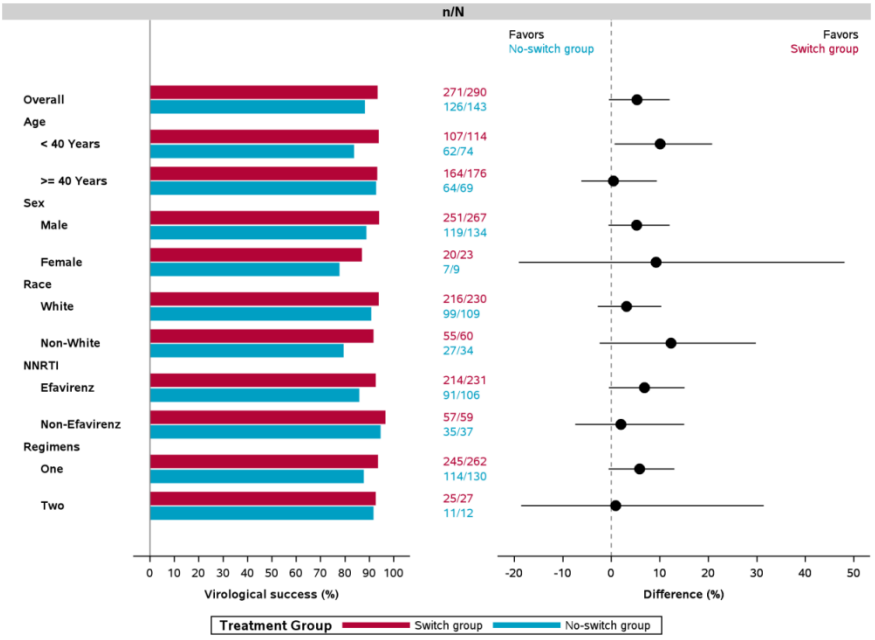

Figure 2. SAS reproduction of Figure 1.

MODIFYING THE FOREST PLOT

The forest plot can be modified in a variety of ways. The user can determine whether each of the columns is displayed (or hidden). A fifth column containing statistics (either p-values or the forest plot information) can be shown. The forest plot itself can be annotated to label the difference and the lower/upper bounds. Reference bands can also be included to enhance the readability. Figure 3, below, modifies Figure 2 to include labels on the forest plot, a column for p-values, and reference bands.

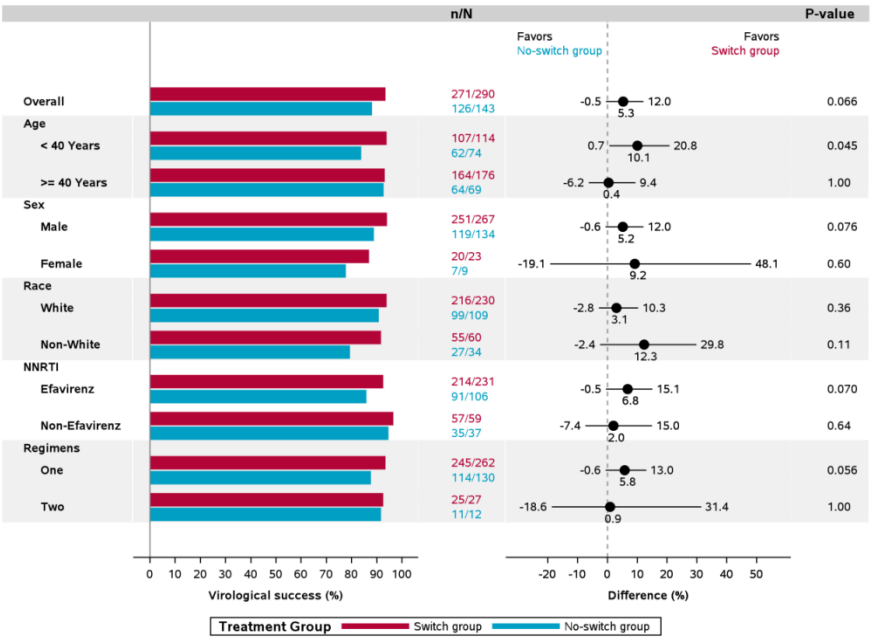

Figure 3. A modified form of Figure 2.

FULL SET OF OPTIONS

The full set of parameters for the template macro are as follows:

```
%macro forest(showcol2=Y,
              showcol3=Y,
              showcol4=Y,
              showcol5=Y,
              colwghts=%str(0.15 0.32 0.10 0.35 0.08),
              barval=%str(0 10 20 30 40 50 60 70 80 90 100),
              xlabbar=%bquote(Responders (%)),
              showpct=N,
              pctfmt=5.1,
              frsttype=DIFF/*RATIO*/,
              frstval=%str(-30 -20 -10 0 10 20 30)/*%str(0 0.5 1 2)*/,
              frstlabl=Y,
              xlabfrst=%bquote(Difference (%)),
              legtitle=%str(Treatment Group),
              ci_col=N,
              statlabl=%str(P-value)
              );
```

Additional options (e.g. colors and treatment labels) are specified in the SGRENDER call below.

BEHIND THE SCENES

The macro carries out a series of steps to produce the final output.

1. Read in a standardized dataset and modify it to include all of the information needed to produce the plot
2. Create a graphics template for the forest plot based on input from the user (e.g. which columns to display and what will be contained therein) – In effect, this creates the “canvas” for the graph
3. Render the graph using PROC SGRENDER and another set of (dynamic) variables (e.g. labels, colors) – Essentially, this is the paint that is added to the canvas

The results can be saved as a ODS Graphics Editor (SGE) file, a Portable Network Graphics (PNG) file, or embedded in a Rich Text Format (RTF) or Portable Document Format (PDF) file using ODS output.

Step 1: Create the dataset needed to produce the plot

SORT ORD	STAT ORD	*GROUP*	*LEVEL*	*N_TXT1*	*N_TXT2*	PERCENT1	PERCENT2	EST	LOW	HIGH	PVAL	*STAT_ TXT*
1	1	Overall		271/290	126/143	93.45	88.11	5.3	-0.5	12.0	0.066	0.066
2	1	Age	< 40 Years	107/114	62/74	93.86	83.78	10.1	0.7	20.8	0.045	0.045
2	2		>= 40 Years	164/176	64/69	93.18	92.75	0.4	-6.2	9.4	1.000	1.00
3	1	Sex	Male	251/267	119/134	94.01	88.81	5.2	-0.6	12.0	0.076	0.076
3	2		Female	20/23	7/9	86.96	77.78	9.2	-19.1	48.1	0.600	0.60
4	1	Race	White	216/230	99/109	93.91	90.83	3.1	-2.8	10.3	0.360	0.36
4	2		Non-White	55/60	27/34	91.67	79.41	12.3	-2.4	29.8	0.110	0.11
5	1	NNRTI	Efavirenz	214/231	91/106	92.64	85.84	6.8	-0.5	15.1	0.070	0.070
5	2		Non-Efavirenz	57/59	35/37	96.61	94.59	2.0	-7.4	15.0	0.640	0.64
6	1	Regimens	One	245/262	114/130	93.51	87.69	5.8	-0.6	13.0	0.056	0.056
6	2		Two	25/27	11/12	92.59	91.67	0.9	-18.6	31.4	1.000	1.00

Table 1. Example input dataset

The variables in italics in Table 1 are character values while the rest are numeric. Y-values are automatically assigned as well which control the vertical placement of all of the elements in the graph.

Thus, the following variables are added to the dataset in Table 1 above.

SORT ORD	STAT ORD	*GROUP*	*LEVEL*	…	*STAT_ TXT*	Y	Y1	Y2	Y3	ZERO	ONE	REF1	REF2	REF3
1	1	Overall		…	0.066	2.5	2	3	2.5	0	0.1			
2	1	Age	< 40 Years	…	0.045	4	5	6	5.5	0	0.1	5	6	4
2	2		>= 40 Years	…	1.00		7.5	8.5	8	0	0.1	7.5	8.5	7
3	1	Sex	Male	…	0.076	9.5	10.5	11.5	11	0	0.1			
3	2		Female	…	0.60		13	14	13.5	0	0.1			
4	1	Race	White	…	0.36	15	16	17	16.5	0	0.1	16	17	15
4	2		Non-White	…	0.11		18.5	19.5	19	0	0.1	18.5	19.5	18
5	1	NNRTI	Efavirenz	…	0.070	20.5	21.5	22.5	22	0	0.1			
5	2		Non-Efavirenz	…	0.64		24	25	24.5	0	0.1			
6	1	Regimens	One	…	0.056	26	27	28	27.5	0	0.1	27	28	26
6	2		Two	…	1.00		29.5	30.5	30	0	0.1	29.5	30.5	29

Table 2. Example transformed input dataset for graphing

The y-values (Y, Y1, Y2, Y3) control where various elements appear on the graph.

ZERO and ONE are used to define the positions for character text (e.g. the labels in the first column), as well as the n/N text and the statistics column (which displays the P-value in Figure 3).

The code which transforms the input dataset in Table 1 to that in Table 2 appears below.

```
data forest;
  set example;
  by sortord statord;
  retain y y1 y2 y3 0;
  if first.sortord then do;
    y = y2 + 1;
    y1 = y - 1.5;
    y2 = y - 0.5;
  end;

  /* Adding 2.5 to each y1 and y2 will insert a small space (of size 0.5)
between each subgroup chunk */
  y1 = y1 + 2.5;
  y2 = y2 + 2.5;
  y3 = mean(y1,y2);
run;

data plot;
  set forest;
  by y;
  /* Y-values associated with the header are only plotted once */
  if not first.y then y = .;
  /* We assign the variables "zero" and "one" as dummy variables to allow for
plotting of text in some of the columns, in particular the labels. */
  zero = 0;
  one = 0.1;
  /* For alternating groups, add variables so that a reference band can be
generated to aid in visualization */
  /* NOTE: To "disable" the reference bands, the user can set the color to
white (cxffffff) in the PROC SGRENDER in Step 3 */
```

```
  if mod(sortord,2) = 0 then do;
    ref1=y1;
    ref2=y2;
    /* Third reference line to fill in any whitespace that may appear */
    if y ^ = . then ref3 = y;
    else if y = . then ref3 = ref1 - 0.5;
  end;
  *** Shift the "Overall" label down ***;
  if group = "Overall" then y = y3;
run;

data plot;
  set plot;
  %if &frsttype=DIFF %then %do;
    ci_txt = strip(put(est,5.1))||"% ("||strip(put(low,5.1))||"% to
"||strip(put(high,5.1))||"%)";
  %end;
  %if &frsttype=RATIO %then %do;
    ci_txt = strip(put(est,5.1))||" ("||strip(put(low,5.1))||" to
"||strip(put(high,5.1))||")";
  %end;
  n_txt1p = strip(n_txt1)||" ("||strip(put(percent1,&pctfmt))||"%)";
  n_txt2p = strip(n_txt2)||" ("||strip(put(percent2,&pctfmt))||"%)";
run;

proc sql noprint;
  select max(y2) into :ymax from plot;
quit;
```

Step 2: Create a graphics template

The following code creates the template used to produce the final plots. This is heavily influenced by the code that Sanjay Matange used to create a forest plot in SAS 9.3 on his blog.

```
%let colcnt =
%eval(%sysfunc(count(&showcol2&showcol3&showcol4&showcol5,Y))+1);

proc template;
  define statgraph ForestPlot;
  *** Assign dynamic variables, where the options are specified in the
SGRENDER procedure ***;
  dynamic _headercolor _trt1labl _trt1color _trt2labl _trt2color _refcolor
_refsize _barsize _ymin _ymax _title _footnote;
    begingraph;
        *** Add a title, if one is specified in the SGRENDER procedure ***;
        EntryTitle _title /;
          layout lattice / columns=&colcnt columnweights=(&colwghts);

          *** Specify column headers ***;
        sidebar / align=top;
            layout lattice / rows=1 columns=&colcnt columnweights=(&colwghts)
                        backgroundcolor=_headercolor opaque=true;

            entry halign=center textattrs=(size=8 weight=bold) "";
                  %if &showcol2=Y %then %do; entry halign=center
textattrs=(size=8 weight=bold) ""; %end;
```

```
            %if &showcol3=Y %then %do; entry halign=center textattrs=(size=8
weight=bold) %if &showpct=N %then %do; "n/N" %end; %else %do; "n/N (%)" %end;
; %end;
            %if &showcol4=Y %then %do; entry halign=center textattrs=(size=8
weight=bold) ""; %end;
            %if &showcol5=Y %then %do; entry halign=center textattrs=(size=8
weight=bold) "&statlabl" ; %end;
             endlayout;
        endsidebar;

            *** Column 1: Labels of the comparisons ***;
          layout overlay / walldisplay=none xaxisopts=(display=none)
               yaxisopts=(reverse=true display=none
                          tickvalueattrs=(weight=bold)
linearopts=(viewmin=_ymin viewmax=_ymax));
             referenceline y=ref1 / lineattrs=(thickness=_refsize
color=_refcolor);
             referenceline y=ref2 / lineattrs=(thickness=_refsize
color=_refcolor);
             referenceline y=ref3 / lineattrs=(thickness=_refsize
color=_refcolor);
             *** Header, using "y" variable ***;
            highlowplot y=y low=zero high=zero / highlabel=group
              lineattrs=(thickness=0) labelattrs=(size=7 weight=bold);
              *** Subgroup, using "y3" variable ***;
            highlowplot y=y3 low=one high=one / highlabel=level
              lineattrs=(thickness=0) labelattrs=(size=7 weight=bold);
            endlayout;

            %if &showcol2=Y %then %do;
            *** Column 2: Barplots of the proportion of successes in each
treatment group ***;
          layout overlay / xaxisopts=(label="&xlabbar" labelattrs=(size=7
weight=bold)
                           linearopts=(tickvaluepriority=true
                              tickvaluelist=(&barval))
/*display=(tickvalues)*/) /* This hides the x-axis label */
               yaxisopts=(reverse=true display=none linearopts=(viewmin=_ymin
viewmax=_ymax)) walldisplay=none;
             referenceline y=ref1 / lineattrs=(thickness=_refsize
color=_refcolor);
             referenceline y=ref2 / lineattrs=(thickness=_refsize
color=_refcolor);
             referenceline y=ref3 / lineattrs=(thickness=_refsize
color=_refcolor);
             *** The barplots are created using a HIGHLOW plot, where the
thickness of the line is controlled by user. Tweaking is required
                 based on the number of rows, height of the output and
spacing between bars. ***;
            highlowplot y=y1 low=zero high=percent1 /
lineattrs=(thickness=_barsize color=_trt1color)
             labelattrs=(size=7 weight=bold) name='trt1'
legendlabel=_trt1labl;
            highlowplot y=y2 low=zero high=percent2 /
lineattrs=(thickness=_barsize color=_trt2color)
             labelattrs=(size=7 weight=bold) name='trt2'
legendlabel=_trt2labl;
```

```
            referenceline x=0;
          endlayout;
          %end;

          %if &showcol3=Y %then %do;
      *** Column 3: Column of values that support the barplots, n/N ***;
          layout overlay / walldisplay=none xaxisopts=(display=none)
               yaxisopts=(reverse=true display=none
                          tickvalueattrs=(weight=bold)
linearopts=(viewmin=_ymin viewmax=_ymax));
            referenceline y=ref1 / lineattrs=(thickness=_refsize
color=_refcolor);
            referenceline y=ref2 / lineattrs=(thickness=_refsize
color=_refcolor);
            referenceline y=ref3 / lineattrs=(thickness=_refsize
color=_refcolor);
        highlowplot y=y1 low=zero high=zero / %if &showpct=N %then %do;
highlabel=n_txt1 %end; %if &showpct=Y %then %do; highlabel=n_txt1p %end;
            lineattrs=(thickness=0) labelattrs=(size=7 color=_trt1color);
        highlowplot y=y2 low=zero high=zero / %if &showpct=N %then %do;
highlabel=n_txt2 %end; %if &showpct=Y %then %do; highlabel=n_txt2p %end;
            lineattrs=(thickness=0) labelattrs=(size=7 color=_trt2color);
          endlayout;
          %end;

          %if &showcol4=Y %then %do;
          *** Column 4: Plot the CI for the difference (and include the
point estimate) ***;
          layout overlay / xaxisopts=(label="&xlabfrst" labelattrs=(size=7
weight=bold)
                       linearopts=(tickvaluepriority=true
                          tickvaluelist=(&frstval)))
               yaxisopts=(reverse=true display=none linearopts=(viewmin=_ymin
viewmax=_ymax)) walldisplay=none;
            referenceline y=ref1 / lineattrs=(thickness=_refsize
color=_refcolor);
            referenceline y=ref2 / lineattrs=(thickness=_refsize
color=_refcolor);
            referenceline y=ref3 / lineattrs=(thickness=_refsize
color=_refcolor);
            %if &frsttype=DIFF %then %do;
            referenceline x=0 / lineattrs=(pattern=shortdash);
            %end;
            %if &frsttype=RATIO %then %do;
            referenceline x=1 / lineattrs=(pattern=shortdash);
            %end;
          highlowplot y=y3 low=low high=high / lineattrs=(thickness=1
color=black) %if &frstlabl=Y %then %do; lowlabel=low highlabel=high %end; ;
            scatterplot y=y3 x=est / markerattrs=(symbol=circlefilled
size=10 color=black) datalabelposition=bottom %if &frstlabl=Y %then %do;
datalabel=est %end; ;
            *** Insert labels that explain the plot, namely that the right
side favors Treatment 1 and the left side favors Treatment 2. The labels for
these are defined in the SGRENDER procedure. ***;
            layout gridded / rows=2 border=false autoalign=(topright);
          entry textattrs=(size=7) halign=right 'Favors';
          entry textattrs=(size=7 color=_trt1color) halign=right _trt1labl;
```

```
              endlayout;
              layout gridded / rows=2 border=false autoalign=(topleft);
                entry textattrs=(size=7) halign=left 'Favors';
                entry textattrs=(size=7 color=_trt2color) halign=left
_trt2labl;
              endlayout;
            endlayout;
            %end;

            %if &showcol5=Y %then %do;
            *** Column 5: Statistics (Numerical Values) columns ***;
          layout overlay / walldisplay=none xaxisopts=(display=none)
               yaxisopts=(reverse=true display=none linearopts=(viewmin=_ymin
viewmax=_ymax));
              referenceline y=ref1 / lineattrs=(thickness=_refsize
color=_refcolor);
              referenceline y=ref2 / lineattrs=(thickness=_refsize
color=_refcolor);
              referenceline y=ref3 / lineattrs=(thickness=_refsize
color=_refcolor);
            highlowplot y=y3 low=zero high=zero / %if &ci_col=Y %then %do;
highlabel=ci_txt %end; %if &ci_col=N %then %do; highlabel=stat_txt %end;
              lineattrs=(thickness=0) labelattrs=(size=7);
            endlayout;
          %end;
        endlayout;

        %if &showcol2=Y %then %do; /* Only display legend if barplot shown */
        *** Add legend to bottom of the graph to explain colored bars ***;
        layout globallegend / type=row weights=preferred
legendtitleposition=left;
              discretelegend 'trt1' 'trt2' / title="&legtitle"
titleattrs=(size=8 weight=bold) across=2 displayclipped=true;
        endlayout;
        %end;
      *** Add footnote text, if specified in the SGRENDER procedure ***;
      entryfootnote halign=left textattrs=(size=7) _footnote;
      endgraph;
  end;
run;
```

Step 3: Render the final graph

The following code produces the final graph. The user can specify some formatting choices such as labels and colors using dynamic variables.

```
proc sgrender data=plot template=ForestPlot;
  dynamic _refcolor='cxf0f0f0'
          /* Set _refcolor='cxffffff' (white) to "hide" the reference bars */
          _refsize=15
          _headercolor='cxd0d0d0'
          _trt1labl='Switch group'
          _trt1color='cxb30437'
          _trt2labl='No-switch group'
          _trt2color='cx009fc3'
          _barsize=11.5
          _ymin=2
```

```
            _ymax=&ymax.
            _title=''
            _footnote='';
run;
```

Of note, the _refsize and _barsize will need to be manually updated depending on the size of the plot and the number of rows to be graphed. As more data is displayed, the content gets scaled down automatically in SAS. However, the _refsize and _barsize are fixed values, so they would need to be reduced as the content increases.

SUMMARY

Building upon the work of Sanjay Matange, this paper presents a user friendly macro which generates a forest plot enhanced with features and options such as a bar plot and numerical values. The code makes heavy use of the HIGHLOWPLOT statement available in SAS 9.3. Using a standardized dataset, the code generates a customized layout using PROC TEMPLATE and creates the final graph using PROC SGRENDER. The user needs to adjust the size of the bars in the final graph, so a few calls of the macro may be required to obtain the desired output.

REFERENCES

Matange, S. Forest Plot with SAS 9.3. http://blogs.sas.com/content/graphicallyspeaking/2013/01/06/forest-plot-with-sas-9-3/. Accessed December 16th, 2014.

Pozniak A, Markowitz M, Mills A, et al. Switching to coformulated elvitegravir, cobicistat, emtricitabine, and tenofovir versus continuation of non-nucleoside reverse transcriptase inhibitor with emtricitabine and tenofovir in virologically suppressed adults with HIV (STRATEGY-NNRTI): 48 week results of a randomised, open-label, phase 3b non-inferiority trial. Lancet Infect Dis. 2014;14(7):590-599.

CONTACT INFORMATION

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

Name: Janette Garner
Company: Gilead Sciences, Inc.
Address: 333 Lakeside Dr
City, State ZIP: Foster City, CA 94404
Work Phone: (650) 372-7329
E-mail: janette.garner@gilead.com